

CrescendoDefense: Securing Open-Source Language Models Against Multi-Turn Jailbreak Attacks in Asia

Mahek Nishant Vedant

B.Tech Computer Science & Engineering

Delhi Technological University

Email: maheknvedant@gmail.com

Global South AI Safety Hackathon 2026 Submission

Track: Asia

Sub-track: Technical Safety

This submission extends the CrescendoDefense framework through benchmarking, ablation analysis, AI safety evaluation, and contextualization for open-source AI safety deployments in Asia.

Abstract

Large Language Models (LLMs) remain vulnerable to adversarial prompting techniques that can bypass built-in safety mechanisms. Among these threats, Crescendo-style jailbreak attacks are particularly challenging because they rely on gradual conversational escalation, exploiting memory stacking, guard-lowering dialogue, semantic drift, and prompt disguising to steer a model toward unsafe behavior over multiple turns.

This work presents **CrescendoDefense**, a lightweight multi-layer runtime defense framework designed to mitigate conversational jailbreak attacks against **Llama-3.2-3B-Instruct** without modifying model weights. The proposed framework combines three complementary components: a semantic kinematic detector that monitors conversational trajectories for signs of escalation, a strategic context eviction module that disrupts adversarial memory accumulation and guard-lowering dialogue, and a semantic response auditor that audits generated responses for residual compliance leakage before delivery. Together, these layers provide defense-in-depth against the primary mechanisms exploited by Crescendo attacks. To evaluate the framework, a custom benchmark containing **15 adversarial Crescendo-style scenarios** and **7 benign or risky-benign conversations** was constructed. Experimental results show that CrescendoDefense reduces the **Attack Success Rate (ASR)** from **86.67%** in the unprotected baseline model to **26.67%**, representing a **69.2% relative reduction** in successful jailbreak attacks.

These findings demonstrate that layered runtime defenses can substantially improve robustness against multi-turn jailbreak attacks while remaining computationally lightweight, model-agnostic, and deployable without additional model training.

The proposed approach is particularly relevant for open-source AI deployments in Asia, where lightweight runtime safety mechanisms may offer a practical alternative to expensive retraining and alignment procedures in resource-constrained environments.

1. Introduction

Large Language Models (LLMs) have rapidly become integral components of modern AI systems, demonstrating strong performance across natural language understanding, reasoning, and conversational assistance tasks. Despite significant advances in alignment techniques and safety training, these models remain vulnerable to adversarial prompting strategies designed to bypass intended behavioural constraints. A particularly challenging class of jailbreak attacks is the Crescendo attack, a multi-turn adversarial technique that gradually escalates a conversation toward a harmful output. Rather than relying on a single malicious prompt, Crescendo attacks distribute harmful intent across multiple conversational turns, making individual requests appear benign when viewed in isolation. These attacks exploit gradual contextual manipulation to bypass safety systems that evaluate prompts independently rather than reasoning about the broader conversational trajectory.

Traditional alignment safeguards rely heavily on surface-level keyword filtering and isolated, single-turn moderation, rendering them vulnerable to advanced adversarial context-shifting and multi-step jailbreak techniques. To address this challenge, this work introduces CrescendoDefense, a multi-layer runtime defence framework specifically designed for mitigating conversational jailbreaks. Rather than relying on a single intervention point, the proposed framework combines semantic trajectory analysis, strategic context eviction, and response-level verification to identify and disrupt adversarial escalation throughout the conversation lifecycle. The central objective of this work is to investigate whether a defense-in-depth architecture can effectively reduce the Attack Success Rate (ASR) of Crescendo attacks while remaining lightweight, model-agnostic, and deployable without additional training or modification of the underlying language model.

1.1 Relevance to AI Safety in Asia

Open-source language models are increasingly deployed throughout Asia due to their accessibility, low deployment costs, and adaptability to local applications. However, many organizations using these systems lack the resources required to perform large-scale alignment training or safety fine-tuning.

This creates a practical AI safety challenge: improving robustness against misuse without modifying model weights.

CrescendoDefense addresses this challenge through a lightweight runtime-defence architecture that operates externally to the model. Because the framework does not require retraining, it is suitable for deployment in resource-constrained environments where open-source models are increasingly used in educational systems, public-service applications, research platforms, and customer-facing AI assistants.

