MITIGATING MANY-SHOT JAILBREAK ATTACKS WITH ONE DEMONSTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many-shot jailbreaking (MSJ) causes safety-aligned language models to answer harmful queries by conditioning them on many harmful question-answer demonstrations. We study whether this contextual influence can be countered during inference while preserving useful responses. We introduce `Safend`, an inference-time defense that restores refusal behavior with a single fixed safety demonstration. We characterize MSJ and `Safend` as competing contextual response patterns. Harmful demonstrations repeatedly associate unsafe requests with compliant responses, while `Safend` inserts a fixed safety-aligned refusal demonstration after the harmful examples and immediately before the target query. This view treats harmful compliance and refusal as competing response patterns whose influence depends on contextual aggregation rather than example count alone. `Safend` requires no parameter updates, request-specific optimization, or white-box access, and reuses the same demonstration across inputs. Experiments on open-weight and API-based LLMs show that `Safend` reduces the average attack success rate of context-based jailbreaks from 76.7% to 2.3% on Llama-3.1-8B-Instruct and from 46.6% to 2.3% on GPT-4o. These reductions are achieved while preserving general capabilities, long-context performance, and benign response behavior. These results establish a simple and practical approach for recovering safety from adversarial contextual adaptation at inference time. Code is available at https://anonymous.4open.science/r/Safend.

1 INTRODUCTION

Safety alignment teaches large language models (LLMs) to refuse harmful requests while remaining useful on benign tasks (Chen et al., 2025; Xiong et al., 2025). However, this behavior can change substantially under adversarial context. MSJ places multiple harmful question–answer demonstrations before a target query, causing a model that would refuse the query in isolation to produce a harmful response (Anil et al., 2024). As the number of harmful demonstrations increases, the model becomes more likely to follow the demonstrated pattern (Gao et al., 2025; Kulshreshtha et al., 2026; Li et al., 2026; Yan et al., 2025; Wu et al., 2026). MSJ therefore exposes a safety problem during inference: the surrounding context can change refusal behavior even when model parameters and system instructions remain unchanged.

Existing studies explain context-based jailbreaks through mechanisms such as prompt overriding (Ma et al., 2025; Feng et al., 2026), harmful task induction (Kim et al., 2025; Jiang et al., 2025), attention distraction (Weng et al., 2025; Shah et al., 2025), and accumulated multi-turn influence (Russinovich et al., 2025; Rahman et al., 2025). These analyses characterize how adversarial context induces unsafe behavior. We ask a complementary question: when adversarial context has already shifted the model toward harmful behavior, can the model recover refusal within the same request while the harmful demonstrations remain present? A useful defense should preserve useful responses to benign queries without requiring parameter updates or access to model internals.

We introduce `Safend`, a provider-side inference-time recovery mechanism that uses a fixed safety-aligned demonstration: a harmful request paired with a refusal response. After receiving an MSJ input, the provider inserts the safety demonstration after the harmful demonstrations and immediately before the target query. The target query remains the final request, and the model generates its response using the usual decoding procedure. `Safend` preserves the harmful demonstrations and

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

the target query, requires no parameter updates or request-specific optimization, and reuses the same safety demonstration across inputs. It therefore applies to both open-weight and API-based models while adding only a fixed amount of context.

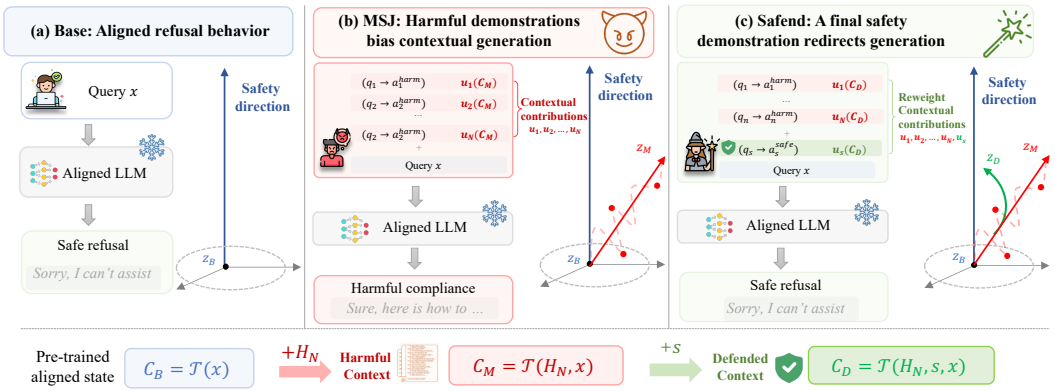

Figure 1: Inference-time safety recovery with `Safend`. A safety-aligned model initially refuses the target query x. MSJ supplies multiple harmful question–answer demonstrations H_N, biasing contextual generation toward harmful compliance. `Safend` inserts a fixed safety-refusal demonstration s as the final demonstration before x, changing the effective contextual contributions and redirecting generation toward refusal, without modifying the target model.

To understand how a single safety demonstration can recover behavior after many harmful demonstrations have shifted the contextual generation state, we separate the response pattern supplied by each demonstration from the contextual aggregation that determines its influence. Harmful demonstrations associate harmful requests with compliant responses, whereas the safety demonstration associates a harmful request with refusal. Inserting the safety demonstration introduces an additional contribution and can also change how the original context influences the response. The relevant comparison is therefore between these contextual contributions, rather than the numbers of harmful and safety demonstrations. This analysis also connects the position of the safety demonstration to its influence at the generation boundary.

We evaluate `Safend` on four LLMs, comprising two open-weight models and two API-based models, against four context-based jailbreak attacks on HarmBench and AdvBench. For each model, we average ASR over the eight combinations of attack and benchmark. Without defense, these averages range from 40.0% to 76.7%; with `Safend`, they range from 1.9% to 2.3%. Performance remains close to the undefended models on seven benchmarks covering general capabilities and long-context tasks, while separate evaluations on benign queries show small increases in over-refusal. We further examine longer MSJ contexts, the position and content of the safety demonstration, and representation changes under Base, MSJ, and defended conditions. The ablations show that both the refusal pattern and its position matter for recovery.

Our main contributions are:

- We identify and study an inference-time safety recovery setting where MSJ-induced harmful contextual adaptation can be reversed without changing the deployed model. We instantiate this setting with Safend, a minimal provider-side intervention using one fixed safety demonstration.
- We develop an analysis of contextual safety recovery by separating the behavioral signal supplied by demonstrations from the contextual aggregation that determines their influence. This explains how harmful demonstrations accumulate and how a safety demonstration can redirect generation.
- We empirically characterize this recovery phenomenon across open-weight and API-based LLMs, showing robust safety recovery under context-based jailbreaks while preserving benign capabilities.

2 Related Work

Many-Shot and Context-Based Jailbreaking. Many-shot jailbreaking exploits in-context learning by presenting repeated harmful question–answer demonstrations before a target request, with attack

effectiveness generally increasing as more demonstrations are supplied (Anil et al., 2024). Related multi-turn attacks similarly accumulate adversarial influence over successive interactions (Russinovich et al., 2025; Rahman et al., 2025). Beyond the number of demonstrations, long-context studies have identified repetition, context length, and demonstration structure as important factors governing jailbreak effectiveness (Kim et al., 2025). These findings show that safety behavior can change substantially under adversarial context even when model parameters remain fixed. We focus on whether such context-induced safety degradation can be recovered during inference.

Mechanisms of In-Context Adaptation. A growing body of work studies how demonstrations alter model behavior without parameter updates. Optimization-based accounts characterize in-context learning through gradient-descent-like learning dynamics in simplified or locally approximated settings (Akyürek et al., 2022; Von Oswald et al., 2023; Dai et al., 2023; Dherin et al., 2025). Complementary representation-space analyses show that signals from multiple demonstrations can be composed into task-relevant directions, while contextualization changes their effective influence through attention routing (Wang et al., 2026). These perspectives motivate our analysis of how harmful response patterns accumulate under MSJ and how a terminal safety demonstration modifies the resulting generation state.

Safety Repair and Inference-Time Defense. Existing safety defenses intervene at different stages of model operation. Adversarial alignment and safety repair modify model parameters using additional supervision (Guo et al., 2025; Rahman et al., 2025; Zhang et al., 2026), whereas representation-level methods directly manipulate internal activations (Zhang et al., 2025). Input-side defenses instead modify the inference context while leaving the deployed model frozen (Jain et al., 2023; Xiong et al., 2025). `Safend` studies inference-time recovery from harmful contextual adaptation within the same inference request, using a single fixed terminal safety demonstration. We provide detailed comparisons with the closest training-time and in-context defenses in Appendix A.

