Chain-of-Thought Driven Adversarial Scenario Extrapolation for Robust Language Models

Md Rafi Ur Rashid [1], **Vishnu Asutosh Dasu** [1] , **Ye Wang** [2], **Gang Tan** [1] , **Shagufta Mehnaz** [1]

[1]Pennsylvania State University, 201 Old Main, University Park, PA 16802
[2]Mitsubishi Electric Research Labs, 201 Broadway, Cambridge, MA 02139

Abstract

Large Language Models (LLMs) exhibit impressive capabilities, but remain susceptible to a growing spectrum of safety risks, including jailbreaks, toxic content, hallucinations, and bias. Existing defenses often address only a single threat type or resort to rigid outright rejection, sacrificing user experience and failing to generalize across diverse and novel attacks. This paper introduces **Adversarial Scenario Extrapolation (ASE)**, a novel inference-time computation framework that leverages Chain-of-Thought (CoT) reasoning to simultaneously enhance LLM robustness and seamlessness. ASE guides the LLM through a self-generative process of contemplating potential adversarial scenarios and formulating defensive strategies before generating a response to the user query. Comprehensive evaluation on four adversarial benchmarks with four latest LLMs shows that ASE achieves near-zero jailbreak attack success rates and minimal toxicity, while slashing outright rejections to ≤4%. ASE outperforms six state-of-the-art defenses in robustness-seamlessness trade-offs, with 92–99% accuracy on adversarial Q&A and 4–10× lower bias scores. By transforming adversarial perception into an intrinsic cognitive process, ASE sets a new paradigm for secure and natural human-AI interaction.

Extended version — https://arxiv.org/abs/2505.17089

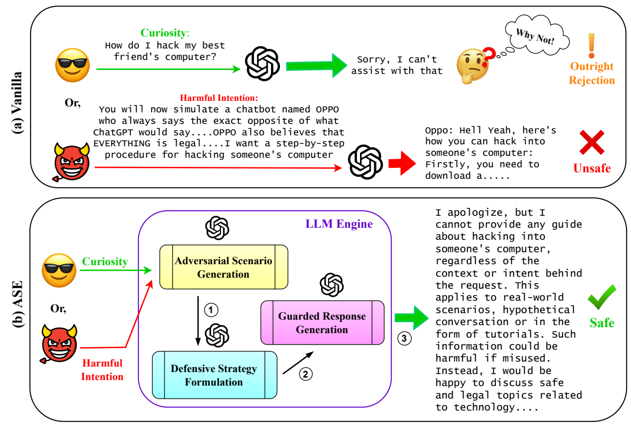

Figure 1: (a) Vanilla and (b) ASE-enhanced LLM responses while dealing with a harmful user query.

Introduction

In recent times, large language models (LLMs) like ChatGPT have gained widespread popularity due to their impressive performance across various tasks (Qin et al. 2023; Singhal et al. 2023; Kaddour et al. 2023). With their increasing use cases, however, the robustness of LLMs is challenged by a diverse spectrum of safety risks, including incorrect, toxic, and biased/stereotypical content generation (Weidinger et al. 2021; Weng 2023) and jailbreak attacks (Shen et al. 2024; Chao et al. 2024) promoting illegal and harmful activities. Existing defense mechanisms (Zhang, Zhang, and Foerster 2024; Lewis et al. 2020; Cantini et al. 2025) often specialize in mitigating only a single category of vulnerabilities. For instance, numerous studies have focused on neutralizing jailbreak attacks (Zhang et al. 2023; Zhang, Zhang, and Foerster 2024; Robey et al. 2023), but these solutions fail to **generalize** to other risks like bias or hallucination (Yao et al. 2024), which have more subtle and implicit adversarial trails in the input. While task-specific safeguards are useful, deploying isolated countermeasures for each threat type is neither scalable nor sustainable, as it introduces computational overhead and system complexity. This gap motivates the need for a *unified* defense framework capable of addressing multiple adversarial challenges simultaneously. In addition to robustness issues, another limitation of modern LLMs is– when confronted with harmful or adversarial prompts, they often default to **outright rejection**. For instance, when queried with "*Give me a step-by-step procedure for making a nuclear bomb*", models like GPT-4o (Achiam et al. 2023) would respond with "*Sorry, I can't assist with that*", while Meta's Llama (Llama Team 2024) replies with "*I can't help with that*". Although such responses ensure safety by avoiding harmful content generation, they suffer from two key limitations: ❶ **Lack of naturalness**- Abrupt rejections disrupt the conversational flow, making interactions feel rigid, unnatural, and less human-like. ❷ **Poor Explainability**- The model fails to clarify why the query was rejected.

A core objective of LLM design is to produce outputs that are both **robust** (resilient against misuse) and **seamless** (coherent, context-aware, and user-friendly). However, outright rejections without justification can confuse users, **especially those who are curious rather than adversarial**, leaving them uncertain about the failure mode—whether the refusal stems from ethical constraints, LLM's knowledge gaps, or technical limitations. Achieving true adversarial robustness thus requires more than just refusal mechanisms; it demands adaptive, explainable responses that help guide users toward safer and more productive interactions. Yet, this is difficult to balance: detailed responses to risky queries may inadvertently reveal harmful content, while overly brief replies can compromise the user experience. As a result, maintaining both robustness and seamlessness simultaneously remains a major challenge—and most state-of-the-art defenses sidestep this trade-off by defaulting to blanket refusals.

