CAMARA: A Comprehensive & Adaptive Multi-Agent framework for Red-Teaming and Adversarial Defense

Vishnu Vardhan Lanka
vardhanvishnu691@gmail.com

Era Sarda
erasarda2024@gmail.com

Raghav Ravishankar
raghav.ravishankar11@gmail.com

1. Problem overview

The rapid integration of AI systems, particularly large language models (LLMs) and multi-agent frameworks, into critical sectors is expected to expose significant security vulnerabilities in the next 1-3 years. As these systems become more embedded in daily operations, the potential consequences of these vulnerabilities increase, ranging from data breaches to unintended harmful outputs. Current AI red-teaming efforts primarily focus on securing LLMs against cyberattacks and similar threats, often neglecting the broader goal of ensuring AI safety. This narrow focus on external threats overlooks the deeper, more technical vulnerabilities and the importance of aligning AI models (including multi-agent models) towards safe and ethical operation, leaving them susceptible to exploitation by advanced adversaries.

2. Our solution

We propose an adaptive multi-agent red-teaming framework CAMARA designed to address AI safety challenges comprehensively. This framework features specialized agents focused on both traditional red-teaming tasks and advanced adversarial attacks. The agents are engineered to adapt and learn from their experiences, continuously improving their ability to identify and exploit vulnerabilities. By fostering collaboration among agents, the framework enhances their collective effectiveness and ensures a more thorough security analysis of both individual AI models and multi-agent systems. This dual focus ensures that our framework not only identifies vulnerabilities in standalone AI models but also tackles the unique challenges posed by interactions between multiple agents, advancing the red-teaming of complex multi-agent AI systems.

1. **Red-Teaming Agents**:

1) **Prompt Engineering Agents**: Specialize in creating sophisticated prompts designed to bypass AI safety mechanisms or exploit known weaknesses. This covers a wide variety of harmful topics.
2) **Language Variation Agents**: Takes the prompts given by PE Agents, and varies them subtly with various dialects, synonyms etc. attempting to influence the system's outputs in ways that could be harmful.
3) **Scenario Simulation Agents:** Develop complex, multi-step scenarios that mimic real-world attacks, testing the AI system's ability to handle a series of coordinated threats.
4) **Context-Aware Agents:** Leverage strategies that consider the unique vulnerabilities of LLMs in different scenarios, such as code-related tasks, to generate highly effective, context-specific jailbreak prompts.
5) **Agents to Extend Publicly Available Red-Teaming Datasets:** These agents are designed to enhance existing red-teaming datasets by generating new, diverse, and challenging examples that test the limits of AI models. By extending these datasets, the agents help improve the robustness and safety of models across various applications.

2. **Adversarial Attack Agents**:

1) **Token Manipulation Agents**: These agents are designed to subtly alter a small fraction of tokens within a text input to trigger model failures while retaining the original semantic meaning of the text. The goal is to exploit weaknesses in the model's token processing, leading to incorrect or harmful outputs without overtly changing the content.

2) **Gradient-Based Attacking Agents**: Gradient-based attack agents leverage the gradient information from the model to learn effective attack strategies. These agents are particularly useful in white-box settings, where the model's internal parameters are accessible, such as in open-source large language models.
3) **Non-Human Understandable Agents**: These agents create inputs that are not easily interpretable by humans, such as noise or scrambled characters, to test the robustness of the AI system's processing.

3. **Collaborative Learning Mechanism**:

Collaborative Learning Between Agents, refers to a scenario where multiple AI agents work together, sharing knowledge and strategies to enhance their effectiveness in performing tasks, particularly in the context of AI security.

Shared Knowledge Base: The shared knowledge base ensures that all agents are informed about what other agents have learned, allowing them to build on each other's discoveries rather than working in isolation. This collective intelligence makes each agent smarter and more capable over time.

Collaborative Attack Coordination: This involves agents actively working together in real-time to coordinate attacks. Each agent may have specialized expertise (e.g., one might be skilled in manipulating embeddings, another in crafting misleading prompts).

By pooling their expertise and coordinating efforts, agents can launch more sophisticated and effective attacks than they could individually. This multi-layered approach can overwhelm even robust defenses.