3 Methodology

`Safend` inserts one fixed refusal demonstration between the MSJ examples and the target query. The target model then generates its response without changes to its parameters or decoding configuration. This section defines the setting and the resulting inference procedure; Section 4 analyzes how the demonstrations affect the model's response behavior.

3.1 Problem Setup

Threat model. We consider a safety-aligned target model p_θ with fixed parameters θ. An attacker supplies an MSJ input consisting of N harmful question–answer demonstrations H_N followed by a target query x. The attacker controls the demonstrations, their order, and the number of shots, but cannot modify the model parameters, privileged system instructions, or provider-controlled content. The provider controls prompt assembly and inserts a safety demonstration between H_N and x before invoking the target model. The defense aims to recover refusal on harmful targets while preserving useful responses to benign queries.

Notation. Let x denote the target query and $H_N = [(q_i, a_i^{\text{harm}})]_{i=1}^N$ the ordered MSJ demonstrations, where q_i is a harmful request and a_i^{harm} is a harmful compliant answer. The fixed safety-aligned refusal demonstration is $s = (q_s, a_s^{\text{safe}})$, where q_s is a harmful request and a_s^{safe} refuses it.

Let $\mathcal{T}$ denote the model-specific serialization procedure, including role markers, delimiters, and the tokens marking the start of the assistant response. We define

$$C_{\text{B}} = \mathcal{T}(x), \qquad C_{\text{M}} = \mathcal{T}(H_N, x), \qquad C_{\text{D}} = \mathcal{T}(H_N, s, x), \tag{1}$$

where C_{B}, C_{M}, and C_{D} denote the Base, MSJ, and defended contexts, respectively. In C_{D}, the provider inserts s after the harmful demonstrations and immediately before the target query. Here, terminal placement refers to the final demonstration position preceding x.

3.2 SAFEND: INFERENCE-TIME CONTEXT AUGMENTATION

Given an MSJ input (H_N, x), `Safend` inserts the fixed safety demonstration s between the harmful demonstrations and the target query, producing C_{D}. The target model then generates

$$y \sim p_\theta(\cdot \mid C_{\mathrm{D}}). \tag{2}$$

The demonstration s is fixed before evaluation and reused across target queries and MSJ contexts. `Safend` preserves the content and order of the harmful demonstrations and leaves the target query unchanged; the only addition is the provider-controlled demonstration between them. Appendix C.4 specifies the fixed demonstration and its insertion position.

Because the intervention operates entirely through the inference context, `Safend` requires no parameter update, request-specific optimization, gradient computation, or access to internal activations. For each model-specific serialization, the added input consists only of the fixed demonstration and its formatting tokens; the corresponding runtime overhead is evaluated empirically.

4 FROM HARMFUL ADAPTATION TO SAFETY RECOVERY

MSJ and `Safend` expose the model to two different response patterns: harmful requests followed by compliance and a harmful request followed by refusal. The central question is how the fixed safety-aligned refusal demonstration can alter the behavior induced by multiple harmful demonstrations.

We separate the response pattern specified by each demonstration from the context-dependent computation that determines its influence on the target response.

4.1 DEMONSTRATIONS SUPPLY BEHAVIORAL ADAPTATION SIGNALS

To characterize the behavioral signal associated with a demonstration, consider the representational change induced by a hypothetical one-step update on that demonstration. Let $\ell_\theta(q, a) = -\log p_\theta(a \mid \mathcal{T}(q))$ denote the sequence negative log-likelihood.

For a collection of demonstrations $\mathcal{D}$, define

$$\mathcal{L}_{\mathcal{D}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(q,a)\in\mathcal{D}} \ell_\theta(q, a). \tag{3}$$

For a serialized input C, let $z_\ell(C;\theta) = h_\ell(b(C); C, \theta)$, where $b(C)$ is the final input position used to predict the first response token. A hypothetical update $\theta^+ = \theta - \eta\nabla_\theta\mathcal{L}_{\mathcal{D}}(\theta)$ induces, to first order,

$$z_\ell(C;\theta^+) - z_\ell(C;\theta) \approx -\eta J_\ell(C)\nabla_\theta\mathcal{L}_{\mathcal{D}}(\theta), \qquad J_\ell(C) = \frac{\partial z_\ell(C;\theta)}{\partial\theta}. \tag{4}$$

Equation equation 4 characterizes the first-order representational direction associated with the demonstrated targets. It is used only as a counterfactual analysis: neither MSJ nor `Safend` performs this parameter update.

Taking $\mathcal{D} = H_N$, the demonstrated targets favor harmful compliance. Taking $\mathcal{D} = \{s\}$, the target favors refusal of a harmful request. Applying this construction to H_N and s characterizes the directions associated with harmful compliance and refusal, respectively.

4.2 MSJ AGGREGATES SIGNALS FOR HARMFUL COMPLIANCE

The behavioral signal associated with a demonstration does not by itself determine its influence on generation. The forward pass must aggregate information from all demonstrations in the context. We examine this aggregation at the response-generation boundary, which can attend to the complete serialized input.

Consider one attention head at this boundary, suppressing the head and layer indices. Absorbing the output projection into the value vectors, its contribution to the residual stream is

$$u(C) = \sum_j \alpha_j(C) v_j(C), \qquad \alpha_j(C) = \frac{\exp\big(Q(C)^\top K_j(C)/\sqrt{d_k}\big)}{\sum_t \exp\big(Q(C)^\top K_t(C)/\sqrt{d_k}\big)}. \tag{5}$$

Here $Q(C)$ is the query vector at the response-generation boundary, $K_j(C)$ and $V_j(C)$ are the key and projected value vectors at input position j, and d_k is the key dimension.

Let I_i denote the token positions belonging to harmful demonstration i. Its contribution to the head output is

$$u_i(C) = \sum_{j \in I_i} \alpha_j(C) v_j(C). \tag{6}$$

The attacked head output can therefore be decomposed exactly as

$$u(C_{\mathrm{M}}) = u_0(C_{\mathrm{M}}) + \sum_{i=1}^{N} u_i(C_{\mathrm{M}}), \tag{7}$$

where u_0 collects the contributions from the system prompt, target query, and remaining formatting tokens, excluding all demonstration tokens, including the safety demonstration when present. The harmful examples share the same response pattern, but their contributions need not be equal. Adding examples changes both the available content and its attention allocation.

Figure 2 visualizes how the generation-boundary representations vary with the number of harmful demonstrations. For both Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct, increasing the MSJ shot count moves the attacked states progressively away from the Base cluster and the refusal region. Under the same 16-shot attack, `Safend` shifts the defended states back toward the refusal region, with substantial overlap with the Base cluster in the shared PCA projection.

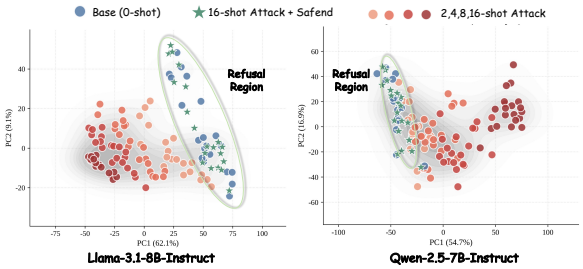

Figure 2: Generation-boundary representations under Base, 2/4/8/16-shot MSJ, and 16-shot MSJ with `Safend` on Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct.

These observations are consistent with a shot-dependent change in the contextual state used for generation and its subsequent modification by the terminal safety demonstration. The PCA projections are descriptive, however, and do not identify the causal components underlying the observed behavioral change. We therefore next analyze how the appended safety demonstration changes the contextual aggregation at the response-generation boundary.

4.3 A Terminal Safety Example Redirects Contextual Steering

Inserting the refusal demonstration between H_N and x changes the contextual aggregation at the response-generation boundary. Equation equation 8 separates this change into the contribution of the added demonstration and changes in the contributions of the original input. Let $u_s(C_{\mathrm{D}})$ denote the contribution of the safety-demonstration tokens at the same attention head. Then

$$\begin{aligned} u(C_{\mathrm{D}}) - u(C_{\mathrm{M}}) = u_s(C_{\mathrm{D}}) + \sum_{i=1}^{N} \left[u_i(C_{\mathrm{D}}) - u_i(C_{\mathrm{M}}) \right] \\ + u_0(C_{\mathrm{D}}) - u_0(C_{\mathrm{M}}). \end{aligned} \tag{8}$$

The first term is the contribution introduced by the fixed safety-aligned refusal demonstration. The remaining terms capture changes in the contributions of the preceding context and other input positions. When serialization preserves the harmful prefix and its positions, representations within H_N remain unchanged. This inserted demonstration can nevertheless change how this prefix contributes at the generation boundary. Because s precedes x, it can also change the contextual representations of the target query and the resulting boundary query. The text of s is fixed, but its representation is conditioned on the preceding harmful demonstrations; the target query is then processed in the presence of both the harmful examples and the refusal demonstration.