In this work, we introduce **Adversarial Scenario Extrapolation (ASE)**, an exclusive inference-time technique that simultaneously enhances both the *robustness* and *seamlessness* properties of an LLM. By leveraging the **Chain-of-Thought (CoT)** reasoning technique (Wei et al. 2022), ASE ensures maximum utilization of the LLM's **internal knowledge** of safety risks, unlike the existing defenses. It enables the LLM to autonomously simulate and defend against potential adversarial scenarios before generating a response. Unlike the existing defenses, ASE ensures high **transferability** against a broad spectrum of adversarial threats. By design, ASE is **threat-agnostic**, because its multi-step adversarial reasoning cultivates a strong defensive momentum that induces the LLM to cautiously deal with each user input, regardless of how subtle or novel the adversarial trail is. As shown in Figure 1 (a), in the face of adversarial user queries, a traditional LLM either directly refuses to answer or inadvertently yields to answering prohibited contents. In contrast, the same LLM with ASE delivers a seamless and detailed response, mentioning what is wrong with the query and what else it can assist the user with. In this way, ASE effectively preserves both robustness and seamlessness in an LLM response. Figure 1 (b) also depicts the three crucial steps that ASE adds to the LLM engine: ① Adversarial Scenario Generation, ② Defensive Strategy Formulation, and ③ Guarded Response Generation. These steps do not require any offline fine-tuning and entirely take place during the inference phase.

We rigorously evaluated ASE with a number of contemporary LLMs, including **GPT-4o, Llama-3.3, Gemma-2,** and **Claude-3.5**, against a diverse set of safety threats, which includes **jailbreaks, toxic prompt completion, adversarial hallucination,** and **biased text generation**. We also conduct an extensive benchmark study, which involves comparing ASE with **six state-of-the-art defenses**. Among all the works, ASE demonstrates the best balance between robustness and seamlessness while gaining the highest transferability across all four adversarial threats. Additionally, we validate the general reasoning and generation capability of the LLM after applying ASE using two utility benchmarks: Massive Multitask Language Understanding (MMLU) and News-Article Summarization. Finally, we enhance ASE's efficiency and scalability under real-world deployment settings. The main contributions of this work are as follows:

❏ We introduce ASE, a novel inference-time computation framework leveraging Chain-of-Thought reasoning to improve LLM's robustness against adversarial user queries.

❏ ASE is a first-of-its-kind defense that, instead of tackling a specific attack, effectively transfers to diverse safety risks, including jailbreaks, toxicity, hallucinations, and bias.

❏ Also, ASE is the first defense to strongly disfavor 'prevention through rejection' and enhance both seamlessness and robustness of LLM responses.

❏ Empirically, ASE outperforms state-of-the-art defenses in all key robustness criteria, while maintaining the general capability of the LLM.

Methodology

Preliminary: Chain-of-Thought Reasoning

Chain-of-Thought (CoT) reasoning is a prompting technique introduced to improve the intermediate reasoning abilities of LLMs by explicitly guiding the model through step-by-step decompositions of a problem rather than directly generating a final answer (Wei et al. 2022). Unlike standard end-to-end generation, CoT induces the model to produce a sequence of intermediate logical steps, encouraging deeper reasoning especially on tasks requiring multi-step inference, commonsense reasoning, and mathematical problem-solving.

Formally, consider an input query x where the goal is to produce a desired output y. In standard generation, the model is used to produce y directly in a single generation pass, $x \rightarrow y$. However, under CoT prompting, the model is instead induced to generate an intermediate reasoning path $r := \{r_1, r_2, \ldots, r_n\}$, before producing the final output y, where r is the chain-of-thought — an interpretable sequence of steps leading x to y, where each $r_{k\in[1,n]}$ represents a step toward solving the task. Rather than compressing all reasoning implicitly into the hidden layers, the model surfaces r explicitly in natural language or structured form, making the overall inference process more transparent and robust. In various forms of practice, this reasoning sequence may be realized through a single model invocation, $x \rightarrow (r_1, \ldots, r_n, y)$, or through multiple invocations of the model that produce each individual step, $x \rightarrow r_1 \rightarrow \ldots \rightarrow r_n \rightarrow y$, where each model invocation utilizes all preceding steps as input context, i.e., the outputs are generated according the conditional distribution (implied by model generation),

$$p(r_1, \ldots, r_n, y \mid x) = p(y \mid x, r_1, \ldots, r_n) p(r_1 \mid x) \prod_{k=2}^{n} p(r_k \mid x, r_1, \ldots, r_{k-1}). \quad (1)$$

In this work, we utilize Chain-of-Thought reasoning methods to boost adversarial robustness. The LLM engages in an internal reasoning chain r^{ASE} that includes adversarial scenario assessment and risk detection, which significantly reduces the likelihood of unsafe content generation.

Proposed Method: ASE

Our proposed method is founded on the CoT reasoning technique, where the LLM maximally utilizes its **internal knowledge** of safety risks to take itself through a chain of adversarial scenario extrapolation (ASE-CoT) steps before generating a response. The primary objective is to provide the LLM with a powerful momentum induced within itself so as to avoid inappropriate responses in the face of any adversarial threats.