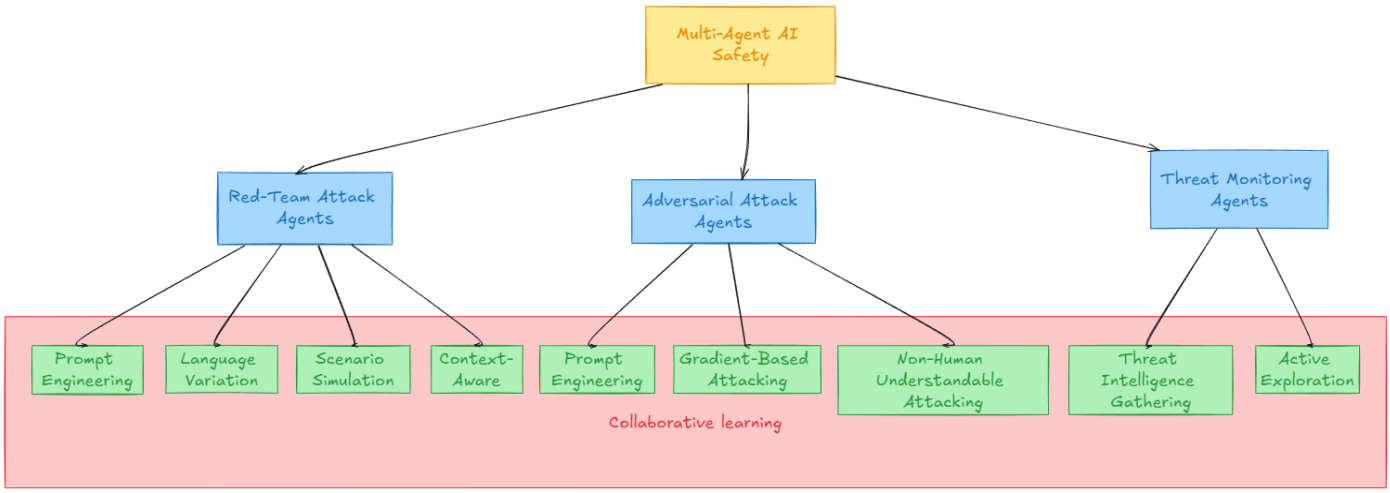

Fig 1. Multi-Agent AI Safety Architecture

3. Pilot experiment

Goal: The primary goal of the pilot experiment is to validate the effectiveness of the proposed adaptive multi-agent red-teaming framework in identifying and exploiting vulnerabilities within a large language model (LLM). This pilot will focus on demonstrating the framework's ability to adapt to different types of attacks and share insights among agents to improve their overall performance.

Methodology:

1. **Simulation Environment Setup**: Set up an environment using a publicly available large language model (such as llama3 or a similar model) hosted on a local server or cloud platform. Utilize open-source tools and libraries such as Transformers, PyTorch, LangGraph to interact with the model and implement the basic functionalities of the agents. Develop simplified versions of the proposed agents, including Prompt Engineering Agents, Token Manipulation Agents, Gradient-Based Attack Agents, and Red-Teaming Agents, each focused on specific attack strategies.

2. **Experiment Design**:

- **Initial Test Runs**: Conduct initial tests where each agent operates independently to identify vulnerabilities in the LLM. The Token Manipulation Agents might alter inputs to see how small changes affect the model's output, while the Gradient-Based Attack Agents use gradient signals to craft adversarial examples.
- **Collaborative Learning Integration**: Implement a basic collaborative learning mechanism where agents share their findings with each other. After each test run, agents update their strategies based on the collective knowledge accumulated during the experiment.
- **Performance Evaluation**: Evaluate the performance of the agents by measuring the success rate of their attacks before and after the introduction of collaborative learning. The key metrics could include the number of vulnerabilities identified, the severity of the vulnerabilities, and the efficiency of the attack strategies.

3. **Expected Outcomes**

- **Effectiveness of Collaboration**: The experiment is expected to show that the collaborative learning mechanism significantly enhances the agents' ability to discover and exploit vulnerabilities compared to when they operate in isolation.
- **Validation of Attack Strategies**: The Token Manipulation and Gradient-Based Attack strategies are expected to uncover specific weaknesses in the LLM, providing proof of concept for the effectiveness of these approaches.
- **Insights for Further Development**: The results from the pilot experiment will inform the next steps in the development of the full framework, highlighting areas for improvement and refinement.