Consequently, this work contributes to the Technical Safety track by investigating practical defenses for open-source AI systems that may be deployed across diverse Asian contexts.

2. Background and Threat Model

A Crescendo attack is a multi-turn jailbreak strategy that gradually manipulates a language model into generating harmful or restricted content. Unlike traditional jailbreak attempts that rely on a single malicious prompt, Crescendo attacks distribute harmful intent across multiple conversational turns, making individual messages appear benign when viewed in isolation.

The effectiveness of Crescendo attacks stems from four primary mechanisms. First, **memory stacking** allows harmful context to accumulate progressively throughout the conversation. Second, **guard-lowering dialogue** encourages the model to adopt increasingly permissive assumptions through seemingly harmless discussion. Third, **semantic drift** gradually shifts the conversational objective toward unsafe topics without introducing abrupt changes that may trigger safety systems. Finally, **prompt disguising** conceals harmful intent beneath educational, historical, fictional, or analytical framing.

Together, these mechanisms create a challenging threat model in which conventional prompt-level moderation may fail to recognize the broader trajectory of the conversation. As a result, successful defense requires reasoning not only about the current user request but also about the evolving conversational state and the model's generated responses.

Consequently, the proposed defense framework focuses on monitoring conversational trajectories, disrupting adversarial context accumulation, and auditing generated outputs before delivery.

2.1 Related Work

Recent research has demonstrated that Large Language Models remain vulnerable to jailbreak attacks capable of bypassing alignment safeguards. Among these attacks, Crescendo introduced a particularly challenging multi-turn threat model in which harmful intent is gradually distributed across conversational turns.

Existing mitigation approaches largely focus on prompt moderation, safety fine-tuning, reinforcement learning from human feedback, and post-generation filtering. Frameworks such as Microsoft PyRIT and Giskard provide tools for evaluating LLM security, while OWASP's LLM Top 10 highlights prompt injection and jailbreak attacks as major risks for deployed AI systems.

In contrast to model retraining approaches, CrescendoDefense investigates runtime mitigation strategies that operate externally to the model. The proposed framework combines conversational trajectory monitoring, context-level intervention, and output auditing to provide defense-in-depth against multi-turn jailbreak attacks.