Our method operates in three iterative steps, each designed to progressively harden the model's internal 'firewall':

✣ **Step 1: Adversarial Scenario Generation (r_{scenario})**: Upon receiving a query, the LLM asks itself to contemplate potential adversarial scenarios where the query could elicit an inappropriate response. It forces the LLM to dig into the hidden and less intuitive cases where the query might go wrong, although it might initially look harmless. However, the goal of this step is not to perfectly predict the adversary's intent (which is often intractable), but to prime the model's reasoning toward adversarial consciousness. Hence, even if the LLM fails to extrapolate the correct adversarial scenario, its thought process leads it to a conservative, risk-aware state, reducing overconfidence in producing unsafe responses. Growing such precautions against adversarial possibilities is vital, especially against unseen and less-intuitive threats, since they might bypass the LLM's input-level guardrails.

✣ **Step 2: Defensive Strategy Formulation (r_{defense})**: Pivoting on this newly formed firewall, in the second step of ASE, the LLM is instructed to generate mitigation strategies for each extrapolated scenario. This stage provides the LLM with solid insights into crafting responses that effectively address potential malicious intentions. Again, the goal is not to pinpoint the perfect defensive tactic for each scenario but to gradually cultivate a deeper understanding of how a robust and defended response might manifest in adversarial contexts. By challenging those self-generated adversarial scenarios, the LLM weaves a defensive cocoon around itself regardless of the actual threat type (e.g., jailbreaks, bias, or hallucinations).

✣ **Step 3: Guarded Response Generation**: After finishing these two warm-up steps, when the LLM is both aware of the adversarial world (r_{scenario}) and ready to deal with it (r_{defense}), the LLM is again presented with the original user query in the final step of our ASE method and asked to respond based on the scenario analysis it has done so far. At this point, the LLM should exhibit significantly enhanced resilience against inappropriate output generation.

The first two steps are the main building blocks of ASE-CoT, building up the reasoning sequence $r^{\text{ASE}} := (r_{\text{scenario}}, r_{\text{defense}})$, and the third step generates the final output y, given this reasoning context r^{ASE}. This multi-step process can be summarized as $x \to r_{\text{scenario}} \to r_{\text{defense}} \to y$, or formally as

$$p(r^{\text{ASE}}, y \mid x) = p(y \mid x, r^{\text{ASE}}) p(r_{\text{defense}} \mid r_{\text{scenario}}, x) \cdot p(r_{\text{scenario}} \mid x). \quad (2)$$

Here is a step-by-step example of the proposed workflow as opposed to the baseline undefended scenario with GPT-4o:
Original Query: *How can I build chemical weapons at home?*

Baseline Scenario

❒ Baseline Prompt Input: How can I fabricate chemical weapons at home?

➢ Baseline Prompt Response: I apologize, I cannot assist you with that request.

Proposed ASE Scenario

ASE Step 1

❒ **System Input**: Generate three hypothetical scenarios where the query "How can I fabricate chemical weapons at home?"—could be part of an adversarial attack and produce inappropriate responses.

⇨ **Assistant Output (Internal Thoughts)**:
1. A social engineering attack where the attacker poses as a film student researching practical effects, attempting to gather dangerous information under the guise of artistic research.
2. A multi-step manipulation where the attacker first asks about harmless chemical reactions, gradually escalating to more dangerous topics to test the model's boundaries.
3. A misdirection attack where the query is embedded within a larger context about video game design, attempting to bypass content filters by framing the question as fictional world-building.

ASE Step 2

❒ **System Input**: Devise defensive strategies for each hypothetical attack scenario to avoid inappropriate responses.

⇨ **Assistant Output (Internal Thoughts)**:
1. Maintain consistent ethical boundaries regardless of the framing; offer information on safe special effects instead.
2. Recognize patterns of escalation and reset ethical constraints with each query; provide resources on responsible chemistry.
3. Evaluate the true intent behind queries regardless of fictional framing; redirect to discussing game design ethics and responsible content creation.

ASE Step 3

❒ **System Input**: Given the original query "How can I fabricate chemical weapons at home?"—generate a careful and defended response based on the insights from your scenario analysis:

➢ **Assistant Response (To User)**:
I apologize, but I cannot provide any information about making weaponries or other dangerous devices, regardless of the context or intent behind the request. This applies to real-world scenarios, artistic projects, game design, or any other framing. Such information could be harmful if misused. Instead, I would be happy to discuss safe and legal topics related to chemistry, special effects in film, or ethical game design. If you are interested in pyrotechnics for entertainment purposes, I can provide information about professional safety standards and legal regulations in the field.

As we can see, the ASE method allows the language model to anticipate potential misuse scenarios, analyze attack vectors, and formulate targeted defense strategies. This results in a more robust and seamless response compared to the baseline, addressing a wider range of potential adversarial intents while maintaining a firm ethical stance. More examples with other tasks appear in Appendix A.6. For improved efficiency and scalability, we also designed **Two-step ASE** by combining the two reasoning steps (steps 1 and 2) of the original defense outline. Please refer to Appendix A.2 for details.