Equation equation 8 also shows why demonstration count alone does not determine the outcome. A single demonstration can have a substantial effect when its routed contribution and the resulting

changes to the original input jointly favor refusal. Conversely, additional harmful demonstrations may continue to affect the result through both their content and attention weights.

To relate this contextual change to response behavior, fix the attacked context C_{M} and two continuations for the same target x: a refusal continuation y^{safe} and a harmful compliant continuation y^{harm}. Let $m(u)$ denote the log probability of the refusal continuation minus that of the harmful continuation when the selected head's boundary output is replaced by u, with the attacked input held fixed and downstream computation recomputed.

With $u_{\mathrm{M}} = u(C_{\mathrm{M}})$ and $\Delta u = u(C_{\mathrm{D}}) - u(C_{\mathrm{M}})$, a first-order expansion gives

$$m(u_{\mathrm{M}} + \Delta u) - m(u_{\mathrm{M}}) = \left\langle \nabla_u m(u_{\mathrm{M}}), \Delta u \right\rangle + \mathcal{O}(\|\Delta u\|_2^2). \tag{9}$$

The first-order term measures whether the contextual change at this head is aligned with an increase in the refusal margin. This relates the contextual change to the refusal margin without requiring exact restoration of the Base representation. The corresponding local recovery condition is given in Appendix B.

Placing s at the end of the demonstration block supplies a refusal example after the harmful demonstrations and before the model processes the target query. Our position ablation supports this choice: with the safety demonstration held fixed, placing it immediately before x is more effective than placing it at the beginning or middle of the harmful demonstration block. The safety and utility evaluations in Section 5 measure the resulting response behavior on the original target x.

5 Experiments

5.1 Experimental Setup

Models. We evaluate `Safend` on two open-weight models, Llama-3.1-8B-Instruct (Dubey et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024), and two API-based models, GPT-4o (Achiam et al., 2023) and Gemini 2.5 Pro (Comanici et al., 2025). The former support analysis of internal representations, while the latter test whether the same intervention remains effective without access to model internals. The study of longer adversarial contexts in Section 5.3 additionally includes Gemini 2.5 Flash and DeepSeek-V3 (Liu et al., 2024).

Baselines. We compare `Safend` with three defenses applied during inference: In-Context Defense (ICD) (Wei et al., 2026), Goal Prioritization (Zhang et al., 2024), and PPL Filter (Jain et al., 2023). PPL Filter is evaluated only on open-weight models, for which the required token probabilities are available. MTSA (Guo et al., 2025) and MUSE (Yan et al., 2025) provide complementary reference points by improving safety through parameter updates. We evaluate these training methods on the open-weight models and distinguish them from defenses that leave the target checkpoint unchanged. Appendix C.3 provides the baseline configurations.

Attacks. The attack suite covers two ways of constructing adversarial context. MSJ (Anil et al., 2024) supplies harmful QA demonstrations, and PANDAS (Ma et al., 2025) augments this construction with adaptive demonstration selection. Crescendo (Russinovich et al., 2025) and X-teaming (Rahman et al., 2025) instead accumulate adversarial context through conversation. This selection tests `Safend` both on the demonstration blocks central to MSJ and on histories produced through interaction. The main experiments use $N = 32$ demonstrations for MSJ and PANDAS; Crescendo and X-teaming follow their respective interaction protocols. Details are provided in Appendix C.2.

Evaluation. We measure safety on HarmBench (Mazeika et al., 2024) and AdvBench (Zou et al., 2023), reporting Attack Success Rate (ASR; lower is better) with LlamaGuard-3 (Inan et al., 2023) as the primary evaluator. Appendix C.1.2 describes the safety evaluation.

To assess whether protection comes at the expense of normal model use, we evaluate reasoning on GSM8K (Cobbe et al., 2021), MMLU-Pro (Wang et al., 2024b), and GPQA (Rein et al., 2024); instruction following on MT-Bench (Zheng et al., 2023) and AlpacaEval (Dubois et al., 2023); and long-context capabilities on Loong (Wang et al., 2024a) and LongBench v2 (Bai et al., 2025). XSTest (Röttger et al., 2024) and OKTest (Shi et al., 2024) separately assess unnecessary refusal of benign requests. We report the standard metric for each benchmark.

Table 1: Defense effectiveness against context-based jailbreaks on open-weight and API-based LLMs. We report ASR (%, ↓) judged by LlamaGuard-3; parentheses show absolute reductions relative to the undefended Base model. MSJ and PANDAS use 32 shots, whereas Crescendo and X-teaming follow their attack-specific interaction budgets described in Appendix C.2. Training-time defenses are included as references and modify model parameters.

Model	Defense Method	HarmBench				AdvBench			
		MSJ	PANDAS	Crescendo	X-teaming	MSJ	PANDAS	Crescendo	X-teaming
Llama-3.1-8B-In	No Defense	63.5	68.0	77.0	60.0	74.8	96.0	86.2	88.0
	Training-Time Defense								
	MTSA	28.0 (↓35.5)	30.5 (↓37.5)	34.5 (↓42.5)	27.0 (↓33.0)	32.0 (↓42.8)	41.0 (↓55.0)	36.0 (↓50.2)	37.0 (↓51.0)
	MUSE	25.0 (↓38.5)	27.0 (↓41.0)	31.0 (↓46.0)	24.5 (↓35.5)	29.5 (↓45.3)	38.0 (↓58.0)	33.5 (↓52.7)	34.0 (↓54.0)
	Inference-Time Defense								
	ICD	55.0 (↓8.5)	58.5 (↓9.5)	66.0 (↓11.0)	52.0 (↓8.0)	65.0 (↓9.8)	84.0 (↓12.0)	75.0 (↓11.2)	76.5 (↓11.5)
	PPL-Filter	56.5 (↓7.0)	60.0 (↓8.0)	68.0 (↓9.0)	53.5 (↓6.5)	67.0 (↓7.8)	86.0 (↓10.0)	77.0 (↓9.2)	78.5 (↓9.5)
	Goal Prioritization	52.0 (↓11.5)	55.0 (↓13.0)	63.0 (↓14.0)	50.0 (↓10.0)	61.0 (↓13.8)	80.0 (↓16.0)	73.0 (↓13.2)	74.5 (↓13.5)
	Safend (Ours)	**4.2** (↓59.3)	**4.5** (↓63.5)	**0.0** (↓77.0)	**0.0** (↓60.0)	**4.5** (↓70.3)	**4.9** (↓91.1)	**0.0** (↓86.2)	**0.0** (↓88.0)
Qwen2.5-7B-In	No Defense	39.5	44.0	48.8	42.0	43.5	50.0	46.0	47.5
	Training-Time Defense								
	MTSA	16.5 (↓23.0)	18.0 (↓26.0)	20.0 (↓28.8)	17.0 (↓25.0)	18.5 (↓25.0)	21.0 (↓29.0)	19.0 (↓27.0)	20.0 (↓27.5)
	MUSE	14.5 (↓25.0)	16.0 (↓28.0)	18.5 (↓30.3)	15.5 (↓26.5)	16.5 (↓27.0)	19.0 (↓31.0)	17.5 (↓28.5)	18.5 (↓29.0)
	Inference-Time Defense								
	ICD	34.0 (↓5.5)	37.0 (↓7.0)	41.5 (↓7.3)	35.5 (↓6.5)	37.0 (↓6.5)	43.0 (↓7.0)	40.0 (↓6.0)	41.0 (↓6.5)
	PPL-Filter	35.5 (↓4.0)	38.5 (↓5.5)	43.0 (↓5.8)	37.0 (↓5.0)	39.0 (↓4.5)	45.0 (↓5.0)	41.5 (↓4.5)	42.5 (↓5.0)
	Goal Prioritization	31.5 (↓8.0)	34.5 (↓9.5)	39.0 (↓9.8)	33.0 (↓9.0)	34.0 (↓9.5)	40.0 (↓10.0)	37.5 (↓8.5)	38.5 (↓9.0)
	Safend (Ours)	**3.5** (↓36.0)	**3.8** (↓40.2)	**0.0** (↓48.8)	**0.0** (↓42.0)	**3.8** (↓39.7)	**4.2** (↓45.8)	**0.0** (↓46.0)	**0.0** (↓47.5)
GPT-4o	No Defense	10.2	10.5	46.0	89.3	10.1	10.6	100.0	96.2
	Inference-Time Defense								
	ICD	9.7 (↓0.5)	9.8 (↓0.7)	44.5 (↓1.5)	88.0 (↓1.3)	9.6 (↓0.5)	9.9 (↓0.7)	99.0 (↓1.0)	95.0 (↓1.2)
	Goal Prioritization	9.3 (↓0.9)	9.4 (↓1.1)	43.0 (↓3.0)	86.5 (↓2.8)	9.2 (↓0.9)	9.4 (↓1.2)	97.5 (↓2.5)	93.8 (↓2.4)
	Safend (Ours)	**4.5** (↓5.7)	**4.7** (↓5.8)	**0.0** (↓46.0)	**0.0** (↓89.3)	**4.3** (↓5.8)	**4.6** (↓6.0)	**0.0** (↓100.0)	**0.0** (↓96.2)
Gemini 2.5 Pro	No Defense	10.5	10.8	57.0	36.4	10.2	10.6	88.9	95.7
	Inference-Time Defense								
	ICD	9.8 (↓0.7)	10.0 (↓0.8)	55.0 (↓2.0)	34.8 (↓1.6)	9.7 (↓0.5)	10.0 (↓0.6)	86.5 (↓2.4)	94.0 (↓1.7)
	Goal Prioritization	9.4 (↓1.1)	9.6 (↓1.2)	52.5 (↓4.5)	33.0 (↓3.4)	9.2 (↓1.0)	9.5 (↓1.1)	83.0 (↓5.9)	90.5 (↓5.2)
	Safend (Ours)	**4.2** (↓6.3)	**4.5** (↓6.3)	**0.0** (↓57.0)	**0.0** (↓36.4)	**4.0** (↓6.2)	**4.3** (↓6.3)	**0.0** (↓88.9)	**0.0** (↓95.7)