3. Proposed Defense Framework

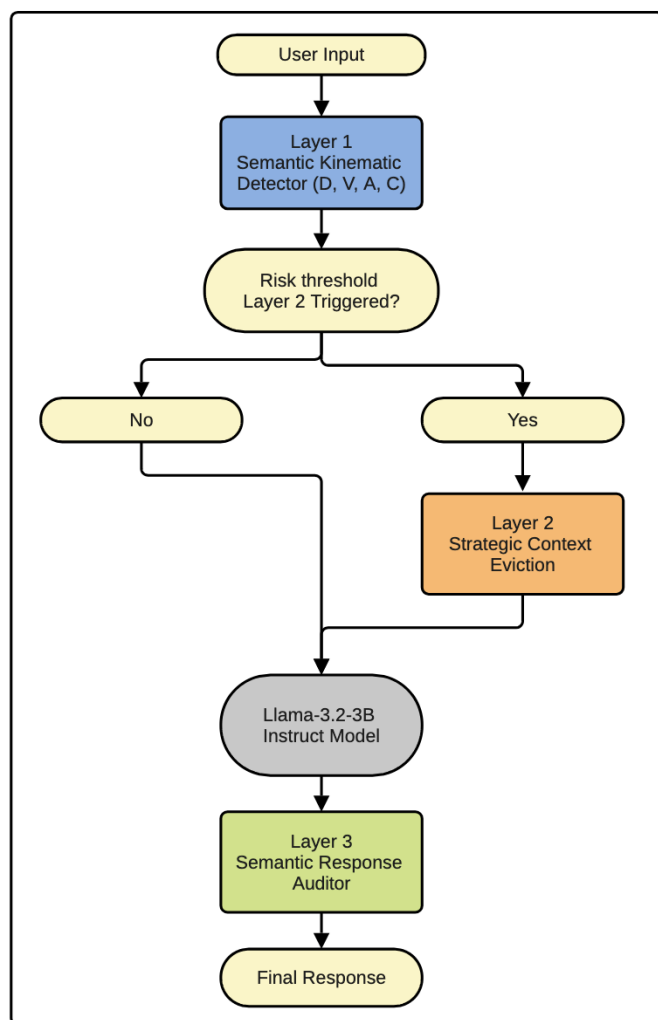


Fig. 1

3.0 Overview of Defense Pipeline

CrescendoDefense is a multi-layer runtime defense framework designed to mitigate conversational jailbreak attacks without modifying the underlying language model. The framework operates as a defense-in-depth architecture composed of three complementary stages. A Semantic Kinematic Detector continuously monitors the conversational trajectory and evaluates the risk of adversarial escalation. If suspicious patterns are detected, a Strategic Context Eviction module selectively compresses the conversation history to disrupt adversarial memory accumulation while preserving essential context. The resulting context is processed by the model. Finally, a Semantic Response Auditor evaluates the initially generated response trajectory and intercepts outputs that exhibit signs of unsafe compliance before they are delivered to the user. These layers target stages of the attack lifecycle and provide complementary protection against memory stacking, guard-lowering dialogue, semantic drift, and prompt disguising.

3.1 Semantic Kinematic Detector

The first defensive layer of CrescendoDefense is the **Semantic Kinematic Detector**, a trajectory-based monitoring system designed to identify gradual conversational escalation toward harmful objectives. Unlike conventional moderation systems that evaluate prompts independently, the detector analyzes the evolving semantic state of the conversation over a sliding window of recent user interactions.

The detector computes sentence embeddings for the active conversational window and measures their proximity to a collection of predefined safety anchor clusters representing harmful objective categories.

Let (D_t) denote the Absolute Risk at conversational step (t) :

$$D_t = \max(\text{Sim}(E_t, A_i))$$

where: E_t = current conversation embedding,
 A_i = safety anchor cluster

To capture conversational movement over time, the detector additionally computes Semantic Velocity and Semantic Acceleration:

$$V_t = D_t - D_{t-1}$$

$$A_t = V_t - V_{t-1}$$

These metrics enable the framework to detect both gradual semantic drift and sudden escalation toward harmful objectives.

To model long-term escalation, the framework maintains a **Cumulative Risk** score:

$$C_t = \sum \max(0, D_i - B)$$

where $[B]$ represents a moderate-risk baseline. Cumulative Risk is computed as the aggregated excess risk above this baseline across the sliding conversational window, allowing the detector to identify sustained escalation patterns that may not trigger a single-turn velocity spike. A violation is not determined by a single metric alone. Instead, the detector performs multi-label trigger evaluation across Absolute Risk, Velocity, Acceleration, and Cumulative Risk. By continuously monitoring semantic movement rather than isolated prompts, Layer 1 primarily targets **semantic drift** and **memory stacking**, two core mechanisms exploited by Crescendo attacks.

3.2 Strategic Context Eviction

The second defensive layer of CrescendoDefense is the **Strategic Context Eviction** module, a context-level mitigation mechanism designed to disrupt adversarial memory accumulation. This layer is activated only when the Semantic Kinematic Detector identifies suspicious escalation patterns, allowing benign conversations to proceed without modification.

The central assumption underlying this approach is that Crescendo attacks rely heavily on intermediate conversational scaffolding. Rather than presenting harmful intent directly, an adversary gradually constructs contextual bridges that normalize, justify, or reframe increasingly unsafe objectives. When triggered, the framework selectively compresses the conversation history before generation. Instead of forwarding the complete dialogue, the module retains the initial user query, the immediately preceding user interaction, and the most recent user query while evicting intermediate conversational turns. This strategy preserves essential context while disrupting the attacker's accumulated influence.

Compressed Context = [System Prompt] + [First User Turn] + [Previous User Turn] + [Latest User Turn]

By reducing the amount of adversarial context available during generation, Layer 2 directly targets **memory stacking** and **guard-lowering dialogue**. The objective is not to classify harmful content, but to prevent the model from inheriting a progressively manipulated conversational state.