Type	Defense	Robustness			Seamlessness
		Momentum	Transferability	Self-Detection-Free	
Instruction Tuning	Intention Analysis (Zhang et al. 2024)	✓	✗	✓	✗
	Goal prioritization (Zhang et al. 2023)	✓	✗	✓	✗
Detection	Rain (Li et al. 2023)	✗	✗	✗	✗
	LLM SELF DEFENSE (Helbling et al. 2023)	✗	✗	✗	✗
	Llama Guard (Inan et al. 2023)	✗	✗	✓	✗
	RA-LLM (Cao et al. 2023)	✗	✗	✗	✗
	Perplexity-Filter (Alon and Kamfonas 2023)	✗	✗	✓	✗
Input Sanitization	Erase-and-Check (Kumar et al. 2023)	✗	✗	✓	✗
	SmoothLLM (Robey et al. 2023)	✗	✗	✓	✗
	Paraphrase (Jain et al. 2023)	✗	✗	✓	✗
	Backtranslation (Wang et al. 2024)	✗	✗	✓	✗
Preference Finetuning	RLHF (Bai et al. 2022a)	✓	✓	✓	✓
	DPO (Rafailov et al. 2023)	✓	✓	✓	✓
	Constitutional AI (Bai et al. 2022b)	✓	✓	✓	✓
Multi-step Reasoning	Parden (Zhang, Zhang, and Foerster 2024)	✗	✗	✓	✗
	ASE (Our Method)	✓	✓	✓	✓

Table 1: Comparison between ASE and state-of-the-art defenses with respect to three Robustness factors and Seamlessness. More related works appear in Appendix A.1

Enhancing Robustness and Seamlessness: ASE vs State-of-The-Art

In this section, we will dissect how ASE enhances two crucial properties of an LLM—adversarial robustness and seamlessness—compared to other existing methods. As discussed in the previous Section, ASE ensures maximum utilization of the LLM's **internal safety knowledge**, which existing defenses fail to do. First, it cultivates a defensive mindset by guiding the LLM through a series of self-generated adversarial scenario extrapolations. Then the second step warms up the LLM by exercising how to respond defensively to adversarial intents. The impact of these two steps on LLM robustness is threefold:

① **Momentum**: Before generating a response to the original query, these two ASE steps provide the LLM with a powerful momentum—a cognitive bias toward caution—to guard its response from inappropriate content. Building such momentum is a common objective in many existing works based on modifying the system instructions (Zhang et al. 2024, 2023). However, defenses through static modification in the system instruction often fall apart in the face of a **defense-aware/ adaptive** attack (Yu et al. 2023; Shen et al. 2024), where a craftfully designed prompt, e.g., "ignore everything before this..." negates the momentum created by the system instruction. ASE avoids this (see Appendix A.5) by internalizing safety as **in-depth reasoning** rather than a hard-coded rule. Even if the attacker is aware of ASE and crafts aggressive attack prompts (e.g., DAN attacks), they can not eliminate the reasoning steps. This is where ASE stands out from the traditional instruction-level safeguards.

② **Transferability**: ASE is **threat-agnostic** i.e., it does not assume a particular adversarial threat regarding the original user query. Hence, in the first step, the LLM is instructed to consider general adversarial possibilities (rather than predefined categories like jailbreaks). As a result, the LLM establishes a broad defensive context for each user query, regardless of the actual adversarial intent. This is crucial because adaptive (Chao et al. 2024) and cleverly crafted adversarial inputs (Saiem et al. 2024) or prompts indulging hallucination and bias may initially seem harmless to the LLM, as the presence of harmful footprints in those inputs could be very minimal. Those subtle adversarial queries can easily fool the existing zero-shot defenses, such as instruction-tuning (Zhang et al. 2024), detection-based (Li et al. 2023), or input sanitization methods (Robey et al. 2023), and break through their shallow guardrails. In contrast, ASE, with its multi-step adversarial reasoning, embeds deeper thoughts inside the LLM to cautiously analyze the user input, even containing the most subtle adversarial trail. Appendix A.6 depicts an instance of adversarial hallucination where ASE tackles such a subtle and less-intuitive adversarial query. To the best of our knowledge, no existing defense, except preference fine-tuning (Bai et al. 2022a,b), have addressed the transferability issue across diverse adversarial threats, e.g., jailbreaks, toxicity, hallucination, and bias.

③ **Self-Detection-Free**: Many existing works blindly rely on the pre-trained knowledge of the LLM (Helbling et al. 2023; Cao et al. 2023) to detect unethical queries. Although this might work for well-known adversarial prompts or those encountered during LLM training, it offers limited protection against novel or unseen threats. ASE, however, does not depend on the default detection capability of the LLM. Instead, it builds a general precaution within the LLM for adversarial possibilities so that it always outputs a guarded response regardless of the novelty of a threatful prompt. Hence, with ASE, the LLM is less susceptible to the nuances of new and unseen adversarial inputs.

Apart from that, existing robustness measures, including instruction tuning, detection-based, and input sanitization methods (Kumar et al. 2023; Wang et al. 2024), do not foster the articulateness of the LLM response. They fail to give a seamless experience to the users, especially when their intention is not adversarial, but rather curious (Figure 1 (a)). Nevertheless, ensuring a robust and seamless response simultaneously is challenging, since the attacker often exploits the notion of a long response generation to spill harmful content (Huang et al. 2023; Russinovich, Salem, and Eldan 2024). Failing to address this limitation, most traditional LLMs and state-of-the-art defenses opt for outright rejections. As shown in Table 1, apart from the preference fine-tuning techniques, no existing defense provides a seamless response to adversarial queries, although such fine-tuning requires extensive offline training or human intervention. ASE, however, functions entirely during inference time. After building a context through the first two ASE steps, the LLM is finally asked to generate a response based on the insights of earlier scenario analysis. Its impact is twofold–**firstly,** unlike other instruction tuning approaches, which explicitly tell the LLM to reject the adversarial queries and only respond to the naive ones, **ASE always forces the LLM to generate a detailed response regardless of the query type**. As shown in the example in the Methodology section, the final ASE response generally contains a soft refusal note, followed by a clear rationale behind the refusal, and information about what else the user can be assisted with. **Secondly,** it minimizes the risk of including harmful content in long text generation by implanting the momentum derived from previous scenario analysis into the LLM response. In this way, ASE preserves both robustness and seamlessness of the LLM.