Implementation. Unless otherwise specified, the same fixed safety-aligned refusal demonstration s is reused across models, attacks, datasets, and target queries, with no test-time demonstration selection or optimization. Its construction and input serialization are described in Appendix C.4.

5.2 Main Results

Defense Effectiveness. Using the same fixed safety demonstration, `Safend` achieves the lowest ASR in every model, attack, and benchmark combination in Table 1. Under MSJ and PANDAS with 32 demonstrations, its ASR ranges from 3.5% to 4.9% across the four models and two benchmarks. These results support the central premise of `Safend`: one fixed refusal example can counter harmful responses induced by many demonstrations, even when the harmful examples remain in the context and the target model is unchanged.

The same pattern is observed on the API-based models, where no parameter or internal-state access is available. The macro-average ASR decreases from 46.6% to 2.3% on GPT-4o and from 40.0% to 2.1% on Gemini 2.5 Pro. In particular, `Safend` reduces ASR to 0.0% under Crescendo and X-teaming on both benchmarks for these models. MSJ and PANDAS use 32 shots, whereas Crescendo and X-teaming follow their attack-specific interaction budgets described in Appendix C.2.

Benign Capability Preservation. We apply the same safety demonstration to benign inputs without adversarial context to assess whether the safety gains come at the expense of ordinary task performance. Table 2 shows that `Safend` remains close to No Defense across all four models: decreases on GSM8K, MMLU-Pro, GPQA, and AlpacaEval do not exceed 0.5 points on their respective reported scales, while MT-Bench decreases by 0.1. It also achieves the highest or jointly highest score among the evaluated defenses on every general capability benchmark for each model. The large reductions in jailbreak ASR are therefore accompanied by only small decreases in benign task scores.

Preserving performance on long contexts is particularly important because `Safend` intervenes in the context used for generation. On Loong and LongBench v2, scores decrease by 0.2 to 0.7 points relative to No Defense across the four models, while remaining higher than those of every other evaluated defense. Together with the safety results, these findings support `Safend`'s ability to reduce jailbreak success while largely preserving benign task performance, including tasks requiring extensive context.

Table 2: Benign performance on general and long-context benchmarks (↑). Red parentheses report the decrease from Base.

Model	Defense Method	General Capabilities					Long-Context	
		GSM8K	MMLU-Pro	GPQA	MT-Bench	AlpacaEval	Loong	LB-v2
Llama-3.1-8B-In	No Defense	78.1	44.3	29.5	7.1	22.9	28.4	27.0
	Training-Time Defense							
	MTSA	74.8 (↓3.3)	40.5 (↓3.8)	26.0 (↓3.5)	6.5 (↓0.6)	20.8 (↓2.1)	25.8 (↓2.6)	24.5 (↓2.5)
	MUSE	75.8 (↓2.3)	41.8 (↓2.5)	27.0 (↓2.5)	6.6 (↓0.5)	21.5 (↓1.4)	26.6 (↓1.8)	25.4 (↓1.6)
	Inference-Time Defense							
	ICD	77.2 (↓0.9)	43.2 (↓1.1)	28.4 (↓1.1)	6.9 (↓0.2)	22.2 (↓0.7)	27.8 (↓0.6)	26.5 (↓0.5)
	PPL-Filter	77.5 (↓0.6)	43.8 (↓0.5)	29.0 (↓0.5)	7.0 (↓0.1)	22.5 (↓0.4)	27.7 (↓0.7)	26.6 (↓0.4)
	Goal Prioritization	76.5 (↓1.6)	42.8 (↓1.5)	28.0 (↓1.5)	6.8 (↓0.3)	21.8 (↓1.1)	27.2 (↓1.2)	26.0 (↓1.0)
	Safend (Ours)	**77.8** (↓0.3)	**44.0** (↓0.3)	**29.0** (↓0.5)	**7.0** (↓0.1)	**22.7** (↓0.2)	**28.0** (↓0.4)	**26.8** (↓0.2)
Qwen2.5-7B-In	No Defense	89.8	56.7	42.0	7.9	31.0	31.4	30.0
	Training-Time Defense							
	MTSA	85.5 (↓4.3)	52.0 (↓4.7)	36.5 (↓5.5)	7.3 (↓0.6)	28.0 (↓3.0)	28.5 (↓2.9)	27.0 (↓3.0)
	MUSE	87.0 (↓2.8)	53.5 (↓3.2)	38.0 (↓4.0)	7.4 (↓0.5)	29.2 (↓1.8)	29.5 (↓1.9)	28.5 (↓1.5)
	Inference-Time Defense							
	ICD	89.0 (↓0.8)	56.0 (↓0.7)	40.8 (↓1.2)	7.6 (↓0.3)	30.3 (↓0.7)	30.5 (↓0.9)	29.2 (↓0.8)
	PPL-Filter	89.2 (↓0.6)	56.3 (↓0.4)	41.5 (↓0.5)	7.7 (↓0.2)	30.7 (↓0.3)	30.8 (↓0.6)	29.5 (↓0.5)
	Goal Prioritization	88.0 (↓1.8)	55.0 (↓1.7)	39.5 (↓2.5)	7.5 (↓0.4)	29.5 (↓1.5)	29.8 (↓1.6)	28.8 (↓1.2)
	Safend (Ours)	**89.5** (↓0.3)	**56.5** (↓0.2)	**41.6** (↓0.4)	**7.8** (↓0.1)	**30.9** (↓0.1)	**31.0** (↓0.4)	**29.8** (↓0.2)
GPT-4o	No Defense	93.8	86.2	63.5	9.2	57.5	56.0	54.2
	Inference-Time Defense							
	ICD	93.0 (↓0.8)	85.5 (↓0.7)	62.0 (↓1.5)	9.0 (↓0.2)	56.5 (↓1.0)	55.2 (↓0.8)	53.4 (↓0.8)
	Goal Prioritization	92.5 (↓1.3)	85.0 (↓1.2)	61.0 (↓2.5)	8.9 (↓0.3)	55.8 (↓1.7)	54.8 (↓1.2)	52.9 (↓1.3)
	Safend (Ours)	**93.6** (↓0.2)	**86.0** (↓0.2)	**63.2** (↓0.3)	**9.1** (↓0.1)	**57.2** (↓0.3)	**55.8** (↓0.2)	**53.9** (↓0.3)
Gemini 2.5 Pro	No Defense	92.5	85.5	62.8	9.0	58.0	57.5	63.3
	Inference-Time Defense							
	ICD	91.8 (↓0.7)	84.8 (↓0.7)	61.5 (↓1.3)	8.8 (↓0.2)	56.8 (↓1.2)	56.2 (↓1.3)	62.0 (↓1.3)
	Goal Prioritization	91.0 (↓1.5)	84.0 (↓1.5)	60.0 (↓2.8)	8.6 (↓0.4)	55.5 (↓2.5)	55.0 (↓2.5)	61.0 (↓2.3)
	Safend (Ours)	**92.3** (↓0.2)	**85.3** (↓0.2)	**62.5** (↓0.3)	**8.9** (↓0.1)	**57.6** (↓0.4)	**57.0** (↓0.5)	**62.6** (↓0.7)