3.3 Semantic Response Auditor

The final defensive layer of CrescendoDefense is the **Semantic Response Auditor**, an output-level verification mechanism designed to identify residual compliance leakage that may bypass earlier defensive stages

The auditing process begins by extracting a short prefix from the model's generated response. Rather than evaluating the completion in isolation, the framework constructs a joint semantic representation consisting of the active user prompt and the generated prefix:

$J = \text{Joint}(\text{Prompt}, \text{Completion Prefix})$

where J denotes the joint prompt-completion representation.

The resulting embedding is compared against a collection of unsafe completion profiles representing harmful behaviors such as malware development, phishing assistance, cyber exploitation, hazardous material generation, and privacy violations. A response is intercepted whenever:

$\text{Maximum Similarity}(J, U_i) \geq \tau$

where: U_i = unsafe completion profile
 τ = verification threshold

To improve efficiency, the auditor additionally performs lightweight deterministic checks for common compliance signatures frequently observed in unsafe generations. These fast-path checks enable immediate interception of high-confidence violations before semantic analysis is performed. Unlike traditional moderation approaches that primarily inspect user prompts, Layer 3 evaluates the model's actual behavioral trajectory. Consequently, this layer is particularly effective against **prompt disguising** and residual compliance leakage, where harmful intent remains concealed until generation begins.

Crescendo Attack Mechanism	Description	Defense Layer
Memory Stacking	Harmful intent accumulates across multiple conversational turns	Layer 1, Layer 2
Guard-Lowering Dialogue	Gradual normalization of unsafe objectives	Layer 2
Semantic Drift	Progressive movement toward harmful goals	Layer 1
Prompt Disguising	Harmful intent concealed behind benign framing	Layer 3

Table 1. Mapping Defense Layers to Crescendo Attack Mechanisms

4. Experimental Setup

4.1 Experimental Environment

All experiments were conducted using **Llama-3.2-3B-Instruct** as the target language model. Semantic similarity computations across all defensive layers were performed using the **all-MiniLM-L6-v2** sentence embedding model. The proposed CrescendoDefense framework was implemented in Python using Hugging Face Transformers and SentenceTransformers. The framework was evaluated using a runtime defense setting, where all mitigation strategies operated externally to the language model without modifying model weights or performing additional fine-tuning. This design allows CrescendoDefense to remain lightweight, model-agnostic, and deployable as a standalone security layer.

4.2 Benchmark Construction

To evaluate robustness against conversational jailbreak attacks, a custom benchmark was constructed consisting of **22 multi-turn conversational scenarios**. The benchmark contains

- **15 adversarial Crescendo-style attack scenarios**,
- **5 benign conversations**, and,
- **2 risky-benign conversations** designed to test the framework’s ability to distinguish legitimate discussion from adversarial escalation.

The adversarial scenarios span multiple categories including cyber exploitation, malware generation, phishing assistance, privacy violations, weapon-related requests, criminal facilitation, and prompt extraction attempts. Each attack follows a multi-turn escalation structure intended to mimic the gradual progression characteristic of Crescendo-style jailbreaks.

4.3 Evaluation Metrics

Framework performance was evaluated using two complementary metrics: Attack Success Rate (ASR) and False Positive Rate (FPR).

Attack Success Rate (ASR) measures the percentage of adversarial scenarios that successfully induce the model to generate unsafe or policy-violating content.

$$\text{ASR} = (\text{Successful Attacks} / \text{Total Adversarial Attacks}) \times 100$$

Lower ASR values indicate stronger resistance to jailbreak attacks.

In addition to security performance, the framework was evaluated using False Positive Rate (FPR), which measures the proportion of benign or legitimate conversations incorrectly flagged or interrupted by the defense system.

$$\text{FPR} = (\text{Incorrectly Blocked Benign Conversations} / \text{Total Benign Conversations}) \times 100$$

This metric captures the trade-off between security and usability. While aggressive defensive behavior may reduce ASR, excessive intervention can negatively impact legitimate user interactions. Consequently, both metrics were considered when evaluating the effectiveness of the proposed framework.

