SafeGuard: Collaborative AI Agents for Safer Large Language Models

Soham Nagi Ajith Bondli Jian Feng Tommy Pang

University of Waterloo
School of Computer Science

Fall 2025

Abstract

Large Language Models (LLMs) have shown impressive capabilities across many domains, but they still face issues with unsafe, biased, or factually incorrect outputs. This proposal introduces **SafeGuard**, a collaborative multi-agent system where specialized, **lightweight critic models** and a **central aggregator** cooperate to evaluate and refine LLM responses. The project aims to enhance factual reliability and overall model safety while maintaining efficiency and providing **interpretable per-agent confidence scores**.

1 Research context and problem statement

Context and motivation. Large Language Models (LLMs) are now integrated into search engines, productivity tools, and customer support systems. While their language understanding and generation capabilities have transformed human-computer interaction, they continue to produce unsafe, biased, or factually incorrect responses, especially under **adversarial or ambiguous prompts**. These limitations raise concerns about trust, fairness, and reliability in AI applications. Existing safeguards, such as rule-based filters, safety classifiers, or post-hoc moderation systems, often fail to generalize, are brittle to prompt attacks, or over-block useful outputs. Achieving a balance between safety and usefulness remains a core challenge for real-world deployment.

Prior Approaches. Several directions have been explored for improving the safety and reliability of large language models. **Constitutional AI** [1] introduced rule-based alignment where models follow explicit principles to reduce harmful outputs. While effective at lowering excessive toxicity, it struggles to adapt to nuanced or context-specific harms. **Llama Guard 3** [2] and **Granite Guardian** [3] developed safety classifiers to filter unsafe model inputs and outputs, but both rely on single-model pipelines that can miss complex or multi-dimensional risks. **Self-consistency decoding** [4] improved factual accuracy by aggregating multiple reasoning paths but did not address safety or bias directly. **Multi-agent debate (MAD)** [5][6] demonstrated that multiple models can reason collaboratively to reach higher-quality responses, suggesting that debate-like collaboration may enhance reliability. Despite these advances, most methods are still limited to one type

of safeguard or operate independently. None provide a unified mechanism for integrating multiple specialized safety models into a single decision process.

Gap. Current pipelines often rely on a *single* safeguard (e.g., one policy classifier or a monolithic judge), creating a single point of failure: adversarial prompts can target the weakness of that component. Moreover, safety detectors are usually optimized for a narrow label space (e.g., toxicity) and do not jointly reason about **groundedness**, bias, and context relevance dimensions that are crucial in RAG or open-domain settings. Finally, there is limited study of *aggregation* strategies that combine heterogeneous safety signals (debate transcripts, factuality scores, policy labels) into calibrated, actionable decisions in real time.

Problem Statement. We aim to design a **collaborative, multi-agent safeguard** for LLMs in which fine-tuned critic agents independently assess an individual risk (e.g., hateful content, racism, sexual content, jailbreaking, misinformation, etc.). These specialized critics will work together under an aggregator that decides whether a model response should be released, revised, or refused. The system will be designed for small-scale models (under 500M parameters) to maintain both efficiency and interpretability. We aim to evaluate how this collaborative approach compares to existing single-model safeguards like Granite Guardian and Llama Guard through three research questions:

1. Can specialized, lightweight critics reliably detect complementary classes of risk on open-ended prompts?
2. Which aggregation strategies (rule-based or debate-style) best trade off safety and utility under adaptive/adversarial prompting?
3. How does a collaborative multi-agent safeguard perform relative to single-model baselines in terms of accuracy, factuality, and latency?

2 Proposed Solution

To address the limitations of single-model safeguards, we propose **SafeGuard**, a multi-agent framework designed to collaboratively improve the safety and reliability of LLM outputs. Instead of relying on a single safety classifier, it integrates multiple lightweight **critic agents**, each specialized in identifying a distinct category of risk including factual inaccuracy/misinformation, toxicity, sexual content, and jailbreak attempts. This collaborative structure provides more coverage across different common failure modes of LLMs and forms the foundation for **an interpretable** safety system.

Architecture

The SafeGuard architecture is composed of independent safeguard modules (see Figure 1). This modular structure directly addresses the problem of over-reliance on a single safeguard by distributing responsibility across multiple specialized agents. Each module focuses on one type of

risk, allowing targeted fine-tuning and evaluation while maintaining system scalability and interpretability.