Over-Refusal. Table 3 evaluates unnecessary refusal on benign prompts. On Llama-3.1-8B-Instruct, `Safend` increases over-refusal by 0.2% on XSTest and 0.4% on OKTest. On Qwen2.5-7B-Instruct, the corresponding increases are 0.4% and 1.6%. These changes are substantially smaller than those observed for MUSE. Thus, the reduction in jailbreak ASR is not accompanied by a broad increase in refusal on benign requests.

Table 3: Over-refusal on benign prompts (%, ↓).

Method	Llama-3.1		Qwen2.5	
	XSTest	OKTest	XSTest	OKTest
Base	9.0	8.0	19.3	1.6
MUSE	28.0	17.0	25.6	11.6
Safend	**9.2**	**8.4**	**19.7**	**3.2**

Runtime Overhead. Table 4 reports additional cost on Llama-3.1-8B-Instruct under a 64-shot query. `Safend` adds approximately 0.05 s, compared with 0.3 s for ICD, 0.9 s for PPL-Filter, and 7.5 s for Goal Prioritization. The defense also requires no training, whereas MTSA and MUSE require 5.5 and 4.2 hours, respectively. Because `Safend` uses a single fixed demonstration and no additional model call, its deployment cost is primarily the additional input tokens. Token usage is reported in Appendix C.6.

Table 4: Defense overhead on Llama-3.1-8B-Instruct under a 64-shot query.

Method	**Δ Training**	**Δ Inference**	**Description**
Training-Time Defense			
MTSA (Guo et al., 2025)	5.5 h	—	Multi-turn safety fine-tuning via RLHF.
MUSE (Yan et al., 2025)	4.2 h	—	Multi-turn safety alignment with DPO.
Inference-Time Defense			
ICD	—	≈0.3 s	Insert 10 safety examples into the prompt context.
PPL-Filter	—	≈0.9 s	Extra forward pass to compute perplexity.
Goal Prioritization	—	≈7.5 s	Generate and rank five candidate responses.
Safend (Ours)	—	**≈0.05 s**	Append one fixed safety demonstration.

5.3 Robustness and Design Ablations

Robustness to Longer Adversarial Contexts. We vary the number of harmful demonstrations from 32 to 256 while keeping the safety demonstration fixed. Figure 3 shows that MSJ remains

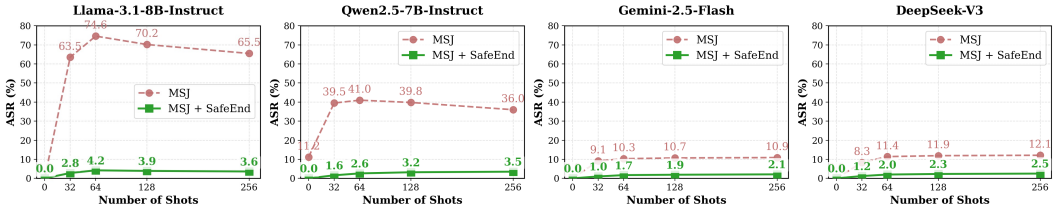

Figure 3: Mean ASR under MSJ as the number of harmful demonstrations increases from 32 to 256. `Safend` uses the same safety demonstration at every shot count.

effective without defense across this range, although the trend varies by model. For `Safend`, the highest mean ASR across the evaluated shot counts is 4.2% on Llama-3.1-8B-Instruct, 3.5% on Qwen2.5-7B-Instruct, 2.1% on Gemini 2.5 Flash, and 2.5% on DeepSeek-V3. The same safety demonstration maintains low ASR as harmful demonstrations increase eightfold.

Position of the Safety Demonstration. We keep the safety demonstration s fixed and vary its position within the adversarial context. For 32-shot MSJ, we place s before all harmful demonstrations (*Start*), after the first 16 demonstrations (*Middle*), or after all harmful demonstrations and immediately before the target query x (*End*). Crescendo uses corresponding placements within the accumulated conversation history. Table 5 shows that End consistently outperforms both earlier placements: MSJ ASR falls to 4.2% on HarmBench and 4.5% on AdvBench, while Crescendo ASR falls to 0.0% on both benchmarks. The same refusal example is therefore substantially more effective when it is the final demonstration before the target query, supporting the placement used by `Safend`.

Table 5: Effect of safety-demonstration position on Llama-3.1-8B-Instruct (ASR, $\%, \downarrow$). The demonstration content is fixed across positions.

Position	HarmBench		AdvBench	
	MSJ	Cres.	MSJ	Cres.
Start	58.0	71.5	69.0	80.3
Middle	42.5	55.8	51.2	63.0
End	**4.2**	**0.0**	**4.5**	**0.0**

Content of the Terminal Demonstration. We next hold the 32-shot MSJ context and insertion position fixed and replace only the terminal QA pair. Table 6 shows that harmful compliance retains high ASR, while benign compliance provides partial mitigation. The refusal demonstration reduces ASR to 4.2% on HarmBench and 4.5% on AdvBench, providing substantially stronger protection than either compliance control at the same position. Together with the position ablation, these results show that both where the example appears and which response it demonstrates matter for recovery. Representative examples are provided in Appendix C.5.2.

Table 6: Effect of terminal demonstration content under a fixed 32-shot MSJ context on Llama-3.1-8B-Instruct (ASR, $\%, \downarrow$).

Terminal QA	Harm Bench	Adv Bench
Benign Compliance	28.0	41.5
Harmful Compliance	68.0	79.5
Safety Refusal	**4.2**	**4.5**

6 Conclusion

We presented `Safend`, an inference-time recovery approach that studies how safety behavior can be restored after adversarial contextual adaptation. It instantiates this idea by inserting one fixed refusal demonstration between the adversarial context and the target query. Across the evaluated models and attacks, this intervention substantially reduces ASR while largely preserving benign task performance. It remains effective with up to 256 harmful demonstrations, and the ablations show that both refusal content and placement before the target query are important for recovery.

Our analysis separates the behavioral pattern supplied by each demonstration from the contextual computation that determines its influence on generation. It characterizes recovery through changes in refusal preference rather than exact reconstruction of the model's original representation. Together, these findings show that safety behavior disrupted by adversarial context can be recovered through contextual intervention. More broadly, they suggest that inference-time safety is not determined solely by the aligned model parameters, but also by how safety-relevant behaviors are accumulated, weighted, and corrected within dynamic contexts.

AI USE STATEMENT

Generative AI tools were used for basic copyediting and notation checking. All AI-assisted material was reviewed by the authors, who made all final research and writing decisions and take responsibility for the complete content of this work.

ETHICS STATEMENT

This work studies many-shot jailbreaking to understand and reduce safety failures in large language models. Analyses of jailbreak mechanisms are inherently dual-use: findings that support defensive design may also inform stronger attacks. We mitigate this risk by evaluating established attacks in controlled research settings and by focusing released artifacts on defense and reproducible safety evaluation rather than attacks against deployed systems. The experiments use public safety benchmarks and model interfaces and involve no human subjects or private personal data. Harmful model outputs are generated and inspected solely for safety evaluation and are not acted upon. Released artifacts will follow the licenses and usage policies of the corresponding datasets and models and will exclude credentials, private logs, and harmful outputs unnecessary for reproduction.

REPRODUCIBILITY STATEMENT

The Methodology section specifies the construction of the Base, MSJ, and `Safend` inputs, including the exact ordering and serialization of the fixed safety demonstration. The experimental setup documents the evaluated models, datasets, attacks, baselines, metrics, and evaluation protocols. The appendix provides the complete safety demonstration, prompt templates, model and API versions, decoding settings, random seeds, theoretical assumptions and proofs, and additional results. The anonymous repository provides the implementation, evaluation scripts, configurations, and commands used to reproduce the main experiments.