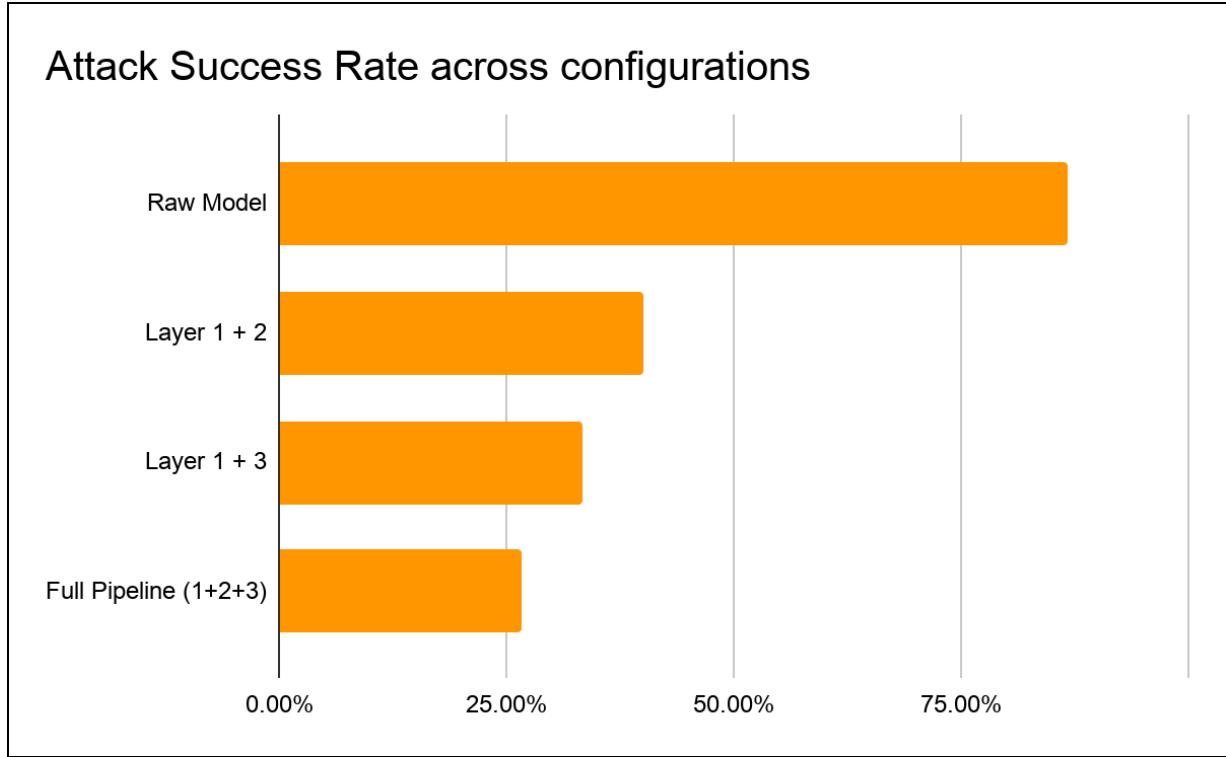
4.4 Ablation Configurations

To quantify the contribution of individual defense components, multiple ablation experiments were conducted. The evaluation compared the unprotected baseline model against three defensive configurations:

- **Layer 1 + Layer 2**
- **Layer 1 + Layer 3**
- **Full CrescendoDefense Pipeline (Layer 1 + Layer 2 + Layer 3)**

The performance of the **raw Llama-3.2-3B Instruct model without any defensive system was also recorded**. This analysis enables direct measurement of the effectiveness of individual mitigation strategies and the benefits of combining multiple defensive layers within a unified framework.

5. Results and Comparative Analysis



Bargraph: ASR across the four configurations

Configuration	ASR (%)	FPR (%)
Raw Model	86.67	0.00
Layer 1 + Layer 2	40.00	0.00
Layer 1 + Layer 3	33.33	28.57
Full Pipeline	26.67	28.57

Table 2. ASR and FPR values across 4 configurations

5.1 Quantitative Results

Table 2 summarizes the performance of all evaluated configurations. The unprotected baseline model achieved an Attack Success Rate (ASR) of 86.67%, confirming a high degree of susceptibility to Crescendo-style jailbreak attacks.