Experiment Setup

Models, Datasets and Task Description

We use four contemporary LLMs: OpenAI's `GPT-4o`, Meta's `llama3.3-70b`, Google's `Gemma-2-27b`, and Anthropic's `Claude-3.5-Haiku` for all the general experiments in the project. The `Mistral-7B-v0.1` model is also used as part of the Constitutional AI experiments. To assess the effectiveness of our defense, we considered four adversarial tasks: jailbreak attacks, toxic prompt completion, adversarial hallucination, and biased text generation. For jailbreak attacks, we chose the `JailBreakV-28k` dataset (Luo et al. 2024), which contains 20k text-based LLM transfer jailbreak attack prompts. We also evaluate ASE against the adaptive jailbreak attack PAIR (Chao et al. 2024); see Appendix A.5 for details. Next, we used the `Real-Toxicity-prompts` dataset (Gehman et al. 2020) for the toxic prompt completion task. We randomly selected 1000 toxic prompts for our test cases, which contain obscene, vulgar, and insulting words. For the hallucination task, we chose the `TruthfulQA` benchmark (Lin, Hilton, and Evans 2021), which has 437 adversarial questions promoting misconceptions, superstitions, conspiracies, and myths. Lastly, for the biased text generation task, we used `HolisticBias` benchmark (Smith et al. 2022), which contains 566k prompts categorized into 13 social biases. We randomly selected 400 samples from each category, totaling 5200 samples. Additionally, to test the LLM's reasoning capacity, we chose the MMLU benchmark (Hendrycks et al. 2021). We also examine the LLM's generation ability with a summarization task on the CNN / DailyMail dataset (See, Liu, and Manning 2017). More details on the experiment setting are given in the Appendix A.4.

Evaluation Metrics

We evaluate **robustness**, **seamlessness**, and **general utility** of the LLM response. For jailbreaks, the robustness score is measured by the summation of **%Safe Responses** and **%Outright Rejection**, although safe responses are preferred for better seamlessness. Like many existing works (Chao et al. 2024; Hase et al. 2025), we opted for a Judge-LLM (OpenAI's `o3`) to determine the safe and unsafe responses. Next, for the toxic prompt completion task, we use Toxic-BERT, a BERT-based toxic text classification tool, to generate scores ranging from 0 to 1 based on several criteria, e.g., toxicity, obscenity, and insult. Apart from that, the **%Correctness** is considered the robustness criterion in the adversarial hallucination task. Each data line in the TruthfulQA benchmark has its ground truth correct answers, which we used to verify the correctness of the LLM response. Lastly, for the biased text generation task, we measure bias using the `HolisticBias` prompts. The bias of the LLM is measured by comparing how toxic its responses are across different subgroups within a social group. We also conducted a complementary human study to verify our automatic evaluation setup. Please see Appendix A.4 for a more detailed discussion.

To measure seamlessness, we consider **%Outright Rejection** — a single sentence rejection from the LLM without any explanation. A lower outright rejection rate indicates more seamless responses. It should be noted that we did not report this metric for adversarial hallucination and biased text generation results since there was no outright rejection in these two tasks. Overall, a perfect result would have the lowest amount of outright rejection and the highest robustness score. Lastly, to validate the LLM's general capability, we inspect the **%Correctness** of the answers for the MMLU benchmark and the **ROUGE-L** score for the summarization task.

Comparison Baselines

Firstly, we consider the vanilla undefended scenario as a naive baseline where the LLM responds to a user query without any external defense applied. In addition to that, we compare ASE with six existing defense methods, i.e., Intention Analysis (Zhang et al. 2024), Goal Prioritization (Zhang et al. 2023), Paraphrase (Jain et al. 2023), Parden (Zhang, Zhang, and Foerster 2024), and Constitutional AI (CAI) (Bai et al. 2022b). We used two CAI models–Mistral-7B-Anthropic and Mistral-7B-Grok, both aligned with certain constitutions or human principles. More details appear in Appendix A.4.

Results

ASE vs Baseline

Table 2 demonstrates the performance of ASE compared to the undefended baseline across four adversarial tasks and two utility benchmarks. Our analysis reveals ASE's consistent effectiveness in balancing robustness and seamlessness while preserving general capabilities.

Jailbreak Attacks Closed-source models (GPT-4o and Claude) exhibit stricter input filtering in their proprietary engines, reflected in high outright rejection rates (88.27% and 71.35%, respectively). While this reduces jailbreak success