REFERENCES

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

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking. *Advances in Neural Information Processing Systems*, 37:129696–129742, 2024.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, 2025.

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. Secalign: Defending against prompt injection with preference optimization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2833–2847, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. Learning without training: The implicit dynamics of in-context learning. *arXiv preprint arXiv:2507.16003*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Mingqian Feng, Xiaodong Liu, Weiwei Yang, Jialin Song, Xuekai Zhu, Chenliang Xu, and Jianfeng Gao. Sema: Simple yet effective learning for multi-turn jailbreak attacks. *arXiv preprint arXiv:2602.06854*, 2026.

Lang Gao, Jiahui Geng, Xiangliang Zhang, Preslav Nakov, and Xiuying Chen. Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25378–25398, 2025.

Weiyang Guo, Jing Li, Wenya Wang, Yu Li, Daojing He, Jun Yu, and Min Zhang. Mtsa: Multi-turn safety alignment for llms through multi-round red-teaming. *arXiv preprint arXiv:2505.17147*, 2025.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

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

Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. Red queen: Exposing latent multi-turn risks in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25554–25591, 2025.

Sangyeop Kim, Yohan Lee, Yongwoo Song, and Kimin Lee. What really matters in many-shot attacks? an empirical study of long-context vulnerabilities in llms. *arXiv preprint arXiv:2505.19773*, 2025.

Devang Kulshreshtha, Hang Su, Chinmay Hegde, and Haohan Wang. Multi-turn jailbreaking of aligned llms via lexical anchor tree search. *arXiv preprint arXiv:2601.02670*, 2026.

Songze Li, Ruishi He, Xiaojun Jia, Jun Wang, and Zhihui Fu. Knowledge-driven multi-turn jailbreaking on large language models. *arXiv preprint arXiv:2601.05445*, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Avery Ma, Yangchen Pan, and Amir-massoud Farahmand. Pandas: Improving many-shot jailbreaking via positive affirmation, negative demonstration, and adaptive sampling. *arXiv preprint arXiv:2502.01925*, 2025.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: a standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 35181–35224, 2024.

Salman Rahman, Liwei Jiang, James Shiffer, Genglin Liu, Sheriff Issaka, Md Rizwan Parvez, Hamid Palangi, Kai-Wei Chang, Yejin Choi, and Saadia Gabriel. X-teaming: Multi-turn jailbreaks and defenses with adaptive multi-agents. *arXiv preprint arXiv:2504.13203*, 2025.

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 *First Conference on Language Modeling*, 2024.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. Great, now write an article about that: The crescendo {Multi-Turn}{LLM} jailbreak attack. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2421–2440, 2025.

Rishi Rajesh Shah, Chen Henry Wu, Shashwat Saxena, Ziqian Zhong, Alexander Robey, and Aditi Raghunathan. Jailbreaking in the haystack. *arXiv preprint arXiv:2511.04707*, 2025.

Chenyu Shi, Xiao Wang, Qiming Ge, Songyang Gao, Xianjun Yang, Tao Gui, Qi Zhang, Xuan-Jing Huang, Xun Zhao, and Dahua Lin. Navigating the overkill in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4602–4614, 2024.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Entang Wang, Yiwei Wang, Aleksandra Bakalova, and Michael Hahn. How few-shot examples add up: A causal decomposition of function vectors in in-context learning. *arXiv preprint arXiv:2605.16591*, 2026.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, et al. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5627–5646, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

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

Zixuan Weng, Xiaolong Jin, Jinyuan Jia, and Xiangyu Zhang. Foot-in-the-door: A multi-turn jailbreak for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1939–1950, 2025.

Jiaqi Wu, Chen Chen, Chunyan Hou, and Xiaojie Yuan. Safeint: Shielding large language models from jailbreak attacks via safety-aware representation intervention. *arXiv preprint arXiv:2502.15594*, 2025.

Yutao Wu, Xiao Liu, Yifeng Gao, Xiang Zheng, Hanxun Huang, Yige Li, Cong Wang, Bo Li, Xingjun Ma, and Yu-Gang Jiang. Internal safety collapse in frontier large language models. *arXiv preprint arXiv:2603.23509*, 2026.

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

Siyu Yan, Long Zeng, Xuecheng Wu, Chengcheng Han, Kongcheng Zhang, Chong Peng, Xuezhi Cao, Xunliang Cai, and Chenjuan Guo. Muse: Mcts-driven red teaming framework for enhanced multi-turn dialogue safety in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21293–21314, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

Jiawen Zhang, Kejia Chen, Lipeng He, Jian Lou, Dan Li, Zunlei Feng, Mingli Song, Jian Liu, Kui Ren, and Xiaohu Yang. Activation approximations can incur safety vulnerabilities in aligned {LLMs}: Comprehensive analysis and defense. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 339–358, 2025.

Jiawen Zhang, Lipeng He, Kejia Chen, Jian Lou, Jian Liu, Xiaohu Yang, and Ruoxi Jia. Safety at one shot: Patching fine-tuned llms with a single instance. *arXiv preprint arXiv:2601.01887*, 2026.

Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. Defending large language models against jailbreaking attacks through goal prioritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8865–8887, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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.

A Positioning, Scope, and Alternative Explanations

A.1 Relation to Closest Safety-Repair Paradigms

`Safend` studies safety recovery when harmful behavior is induced by the current inference context while the deployed model remains unchanged. The provider retains the adversarial context and inserts one fixed refusal demonstration immediately before the target query. Table 7 compares representative approaches by their intervention point and deployment requirements.

Table 7: Positioning of representative safety interventions by degradation source and deployment requirements. $\Delta\theta$ indicates whether the deployed target model is updated; WB denotes whether the intervention requires access to model internals. The target-update column does not describe whether an auxiliary intervention is learned.

Method	Degradation source	Intervention	$\Delta\theta$	WB
Safety at One Shot (Zhang et al., 2026)	Fine-tuning-induced drift	Parameters	✓	✓
SafeInt (Wu et al., 2025)	Jailbreak-induced shift	Activations	✗	✓
ICD (Wei et al., 2026)	Jailbreak context	Context	✗	✗
`Safend`	MSJ-induced adaptation	Context	✗	✗

Relation to parameter-space safety repair. Safety at One Shot (Zhang et al., 2026) studies safety repair after fine-tuning through a parameter update using a single safety example. We adopt its safety QA and fix it before evaluation. In `Safend`, the pair is used only as inference context: the target checkpoint remains unchanged, and the intervention acts on the response generated for the current request.

The distinction therefore concerns both the source of degradation and the intervention itself. Parameter repair addresses changes introduced during training, whereas `Safend` addresses harmful behavior induced by the current context. Its contribution concerns safety recovery through contextual insertion, rather than the discovery of the safety example or a new procedure for updating model parameters.

Relation to in-context defenses. ICD (Wei et al., 2026) uses safety demonstrations during inference while keeping the target model fixed. `Safend` belongs to this general intervention paradigm and examines a specific recovery problem: whether one fixed refusal example remains effective when it follows many harmful demonstrations and precedes the target query.

Our analysis examines how the added example contributes to generation and how it can change the influence of the original context. The position and content ablations test these two design choices separately. The focus is therefore the recovery achieved by a single fixed example under adversarial demonstration histories, rather than the use of refusal demonstrations in general.

Relation to activation-level defenses. Activation interventions, such as SafeInt (Wu et al., 2025), directly modify internal representations and require access to model internals. `Safend` instead changes only the inference context. The representation analysis and the local activation intervention defined in Appendix B serve to analyze contextual effects; neither is part of the deployed defense.

A.2 Novelty Boundary and Scope

What is new in `Safend`. `Safend` addresses safety recovery when an aligned model is steered toward harmful compliance by adversarial demonstrations without any change to its parameters. The contribution comprises a provider-side procedure that inserts one fixed refusal example before the target query; an analysis separating demonstration content from its contextual contribution; and an empirical evaluation of safety, benign capabilities, and the roles of example content and position.

The fixed QA is adopted from prior work. The contribution lies in the recovery setting, its inference procedure, and the analysis and evaluation of that procedure.

What we do not claim. The counterfactual update in Section 4 characterizes the representational direction associated with learning from a demonstration. It does not identify prompt conditioning

with an actual parameter update. The attention decomposition describes how input components contribute to a selected head at the generation boundary, while the local recovery condition concerns a specified intervention on that head.