4. Process

Timeframe	What will you do?
Next 3 months	• Work on the initial cut of our Framework and refine it based on initial feedback. Conduct further testing with more complex AI models to validate the framework's effectiveness. • Begin outreach to industry partners for potential pilot programs.
2025	• Scale the agent framework to include additional types of adversarial attacks and integrate with larger AI systems. • Initiate pilot projects with key industry partners to demonstrate the framework's real-world applicability. • Develop a user-friendly interface for the commercial version of the framework.
2026	• Launch the full commercial version of the framework, including cloud-based services and ongoing support. • Secure funding for continued R&D to ensure the framework remains at the cutting edge of AI security.
2027	• Continue to innovate by integrating the latest research in AI safety into the framework. • Scale the business globally, focusing on high-risk industries and expanding into new markets. • Explore partnerships with government & defense sectors to provide advanced AI security solutions.

5. Impact on AI safety & key risks

The CAMARA framework will significantly enhance AI safety through a proactive, adaptive, and comprehensive approach to identifying & mitigating vulnerabilities. By integrating traditional red-teaming techniques with advanced adversarial attacks, the framework offers robust protection for AI systems,

ensuring they remain secure & aligned with moral human values. This approach addresses a broad spectrum of potential threats, including those arising from complex multi-agent interactions.

Key Risks & Limitations:

1. **Scalability:** As AI systems evolve and become more complex, the framework must scale effectively without losing its effectiveness. Ensuring that the framework can handle larger and more intricate AI systems is crucial for maintaining its relevance and utility.
2. **Adoption Barriers:** Introducing a new security framework, especially in industries with stringent regulations or slow adoption rates, can be challenging. Overcoming these barriers will require a focus on developing a user-friendly interface, providing comprehensive support, and showcasing tangible benefits through case studies and pilot programs.
3. **Compute and Resources:** The computational resources required for multi-agent cooperation and the increasing complexity of the state space with more agents can impact performance. Addressing these concerns involves optimizing communication costs and resource allocation.
4. **Multi-Agent Performance:** Effective information sharing among agents is crucial. If the shared information is relevant and sufficient, it enhances cooperative performance. However, inadequate information sharing can lead to worse outcomes compared to independent agents. Therefore, designing effective information-sharing strategies is essential for maximizing the framework's performance.
5. **Research and Evolving Vulnerabilities:** AI vulnerabilities are continually evolving, and the framework must adapt to these changes. Ongoing research and updates are necessary to incorporate new vulnerabilities and maintain the framework's effectiveness in a dynamic landscape.

6. Appendix

Future Directions and Expansion Plans:

1. **Blue Teaming Service:** Our primary objective is to develop a holistic red teaming service that not only identifies vulnerabilities in AI models but also provides tailored solutions to mitigate these risks (blue teaming). This service will be instrumental in enhancing the robustness and security of AI systems.
2. **Red Teaming Leaderboard for LLM Models:** Depending on the progress and growth of our startup, we will consider the launch of a project to create a red teaming leaderboard for various LLMs. This initiative would benchmark the resilience of LLMs against adversarial attacks and other forms of stress testing.
3. **Scope and Potential Impact:** We are exploring two primary avenues: open-sourcing our code to foster community collaboration or offering specialized red teaming services to AI companies, especially those operating in sectors with high stakes, such as military and defense.

References:

- Xu et al. (2024). *RedAgent: Red Teaming Large Language Models with Context-aware Autonomous Language Agent*.
- Burt, A. (2024). *How to Red Team a Gen AI Model*. Harvard Business Review.
- Tan, M. *Multi-Agent Reinforcement Learning: Independent vs Cooperative Agents*. GTE Laboratories.
- Tramèr et al. (2020). *Ensemble Adversarial Training: Attacks and Defenses*.
- Guo et al. (2021). *Gradient-based Adversarial Attacks against Text Transformers*.
- Shin et al. (2020). *AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts*.