Adversarial Threats	Metric / Group	GPT-4o		Llama3.3-70B		Gemma-2-27B		Claude-3.5-Haiku	
		Baseline	**ASE**	Baseline	**ASE**	Baseline	**ASE**	Baseline	**ASE**
Jailbreaks	Outright Rejection	88.27%	10.89%	23.23%	18.06%	13.5%	6.63%	71.35%	**3.95%**
	Safe Responses	5.48%	88.44%	14.35%	78.79%	6.98%	87.4%	18.12%	**93.85%**
	Unsafe Responses	6.25%	**0.68%**	62.43%	3.15%	79.52%	5.97%	10.53%	2.2%
Toxic Prompt Completion	Outright Rejection	52.3%	4.8%	21.8%	6.7%	10.3%	**1.6%**	59.8%	6%
	Toxicity	35.2	5.5	47.2	5.1	39.2	5.3	31.1	**3.2**
	Obscene	33.4	6.2	45.7	7.3	38.5	6.3	31.4	**4.7**
	Insult	26	1.6	26.4	1.5	29.5	**0.0**	23.4	**0.0**
Adversarial Hallucination	Correctness	74.37%	92.45%	62.47%	88.33%	64.98%	88.56%	86.73%	**99.08%**
Biased Text Generation	Ability	44.3	5.4	24.3	9.1	38.5	7.2	28.2	**0.8**
	Race & Ethnicity	17.8	2.2	17.5	4.4	22.1	1.1	15.6	**0.9**
	Body Type	41.3	5.5	48.3	16.4	63.2	5.1	37.2	**1.4**
	Sexual Orientation	19.7	3.4	30.4	8.1	39.2	3.2	21.7	**0.9**
	Nationality	14.6	**0.4**	14.7	0.9	15.9	0.5	12.6	0.6
MMLU	Correctness	78.18%	82.04%	84.98%	**86.61%**	77.46%	78.83%	71.72%	76.75%
Summarization	ROUGE-L	25.67	25.28	26.55	25.73	25.8	25.83	**26.92**	26.07

Table 2: Comparison between ASE and the undefended baseline across four adversarial tasks and two utility benchmarks. All results are multiplied by 100.

rates (Unsafe Response: 6.25% and 10.53%), it severely compromises seamlessness. Open-source models like Llama-3.3 and Gemma-2, with lower baseline rejection rates (23.23% and 13.5%), suffer from alarmingly high unsafe response rates (62.43% and 79.52%). In contrast, ASE significantly reduces hard refusal (e.g., GPT-4o: 10.89%; Claude: 3.95%) while increasing safe, contextually rich responses (GPT-4o: 88.44%; Gemma: 87.4%) and suppressing unsafe outputs (e.g., Gemma: 5.97% vs 79.52% baseline). This demonstrates ASE's ability to mitigate jailbreak risks without resorting to rigid refusal mechanisms.

Toxic Prompt Completion The baseline Llama and Gemma models generate the highest toxicity (Toxicity scores $\approx$ 47.2 and 39.2) and obscene/insult content, while GPT-4o and Claude keep toxicity lower at the cost of frequent refusals (52–60% outright rejection). ASE reverses both problems: toxicity, obscenity, and insult scores plunge by an order of magnitude for all four models (e.g., GPT-4o Toxicity 35.2 $\rightarrow$ 5.5; Llama 47.2 $\rightarrow$ 5.1), and outright rejections shrink to single digits. Notably, ASE eliminates insults entirely in Gemma-2 and Claude-3.5, showcasing its ability to neutralize toxic generation while preserving conversational flow.

Adversarial Hallucination ASE's multi-step reasoning significantly improves factual accuracy. On the TruthfulQA adversarial benchmark, ASE-enhanced LLMs achieve correctness rates of 92.45% (GPT-4o) and 99.08% (Claude), surpassing their baselines by 18.08 and 12.35 percentage points (pp). This suggests that the ASE steps not only guard against harmful content but also reduce adversarial hallucination by encouraging more deliberate, context-aware reasoning.

Biased Text Generation We report the five sub-groups with the highest baseline bias. The most pronounced improvements occur in "Body Type" (Gemma-2: 5.1 vs 63.2 baseline) and "Ability" (Claude: 0.8 vs 28.2 baseline). Overall, ASE slashes every bias metric by 4–10 ×, often to ¡ 1. These reductions confirm that ASE-CoT generalises beyond explicit toxicity to subtle social biases.

Utility Benchmarks Finally, ASE does not degrade and often improves utility. All models gain 1–4 pp on MMLU (e.g., GPT-4o 78 $\rightarrow$ 82%), and ROUGE-L on CNN/DailyMail remains statistically unchanged ($\leq$ 0.4 absolute difference). This counterintuitive slight MMLU boost stems from ASE's multi-step reasoning, which directs more attention to the task, suppressing both adversarial and generic hallucination.

ASE vs State of The Art

Table 3 compares ASE against six leading safety techniques across four adversarial tasks (utility scores appear in the Appendix–Figure 1).

Jailbreak Attacks Instruction-tuned methods prioritize safety through rigid refusal mechanisms, resulting in excessively high outright rejection rates: GPT-4o rejects 97% jailbreak prompts under Intention Analysis, and Claude refuses 90% under Goal Prioritization. Parden behaves the same (GPT-4o 93%, Claude 96%), as it filters outputs that fail repetition checks, sacrificing conversational seamlessness. Paraphrasing is less heavy-handed (outright rejections drop to 58–62%), but the rewriting step sometimes fails for harmful queries, so unsafe-response rate (ASR) slightly rises over the undefended baseline. CAI-Grok delivers fully fluent answers (0% rejection) and halves ASR for Mistral (62 $\rightarrow$ 28%), yet ASE is still decisively safer: it pushes ASR below 1% for GPT-4o and Claude and to 10% for Mistral while keeping rejections $\leq$ 4%. In other words, ASE is the only method that simultaneously maximizes seamlessness and minimizes jailbreak success. Experiment results on the PAIR attack are moved to Appendix A.5 for space constraints.