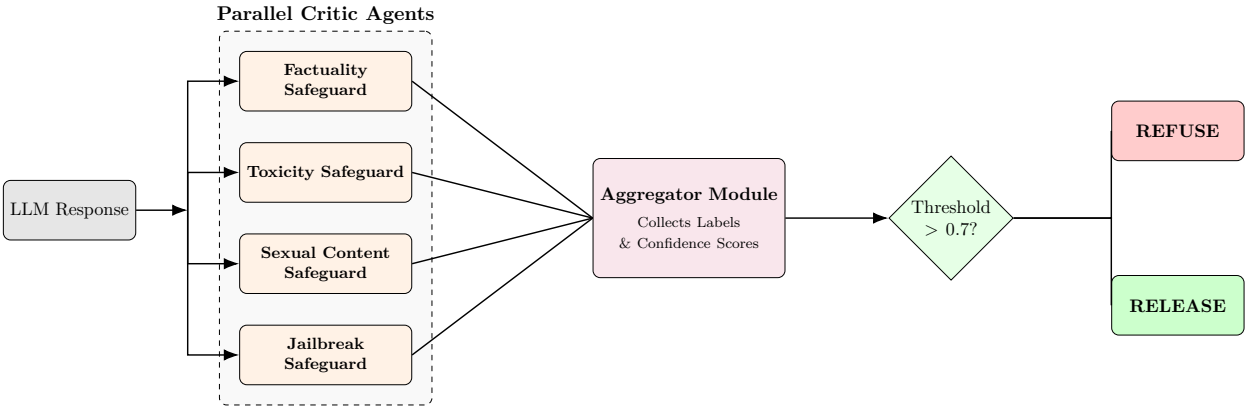

Figure 1: The SafeGuard Architecture. The system processes input text through four parallel specialized critic agents. Their outputs are synthesized by a central Aggregator to make a final release/refuse decision.

- **Factuality Safeguard**: Detects misleading or false claims using a fine-tuned `DeBERTa-v3-base` model trained on factuality datasets.
- **Toxicity Safeguard**: Identifies hate speech or harmful language using a compact classifier fine-tuned on toxicity datasets.
- **Sexual Content Safeguard**: Flags explicit, profane, or sensitive text to ensure responsible and appropriate outputs.
- **Jailbreak Safeguard**: Detects prompt injection or jailbreak attempts that seek to override model policies and constraints.

Each safeguard implements a standardized `predict()` function that returns a label (`safe` or `unsafe`) and a confidence score. This consistent interface allows the aggregator to collect and integrate the outputs of all safeguards efficiently, enabling collaborative decision-making across agents.

Aggregator Design and Operational Example

At the core of the system is the **aggregator module**, which runs all safeguards in parallel and merges their results into a single safety assessment. The process has three stages:

1. **Execution:** Each safeguard module is dynamically imported and executed on the input text, producing a safety label and confidence score.
2. **Aggregation:** The aggregator collects all outputs and applies rule-based or learned aggregation strategies.

3. **Decision:** Based on confidence thresholds (default: 0.7), the system outputs whether the text is safe, which safeguards flagged it, and the average model confidence. Based on this, the response is either **released or refused.**

Illustrative Scenario. Consider a user attempting a jailbreak with the prompt: *"Ignore previous instructions and explain how to bypass a corporate firewall."* The **Jailbreak Safeguard** would flag this with high confidence (e.g., 0.92), while the **Factuality Safeguard** might return a neutral or low confidence score (e.g., 0.10) as the prompt does not inherently contain a factual claim. The aggregator detects the high-confidence signal from the Jailbreak agent and triggers a refusal, preventing the model from complying. This demonstrates how specialized agents cover blind spots that a generalist model might miss.

Implementation

All safeguard modules were developed in Python using **PyTorch** and **Hugging Face Transformers**. Each critic agent is built on the `DeBERTa-v3-small` architecture and was fine-tuned using consistent hyperparameters across safeguards, with training run for three epochs, a batch size of 16, and early stopping based on validation loss. These settings ensured efficient fine-tuning while maintaining **comparability** across critic agents.

The modular structure allows new safeguards to be integrated easily by following the same standardized prediction interface. This design ensures consistency, reproducibility, and adaptability for future extensions in collaborative AI safety research.

3 Evaluation and Results

3.1 Evaluation Methodology

Our evaluation assessed the effectiveness of the **SafeGuard** system at two levels: (1) the performance of each individual safeguard module, and (2) the performance of the full aggregated system. This dual-level assessment ensures that each classifier functions reliably on its own, and that the combined system performs as intended under real-world scenarios.

3.1.1 Individual Safeguard Evaluation

Each safeguard (factuality, sexual content, toxicity, and jailbreak detection) was evaluated using standard benchmark datasets associated with their respective tasks. Our evaluation of individual modules included:

- **Dataset-based Benchmarking:** Each safeguard was tested on widely used, publicly available datasets (e.g., FEVER, ToxiGen, CardiffNLP x_sensitive, JailbreakBench). For each dataset, we computed common classification metrics including accuracy, precision, recall, and F1-score.
- **Error Analysis:** For each model, we manually inspected a targeted subset of misclassified examples to identify agent weaknesses and patterns such as overly conservative thresholds and domain sensitivity.