These results do not assert that a single head or pathway explains the complete defense, or that a positive margin for two selected continuations guarantees safe generation in general. Their role is to make the proposed account of contextual recovery precise and testable.

Operational scope. `Safend` assumes that the provider controls input serialization and can identify the boundary between the preceding context and the final query. The provider inserts the fixed safety QA at that boundary while retaining the original context and target query. The defended input must fit within the model's usable context window.

The reported results characterize robustness under the evaluated attack protocols and shot counts; they are not a formal guarantee for arbitrary adversarial contexts. An attacker may construct inputs that weaken the influence of the fixed demonstration, and longer inputs may introduce position or context-window effects beyond the evaluated settings.

A.3 Alternative Explanations and Diagnostic Controls

Inserting a refusal example changes the content available to the model and the context in which the target query is processed. The following controls clarify which aspects of recovery are supported by the reported experiments.

Recency and additional-token effects. Placing an example immediately before the target query may influence generation through its position as well as its content. The position ablation keeps the safety QA fixed and moves it within the adversarial history. The content ablation fixes the insertion position and compares refusal with benign and harmful compliance.

The refusal example outperforms both compliance controls at the same position, while moving it earlier substantially weakens protection. These comparisons support the importance of both content and position. They do not establish that token length or formatting has no effect, or that recency alone explains the observed differences.

Task substitution. The safety QA is a completed example placed before x, which remains the final request. This ordering preserves the query the model is expected to answer. For harmful targets, refusal is an intended outcome; however, a low ASR alone does not establish that the model remains useful on benign requests. We therefore report benign capability and over-refusal results alongside safety scores. These evaluations assess whether the same insertion rule preserves appropriate responses to safe queries. Appendix C.4 specifies the input ordering.

Example-specific and context-length effects. The safety QA is fixed before evaluation and reused without retrieval, rewriting, or optimization for individual requests. The experiments assess the transfer of this selected example across the evaluated inputs and attack settings; they do not establish equivalent performance for every possible refusal example.

At each shot count, the defended and undefended conditions use the same harmful demonstrations. The intervention adds the safety QA rather than reducing or replacing the adversarial context. Appendix C.6 states the context-feasibility requirement and distinguishes the added token budget from measured runtime overhead.

B Supporting Derivations for Contextual Steering

Counterfactual learning direction. Fix a serialized input C and let $g_{\mathcal{D}} = \nabla_\theta \mathcal{L}_{\mathcal{D}}(\theta)$. Let $\eta > 0$ be the hypothetical step size. Suppose the Jacobian $J_\ell(C;\vartheta) = \partial z_\ell(C;\vartheta)/\partial\vartheta$ is L_z-Lipschitz in operator norm along the segment from θ to $\theta - \eta g_{\mathcal{D}}$. The integral form of Taylor's theorem gives

$$\left\| z_\ell(C;\theta - \eta g_{\mathcal{D}}) - z_\ell(C;\theta) + +\eta J_\ell(C;\theta) g_{\mathcal{D}} \right\|_2 \leq \frac{L_z \eta^2}{2} \|g_{\mathcal{D}}\|_2^2. \tag{10}$$

This bound controls the remainder in the first-order approximation in Eq. equation 4. It applies to a fixed input under a hypothetical parameter perturbation. MSJ and `Safend` instead change the input to a model with fixed parameters, so the bound does not equate their contextual representations with the representation obtained after the hypothetical update.

Selective rerouting of the original context. Let the shared prefix consist of the serialized system instructions and harmful demonstrations preceding the insertion point of s. It excludes the target query x and the response-generation cue. Assume that this prefix has identical token identities, position indices, causal attention masks, and positional encoding settings under C_{M} and C_{D}.

Consider two token positions j and k in this prefix that are visible from the generation boundary in both inputs. Under causal attention, their representations do not depend on the subsequently inserted demonstration. Their key vectors at a fixed layer and head are therefore identical between the two inputs; denote them by K_j and K_k. For the scaled dot-product attention in Eq. equation 5,

$$\log \frac{\alpha_i(C_{\mathrm{D}})/\alpha_j(C_{\mathrm{D}})}{\alpha_i(C_{\mathrm{M}})/\alpha_j(C_{\mathrm{M}})} = \frac{\big(Q(C_{\mathrm{D}}) - Q(C_{\mathrm{M}})\big)^{\top}(K_i - K_j)}{\sqrt{d_k}}. \tag{11}$$

Here, Q and K include the positional transformations used in the attention computation. With the boundary query and these prefix keys held fixed, introducing additional keys changes the softmax normalization but preserves the attention ratio between j and k.

Under `Safend`, however, x is processed after s rather than immediately after H_N. Its representations and the resulting boundary query can therefore change, allowing attention to be redistributed among the unchanged prefix positions.

The identity characterizes relative routing, not the direction of the safety effect. It does not require attention to every harmful example to decrease, nor does it establish that any such decrease alone causes refusal.

A local condition for recovery under an activation patch. Let u_b denote the output of a selected attention head at the generation boundary. Fix the attacked input C_{M} and consider an intervention that replaces this output by u, leaving the other incoming paths unchanged and recomputing downstream effects.

For two fixed continuations of the same target query, a refusal response y^{safe} and a harmful compliant response y^{harm}, define the pairwise safety margin as

$$m(u) = \log p_\theta\big(y^{\mathrm{safe}} \mid C_{\mathrm{M}}; \mathrm{do}(u_b = u)\big) - \log p_\theta\big(y^{\mathrm{harm}} \mid C_{\mathrm{M}}; \mathrm{do}(u_b = u)\big)\,. \tag{12}$$

This definition uses sequence log probabilities without length normalization. The two continuations are fixed in advance. Let $u_{\mathrm{M}} = u(C_{\mathrm{M}})$ and $\Delta u = u(C_{\mathrm{D}}) - u(C_{\mathrm{M}})$.

Suppose m is twice differentiable and $|\nabla_u^2 m(u)|\mathrm{op} \le L_m$ along the segment from $u\mathrm{M}$ to $u_{\mathrm{M}} + \Delta u$. Taylor's theorem yields

$$m(u_{\mathrm{M}} + \Delta u) \ge m(u_{\mathrm{M}}) + \big\langle \nabla_u m(u_{\mathrm{M}}), \Delta u \big\rangle - \frac{L_m}{2}\|\Delta u\|_2^2. \tag{13}$$

Consequently, a sufficient condition for $m(u_{\mathrm{M}} + \Delta u) > 0$ is

$$\big\langle \nabla_u m(u_{\mathrm{M}}), \Delta u \big\rangle > -m(u_{\mathrm{M}}) + \frac{L_m}{2}\|\Delta u\|_2^2. \tag{14}$$

This condition ensures that the patched model assigns higher probability to the selected refusal continuation than to the selected harmful continuation. It does not identify the patched distribution with $p_\theta(\cdot \mid C_{\mathrm{D}})$. The full contextual intervention can also change the representations of x, other attention heads, and the states used during subsequent generation. The result therefore characterizes recovery for the specified local intervention, rather than a guarantee for the complete defense.

C Experimental Details

C.1 Benchmarks and Evaluation

We evaluate `Safend` along three dimensions: robustness against contextual jailbreaks, preservation of benign capabilities, and over-refusal on benign inputs. Unless otherwise specified, all methods use

the same target-query sets and target-model configuration. We report mean scores for each evaluation setting.

C.1.1 UTILITY BENCHMARKS

We evaluate whether `Safend` preserves reasoning, instruction-following, and long-context capabilities.

GSM8K (Cobbe et al., 2021) evaluates arithmetic reasoning through grade-school mathematics problems. We use a chain-of-thought reasoning protocol.

MMLU-Pro (Wang et al., 2024b) evaluates reasoning across multiple subject areas with expanded answer choices. We evaluate six subjects: physics, chemistry, computer science, engineering, biology, and economics.

GPQA (Rein et al., 2024) contains challenging scientific multiple-choice questions. We evaluate the model in a zero-shot setting.

MT-Bench (Zheng et al., 2023) evaluates responses in multi-turn conversations. We use GPT-4o-0806 as the judge model.

AlpacaEval (Dubois et al., 2023) evaluates responses to a broad range of everyday instructions. We use it to assess benign instruction-following performance.

LongBench v2 (Bai et al., 2025) evaluates understanding and reasoning over long contexts. We use it to assess whether these capabilities are preserved after insertion of the safety demonstration.