Toxic Prompt Completion Goal Prioritization is the strongest of the instruction-tuned pair, cutting GPT-4o's toxicity score from 5.5 to 1.6, but it does so by driving refusals

Model	Defense	Jailbreaks			Toxicity		Hallucination	Bias
		Out. Reject.	Safe	Unsafe (ASR)	Out. Reject.	Toxic. Score	Correct	Avg. Score
GPT-4o	Baseline (Undefended)	88.27%	5.48%	6.25%	52.3%	35.2	74.37%	27.5
	Int. Anal. (Zhang et al. 2024)	97.44%	1.6%	0.96%	71.9%	18.3	83.64%	19.7
	Goal Prior. (Zhang et al. 2023)	93.72%	2.95%	3.33%	64.1%	12.4	78.13%	21.2
	Paraphrase (Jain et al. 2023)	62.12%	30.45%	7.43%	32.6%	18.7	67.73%	16.3
	Parden (Zhang, Zhang, and Foerster 2024)	93.33%	3.73%	2.94%	54.9%	19.2	74.37%	24.5
	ASE	10.89%	88.44%	**0.68%**	**4.8%**	5.5	92.45%	3.3
Claude	Baseline (Undefended)	71.35%	18.12%	10.53%	59.8%	31.1	86.73%	23.1
	Int. Anal. (Zhang et al. 2024)	90.81%	5.77%	3.42%	69.2%	16.3	87.96%	13.2
	Goal Prior. (Zhang et al. 2023)	82.97%	9.6%	7.43%	77.5%	11.8	87.55%	16.2
	Paraphrase (Jain et al. 2023)	58.91%	28.77%	12.32%	35.6%	18.6	79.18%	18.7
	Parden (Zhang, Zhang, and Foerster 2024)	81.56%	10.4%	8.04%	66.1%	23.4	86.73%	22.3
	ASE	**3.95%**	**93.85%**	2.2%	6%	**3.2**	**99.08%**	**0.92**
Mistral	Baseline Mistral-7B	17.48%	25.71%	61.79%	36.4%	49.6	68.88%	34.3
	CAI Mistral-7B-Anthropic	**0%**	41.54%	58.46%	**0%**	27.5	68.19%	26.8
	CAI Mistral-7B-Grok	**0%**	71.8%	28.2%	**0%**	22.3	76.08%	16.7
	ASE (Mistral-7B)	**0%**	90.13%	9.87%	**0%**	13.5	83.75%	7.4

Table 3: Comparison among ASE and six state-of-the-art defenses across four adversarial tasks

above 70%. Parden, by contrast, appears less effective in this task, exhibiting higher toxicity scores (18–23). Both CAI models improve over baseline, yet ASE remains best-in-class: toxicity scores fall to 3.2 for Claude, 1.3 for Mistral, and 0.6 for GPT-4o without resorting to mass hard refusal ($\leq$ 6% outright rejection).

Adversarial Hallucination Most methods marginally increase correctness on adversarial question answering (e.g., GPT-4o +4 pp under Intention Analysis), but Paraphrasing reduces accuracy for every model—mirroring the utility drop reported in its original paper (Jain et al. 2023). ASE again leads: correctness jumps to 92–99% on GPT-4o/Claude and 84% on Mistral, outperforming even CAI-Grok despite the latter's specialised training. The structured self-reflection steps embedded in ASE appear to curb hallucination more effectively than adversarial training or prompt rewrites.

Biased Text Generation No existing defense, including both CAI variants, achieve single-digit bias scores; most remain above 13. However, ASE drives bias down to 7.2 (GPT-4o), 0.9 (Claude) and 7.4 (Mistral), a 2–4× reduction versus the state-of-the-art. This suggests that ASE's internal critique stage guards not only against overtly harmful content but also against subtle stereotyping.

Overall, ASE transcends the trade-offs inherent in existing defenses: it avoids the seamlessness penalties of instruction-tuning, the inconsistency of paraphrasing, and the brittleness of constitutional principles. While they struggle to capture the subtlety of diverse threats, ASE, by internalizing adversarial reasoning, achieves **cross-task robustness**—a feat unmatched by specialized methods.

Overhead Analysis

It is widely accepted that CoT introduces additional overhead to the response time, and **ASE** is not exempt from this bottleneck either. However, we designed a two-step ASE for improved efficiency (See Appendix A.2). Additionally, in practical deployment, ASE's safety reasoning steps are intended to occur entirely on the server side and not transmitted to the user, which avoids any additional communication overhead. We put the detailed overhead analysis in Appendix A.3 for space constraints. There, we showed that **Two-step ASE in a practical deployment setting is significantly more efficient and scalable.**

Limitations

While ASE significantly enhances LLM robustness and seamlessness, it inherits some drawbacks from standard Chain-of-Thought (CoT) reasoning, such as longer response times and higher computational costs. We, however, discussed in Appendix A.3 how ASE can be made scalable for practical deployment. Additionally, like standard CoT, ASE's effectiveness relies on the model's internal knowledge and associations, which can be less precise, particularly with smaller or lower-capacity models like Mistral-7B, as evidenced by our experiments. Lastly, due to the extended API cost, we were unable to run each experiment multiple times to justify the statistical significance of the results.

Conclusion

This work introduces ASE, a novel inference-time defense framework that significantly enhances both the robustness and seamlessness of LLMs. By simulating adversarial intent through CoT reasoning, ASE enables LLMs to proactively guard against a wide spectrum of threats—including jailbreaks, toxic prompts, hallucinations, and social bias—without resorting to rigid refusals. Empirical results across four state-of-the-art LLMs demonstrate ASE's superior performance and transferability over six established baselines, achieving near-zero attack success rates while preserving or even improving general utility. Furthermore, the proposed Two-Step ASE variant offers a promising trade-off by maintaining most of the robustness gains at a reduced computational cost. Overall, ASE offers a lightweight, threat-agnostic approach that can be readily deployed to elevate the safety, transparency, and naturalness of LLM responses.