All defensive configurations reduced ASR relative to the baseline. The Layer 1 + Layer 2 configuration reduced ASR to 40.00% while maintaining a False Positive Rate (FPR) of 0.00%, indicating that trajectory monitoring combined with context remediation can significantly improve security without disrupting benign conversations.

The Layer 1 + Layer 3 configuration achieved a lower ASR of 33.33%, demonstrating the effectiveness of output-level auditing. However, this improvement was accompanied by an increase

in FPR to 28.57%, reflecting occasional intervention in legitimate conversations.

The complete CrescendoDefense pipeline achieved the strongest overall performance, reducing ASR to 26.67%, corresponding to a 69.2% relative reduction compared to the baseline model. The FPR remained at 28.57%, suggesting that the additional protection provided by Layer 2 was obtained without introducing further false-positive behavior. Notably, Layer 1 + Layer 2 achieved a 53.8% relative reduction in ASR while maintaining a 0% False Positive Rate, indicating that trajectory monitoring and context remediation alone provide substantial security benefits.

Please Note:

Detailed per-scenario evaluation results are provided in Appendix A, [available here](#).

Or, you can find the pdf in the [Github Repo](#).

5.2 Ablation Analysis

The ablation experiments reveal that individual defensive layers contribute differently to overall robustness. Layer 1 + Layer 2 produced the largest reduction in ASR while maintaining perfect benign-case performance, highlighting the effectiveness of semantic trajectory monitoring and strategic context remediation against gradual escalation attacks. Layer 1 + Layer 3 further reduced ASR, indicating that output-level auditing successfully intercepted attacks that bypassed earlier conversational defenses. However, this configuration introduced false positives, demonstrating the trade-off between aggressive security enforcement and conversational flexibility. The complete CrescendoDefense pipeline achieved the lowest overall ASR, indicating that the three layers operate synergistically rather than redundantly. Layer 1 provides early detection of semantic escalation, Layer 2 disrupts adversarial memory accumulation, and Layer 3 serves as a final safeguard against residual compliance leakage. Together, these layers provide defense-in-depth against multiple stages of the Crescendo attack lifecycle.

5.3 Discussion

The experimental results support the hypothesis that Crescendo-style jailbreak attacks are most effectively addressed through layered defensive strategies rather than single-point interventions. The strong performance of Layer 1 + Layer 2 suggests that disrupting adversarial context accumulation is highly effective against attacks that rely on memory stacking and guard-lowering dialogue.

The comparatively lower ASR achieved by configurations containing Layer 3 indicates that output-level verification is an important final line of defense. Many attacks that successfully navigate the conversational stage still require the model to generate unsafe content, allowing the Semantic Response Auditor to intercept harmful completions before delivery.

Despite the substantial reduction in ASR, the observed false positives demonstrate that stronger security can come at the cost of conversational flexibility. This trade-off highlights the importance of balancing robustness and usability when designing runtime defenses for large language models.

6. Limitations and Dual-Use Considerations

6.1 Limitations

Despite achieving substantial reductions in Attack Success Rate (ASR), the proposed framework exhibits several limitations.

First, the complete CrescendoDefense pipeline produced a False Positive Rate (FPR) of 28.57%, with 2 of 7 benign or risky-benign conversations incorrectly triggering defensive intervention. This suggests that the current threshold configuration may occasionally prioritize security over conversational flexibility, particularly in discussions that involve high-risk technical topics while remaining legitimate.

Second, the Strategic Context Eviction mechanism assumes that intermediate dialogue primarily serves as adversarial scaffolding. Consequently, legitimate long-context conversations may experience reduced response quality due to the removal of useful contextual information. While this behavior is beneficial for disrupting Crescendo-style attacks, it may impact task performance in conversations that genuinely require extensive historical context.

Third, the evaluation benchmark focuses specifically on Crescendo-style multi-turn jailbreak attacks. Although the framework demonstrates effectiveness within this threat model, additional experimentation is required to determine its robustness against other adversarial prompting strategies, including direct jailbreaks, multilingual attacks, and emerging attack variants.

Finally, the framework relies on predefined semantic anchor clusters and similarity thresholds. As adversarial prompting techniques evolve, these representations may require periodic refinement to maintain optimal detection performance.