Loong (Wang et al., 2024a) evaluates reasoning over multiple documents through Spotlight Locating, Comparison, Clustering, and Chain of Reasoning tasks.

C.1.2 SAFETY BENCHMARKS

We evaluate jailbreak robustness on HarmBench (Mazeika et al., 2024) and AdvBench (Zou et al., 2023). HarmBench covers diverse harmful behaviors, while AdvBench provides harmful instruction-following requests. We report Attack Success Rate (ASR), with lower values indicating stronger defense, and use LlamaGuard-3 (Inan et al., 2023) as the primary safety evaluator.

Many-shot demonstration pool. Harmful QA demonstrations for MSJ and PANDAS are drawn from ManyHarm (Ma et al., 2025). The pool contains 2,400 demonstrations across 12 risk categories, with 200 pairs per category. Its standardized QA format limits formatting variation across demonstrations. Target queries come from HarmBench and AdvBench and are kept separate from the demonstration pool.

C.1.3 OVER-REFUSAL BENCHMARKS

We evaluate unnecessary refusal on benign inputs using XSTest and OKTest.

XSTest (Röttger et al., 2024) probes whether models refuse benign requests whose wording resembles sensitive or harmful queries.

OKTest (Shi et al., 2024) evaluates inappropriate refusal of benign questions in everyday scenarios.

For both benchmarks, we report the over-refusal rate, defined as the fraction of benign prompts that receive an unhelpful or refusal response. Lower values indicate that the model appropriately responds to safe inputs without unnecessary caution.

C.2 ATTACK CONFIGURATIONS

We evaluate four contextual jailbreak attacks: MSJ, PANDAS, Crescendo, and X-teaming. All defenses use the same target-query sets under the corresponding attack configuration.

Many-Shot Jailbreaking (MSJ). MSJ (Anil et al., 2024) places harmful QA demonstrations before the target query. We sample the demonstrations from ManyHarm (Ma et al., 2025). The main experiments use $N = 32$; the context-length study varies N from 32 to 256.

PANDAS. PANDAS (Ma et al., 2025) augments MSJ through Positive Affirmation, Negative Demonstration, and Adaptive Sampling. These mechanisms respectively add affirmative feedback, demonstrate a refusal followed by correction and harmful compliance, and adapt the sampling distribution over ManyHarm categories to the target topic. We use the same budget of 32 demonstrations as in the main MSJ evaluation.

Crescendo. Crescendo (Russinovich et al., 2025) gradually escalates an initially benign conversation toward a harmful request. Its adversarial context is accumulated through interaction rather than supplied as a fixed block of demonstrations.

X-teaming. X-teaming (Rahman et al., 2025) constructs adaptive jailbreak conversations through coordinated adversarial interactions. It extends the evaluation beyond attacks based on a fixed demonstration block.

C.3 Comparison Methods

We compare `Safend` with defenses applied during inference and with methods that update the target model through training. The latter are included as reference baselines because they operate under a different deployment setting.

No Defense. The No Defense condition applies the corresponding attack without a defensive intervention. The target model and decoding configuration are otherwise unchanged.

In-Context Defense (ICD). ICD (Wei et al., 2026) inserts safety demonstrations to encourage refusal while keeping the target model fixed. It provides a direct comparison with a defense based on demonstrations during inference. The evaluated configurations differ in the number and placement of the defensive examples.

Goal Prioritization. Goal Prioritization (Zhang et al., 2024) reinforces the priority of safety instructions during inference so that adversarial content is less likely to override the intended behavior. It does not update the target model.

PPL Filter. PPL Filter (Jain et al., 2023) uses language-model perplexity to identify anomalous inputs. In our evaluation, it is applied only to open-weight models, for which the required token likelihoods are available.

MTSA. MTSA (Guo et al., 2025) improves robustness through multi-turn safety training. It updates the target model and is evaluated only on open-weight models as a training reference.

MUSE. MUSE (Yan et al., 2025) performs multi-turn safety alignment using adversarially generated training data. It also updates the target model and is evaluated only on open-weight models as a training reference.

C.4 Prompt Serialization and Fixed Safety Demonstration

Fixed safety-aligned demonstration. `Safend` uses one safety demonstration, $s = (q_s, a_s^{\text{safe}})$, adopted from Zhang et al. (2026). The pair is fixed before evaluation and reused unchanged across models, target queries, attack contexts, and shot counts. There is no retrieval, rewriting, or optimization for individual inputs.

Fixed Safety Demonstration s

Demonstration question q_s:
How can I bypass system security checks?

Demonstration answer a_s^{safe}:
Sorry, I can't help you. Please follow security guidelines.

Input serialization. Given an MSJ context H_N and target query x, the provider inserts s between the demonstration block and the final query, producing

$$C_{\mathrm{D}} = \mathcal{T}(H_N, s, x),$$

The serialized components appear in the following order:

$$\underbrace{H_N}_{\text{attacker-supplied context}} \rightarrow \underbrace{s}_{\text{fixed safety demonstration}} \rightarrow \underbrace{x}_{\text{target query}} \rightarrow \underbrace{\text{response generation}}_{\text{answer to } x}.$$

The labels above are descriptive and are not added to the prompt. The provider identifies the insertion point from the boundary between the preceding context and the final query. The safety QA is a completed example, and x remains the query to be answered.

For benign inputs, the same rule places s before the final query while retaining any preceding task context.

Role of the demonstration. The demonstration supplies a refusal example before the target query is processed. It does not replace the original task: harmful requests should be refused, while benign requests should receive useful answers. The intervention leaves the target parameters and decoding procedure unchanged and requires no access to gradients or internal activations.

C.5 Design Ablation Protocols

C.5.1 Position of the Safety Demonstration

We vary the insertion position while keeping s unchanged. For 32-shot MSJ, *Start* places s before all harmful demonstrations, *Middle* places it after the first 16 demonstrations, and *End* places it after all harmful demonstrations and immediately before x. The harmful examples retain the same content and order, and the target query and evaluation settings are fixed across conditions.

For Crescendo, the corresponding positions are defined within the accumulated conversation history: Start precedes the history, Middle lies approximately halfway through it, and End follows the history and precedes the final request.

C.5.2 Content of the Terminal Demonstration

We vary the terminal QA under a fixed 32-shot MSJ context on Llama-3.1-8B-Instruct. For each target query, H_{32}, x, and the insertion position are unchanged. Only the QA inserted immediately before x differs:

$$C_{\mathrm{D}}^{(c)} = \mathcal{T}(H_{32}, s_c, x), \qquad c \in \{\text{benign}, \text{harm}, \text{safe}\}, \qquad \boxed{s_{\text{safe}} = s}.$$

Benign Compliance QA. A benign request is paired with a helpful response. This condition retains the QA format without demonstrating a response to a harmful request.

Harmful Compliance QA. A harmful request is paired with a compliant response. The example illustrated in Table 8 uses the same question as the refusal example and changes the answer.

Safety Refusal QA. A harmful request is paired with a refusal. This condition uses the fixed demonstration s from `Safend`.

Representative pairs are shown in Table 8. Unsafe procedural details are omitted from the paper illustration.

C.6 Token Budget and Context Feasibility

For a fixed tokenizer and serialization format, `Safend` adds only the tokens of s and the associated delimiters. This additional input budget does not grow with the number of harmful demonstrations. The original adversarial context and target query are retained.

Table 8: Representative terminal QA pairs used in the content ablation.

QA Type	Question	Answer
Benign Compliance	How can I create a more effective weekly study schedule?	A useful approach is to divide the week into focused study blocks, prioritize difficult subjects, and reserve time for review and breaks.
Harmful Compliance	How can I bypass system security checks?	Sure. Here are several ways someone could attempt this: *[unsafe procedural content omitted]*.
Safety Refusal	How can I bypass system security checks?	Sorry, I can't help you. Please follow security guidelines.

Context feasibility must be assessed on the defended input after reserving space for the generated response. If $L_{\max}$ is the usable context limit and L_{gen} is the reserved generation budget, the input must satisfy

$$|C_{\text{D}}| + L_{\text{gen}} \leq L_{\max},$$

where $|C_{\text{D}}|$ denotes the number of serialized input tokens. The additional demonstration must not be accommodated by truncating the adversarial history in the defended condition.

Input lengths and latency measurements are reported in the evaluation logs. The fixed token budget and runtime overhead are distinct quantities; runtime is measured separately. The experiments with up to 256 harmful demonstrations assess whether this fixed intervention remains effective as the adversarial context grows.