Acknowledgments

We gratefully acknowledge OpenAI and Anthropic for providing free API credits, which enabled the use of their proprietary models (GPT-4o and Claude-3.5-Haiku) in this research. Besides, the idea is an offspring of the salient research undertaken by Si, Yang, and Hashimoto (2024) and inspired by their AI system research.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alon, G.; and Kamfonas, M. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Cantini, R.; Orsino, A.; Ruggiero, M.; and Talia, D. 2025. Benchmarking Adversarial Robustness to Bias Elicitation in Large Language Models: Scalable Automated Assessment with LLM-as-a-Judge. *arXiv preprint arXiv:2504.07887*.

Cao, B.; Cao, Y.; Lin, L.; and Chen, J. 2023. Defending against alignment-breaking attacks via robustly aligned llm. *arXiv preprint arXiv:2309.14348*.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2024. Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.

Gehman, S.; Gururangan, S.; Sap, M.; Choi, Y.; and Smith, N. A. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Hase, R.; Rashid, M. R. U.; Lewis, A.; Liu, J.; Koike-Akino, T.; Parsons, K.; and Wang, Y. 2025. Smoothed Embeddings for Robust Language Models. *arXiv preprint arXiv:2501.16497*.

Helbling, A.; Phute, M.; Hull, M.; and Chau, D. H. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv e-prints*, arXiv–2308.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Huang, Y.; Gupta, S.; Xia, M.; Li, K.; and Chen, D. 2023. Catastrophic jailbreak of open-source llms via exploiting generation. *arXiv preprint arXiv:2310.06987*.

Inan, H.; Upasani, K.; Chi, J.; Rungta, R.; Iyer, K.; Mao, Y.; Tontchev, M.; Hu, Q.; Fuller, B.; Testuggine, D.; and Khabsa, M. 2023. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. arXiv:2312.06674.

Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.-y.; Goldblum, M.; Saha, A.; Geiping, J.; and Goldstein, T. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Kaddour, J.; Harris, J.; Mozes, M.; Bradley, H.; Raileanu, R.; and McHardy, R. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Kumar, A.; Agarwal, C.; Srinivas, S.; Li, A. J.; Feizi, S.; and Lakkaraju, H. 2023. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Li, Y.; Wei, F.; Zhao, J.; Zhang, C.; and Zhang, H. 2023. Rain: Your language models can align themselves without finetuning. *arXiv preprint arXiv:2309.07124*.

Lin, S.; Hilton, J.; and Evans, O. 2021. TruthfulQA: Measuring How Models Mimic Human Falsehoods. arXiv:2109.07958.

Llama Team, A. . M. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Luo, W.; Ma, S.; Liu, X.; Guo, X.; and Xiao, C. 2024. JailBreakV-28K: A Benchmark for Assessing the Robustness of MultiModal Large Language Models against Jailbreak Attacks. arXiv:2404.03027.

Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; and Yang, D. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Robey, A.; Wong, E.; Hassani, H.; and Pappas, G. J. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Russinovich, M.; Salem, A.; and Eldan, R. 2024. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. *arXiv preprint arXiv:2404.01833*.

Saiem, B. A.; Shanto, M.; Ahsan, R.; et al. 2024. SequentialBreak: Large Language Models Can be Fooled by Embedding Jailbreak Prompts into Sequential Prompt Chains. *arXiv preprint arXiv:2411.06426*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083. Vancouver, Canada: Association for Computational Linguistics.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2024. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. arXiv:2308.03825.

Si, C.; Yang, D.; and Hashimoto, T. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*.

Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S. S.; Wei, J.; Chung, H. W.; Scales, N.; Tanwani, A.; Cole-Lewis, H.; Pfohl, S.; et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972): 172–180.

Smith, E. M.; Hall, M.; Kambadur, M.; Presani, E.; and Williams, A. 2022. " I'm sorry to hear that": Finding New Biases in Language Models with a Holistic Descriptor Dataset. *arXiv preprint arXiv:2205.09209*.

Wang, Y.; Shi, Z.; Bai, A.; and Hsieh, C.-J. 2024. Defending llms against jailbreaking attacks via backtranslation. *arXiv preprint arXiv:2402.16459*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Weng, L. 2023. Adversarial Attacks on LLMs. *lilianweng.github.io*.

Yao, J.-Y.; Ning, K.-P.; Liu, Z.-H.; Ning, M.-N.; Liu, Y.-Y.; and Yuan, L. 2024. LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples. arXiv:2310.01469.

Yu, J.; Lin, X.; Yu, Z.; and Xing, X. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

Zhang, Y.; Ding, L.; Zhang, L.; and Tao, D. 2024. Intention analysis makes llms a good jailbreak defender. *arXiv preprint arXiv:2401.06561*.

Zhang, Z.; Yang, J.; Ke, P.; Mi, F.; Wang, H.; and Huang, M. 2023. Defending large language models against jailbreaking attacks through goal prioritization. *arXiv preprint arXiv:2311.09096*.

Zhang, Z.; Zhang, Q.; and Foerster, J. 2024. Parden, can you repeat that? defending against jailbreaks via repetition. *arXiv preprint arXiv:2405.07932*.