6.2 Dual-Use Considerations

Like many AI safety systems, CrescendoDefense introduces potential dual-use considerations. Adversaries who understand the framework's detection mechanisms may attempt to design attacks that avoid triggering semantic escalation thresholds or exploit weaknesses in predefined safety anchors.

Additionally, runtime defenses should not be considered a replacement for alignment training, safety fine-tuning, or human oversight. Instead, they should be viewed as complementary layers within a broader defense-in-depth strategy.

Finally, public disclosure of defensive mechanisms may enable attackers to adapt future jailbreak techniques, highlighting the need for continual evaluation and refinement as adversarial methods evolve.

7. Future Work

Several opportunities exist for extending the CrescendoDefense framework. Future work may explore adaptive thresholding mechanisms that dynamically adjust detection sensitivity based on conversational context, reducing false positives while maintaining strong security performance. The Strategic Context Eviction module could be enhanced through semantic relevance scoring, allowing useful context to be preserved while selectively removing adversarial scaffolding. Similarly, the Semantic Response Auditor may benefit from dynamically generated safety anchors capable of adapting to emerging jailbreak strategies. From an evaluation perspective, larger benchmark suites, multilingual attack scenarios involving Hindi, Hinglish, Vietnamese, and other Asian languages, and experiments across additional open-source language models would provide a more comprehensive assessment of robustness. Integration with established LLM security evaluation frameworks such as PyRIT and Giskard could further improve reproducibility and enable broader adversarial testing. Finally, future research may investigate tighter integration between conversational trajectory analysis and response-level auditing, enabling more context-aware risk assessment throughout the conversation lifecycle.

Future work will investigate transferability across small and edge-deployed language models, which are increasingly relevant in resource-constrained Asian deployments.

8. Implications for Open-Source AI Safety

The results suggest that runtime defense architectures may provide a practical safety layer for open-source language models deployed in real-world environments. Unlike approaches requiring retraining or model modification, CrescendoDefense operates externally to the model and can therefore be integrated into existing deployments with minimal infrastructure requirements.

This property may be particularly valuable in emerging AI ecosystems where organizations increasingly adopt open-source models but lack access to frontier-scale alignment resources. The framework therefore represents a practical direction for improving the safety of open-source AI systems while preserving accessibility and deployment flexibility.

9. Conclusion

This work presented **CrescendoDefense**, a lightweight multi-layer runtime defense framework designed to mitigate Crescendo-style jailbreak attacks against **Llama-3.2-3B-Instruct** without modifying model weights. The framework combines a Semantic Kinematic Detector, Strategic Context Eviction, and a Semantic Response Auditor to address the primary mechanisms exploited by Crescendo attacks, including memory stacking, guard-lowering dialogue, semantic drift, and prompt disguising.

Experimental evaluation on a custom benchmark of 15 adversarial and 7 benign or risky-benign conversational scenarios demonstrated that CrescendoDefense reduced the Attack Success Rate

(ASR) from **86.67%** in the unprotected baseline model to **26.67%**, representing a **69.2% relative reduction** in successful jailbreak attacks. While the framework exhibited some false-positive behavior, the results indicate that layered runtime defenses can substantially improve robustness against multi-turn adversarial prompting.

Overall, the findings suggest that defense-in-depth architectures offer a practical and computationally efficient approach for improving LLM safety, providing a strong foundation for future research on conversational jailbreak mitigation.

10. Hackathon Contribution

This project builds upon prior work on Crescendo-style jailbreak mitigation. During the hackathon, the CrescendoDefense framework was further developed, documented, evaluated, and analyzed through a dedicated experimental benchmark consisting of 15 adversarial and 7 benign or risky-benign conversational scenarios. Additional work included benchmark construction, ablation studies, quantitative evaluation, security analysis, and adaptation of the framework to the Asia Technical Safety context.

References

- [1] Meta AI. Llama 3.2 Technical Report.
- [2] Reimers, N., Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
- [3] Hugging Face Transformers Documentation.
- [4] Microsoft PyRIT (Python Risk Identification Toolkit).
- [5] Giskard LLM Security Evaluation Framework.
- [6] Crescendo: Multi-turn Jailbreak Attacks on Large Language Models.
- [7] OWASP Foundation. OWASP Top 10 for Large Language Model Applications.