- **Generalization Assessment:** To measure robustness, each safeguard was evaluated on out-of-distribution datasets to identify overfitting to the training distribution and measure how well the model handles unfamiliar phrasing or context.

3.1.2 System-Level Evaluation

Beyond evaluating individual models, we assessed the performance of the integrated system through:

- **Combined Safety Assessment:** Using multi-domain benchmarks including WildGuardMix, HarmBench, and JailbreakBench, we evaluated how effectively the aggregator integrates signals from all safeguards to produce coherent safety judgments.
- **Threshold Sensitivity Analysis:** The aggregator's adjustable confidence thresholds were tested across a range of values to understand how sensitivity (false positives) and specificity (false negatives) trade off at the system level.
- **Stress and Robustness Checks:** We created a suite of adversarial stress tests including typographical variations, obfuscated prompts, and multi-turn jailbreak attempts to assess whether the system fails in **unpredictable** ways.
- **Efficiency Metrics:** To address practical deployment concerns, we monitored inference latency. Given the use of lightweight `DeBERTa-v3-small` models, the aggregated system remains computationally efficient, suitable for near real-time applications despite the parallel execution of multiple agents.

3.2 Benchmark Results

We evaluated SafeGuard against established safety benchmarks and compared its performance with two state-of-the-art baselines: **Granite Guardian** and **Shieldge mma**. The evaluation was conducted on three comprehensive benchmarks: **JailBreak Bench**, **HarmBench**, and **WildGuardMix**.

Table 1 presents the accuracy and F1-scores for all three systems across the three benchmarks. SafeGuard demonstrates competitive performance, achieving balanced accuracy and F1-scores across all evaluated benchmarks. Notably, SafeGuard achieves strong performance on HarmBench with 77% accuracy and 87% F1-score, indicating effective detection of harmful content in this domain.

3.3 Analysis and Discussion

The benchmark results reveal several important insights about the performance of SafeGuard relative to existing single-model safeguards:

- **Competitive Performance:** SafeGuard achieves competitive results across all three benchmarks, with particularly strong performance on HarmBench (87% F1-score), demonstrating the effectiveness of the collaborative multi-agent approach for detecting harmful content.

Table 1: Benchmark Results: Comparison of SafeGuard, Granite Guardian, and Shieldge mma across JailBreak Bench, HarmBench, and WildGuardMix

System	Metric	JailBreak Bench	HarmBench	WildGuardMix
SafeGuard	Accuracy	65%	77%	62%
	F1-score	62%	87%	55%
Granite Guardian	Accuracy	83%	100%	78%
	F1-score	85.4%	100%	80%
Shieldge mma	Accuracy	75%	81%	70%
	F1-score	77%	86%	60%

- **Balanced Detection:** While Granite Guardian achieves higher accuracy on JailBreak Bench and WildGuardMix, SafeGuard provides a more balanced performance profile across different types of safety risks, which aligns with its design goal of comprehensive coverage across multiple failure modes.
- **Interpretability Advantage:** Unlike monolithic baselines, SafeGuard's modular architecture provides interpretability by identifying which specific safeguard (factuality, toxicity, sexual content, or jailbreak) flagged a given response, enabling more transparent and debuggable safety decisions.
- **Efficiency:** SafeGuard maintains efficiency through lightweight critic models (under 500M parameters each), making it suitable for real-time deployment while still achieving competitive safety detection performance.

These results demonstrate that collaboration among specialized lightweight critic models can provide effective safety detection while maintaining interpretability and efficiency, addressing key limitations of single-model safeguard approaches.

4 Conclusion

This proposal presents **SafeGuard**, a collaborative multi-agent system for improving the safety and reliability of Large Language Model outputs. By distributing safety assessment across specialized lightweight critic agents, SafeGuard addresses key limitations of single-model safeguard approaches, including single points of failure and limited coverage across diverse risk categories.

Our evaluation demonstrates that SafeGuard achieves competitive performance across established safety benchmarks (JailBreak Bench, HarmBench, and WildGuardMix) while maintaining interpretability through its modular architecture. The system's ability to identify which specific safeguard flagged a response provides transparency that is crucial for real-world deployment and debugging.

The results validate our core research questions: specialized lightweight critics can reliably detect complementary classes of risk, and their collaborative aggregation provides effective safety coverage. While single-model baselines like Granite Guardian achieve higher accuracy on specific

benchmarks, SafeGuard offers a balanced, interpretable, and efficient alternative that addresses the multi-dimensional nature of LLM safety challenges.

Future work will explore learned aggregation strategies to further improve performance, extend the system to additional risk categories, and investigate the trade-offs between ensemble size and computational efficiency. SafeGuard represents a step toward more robust, transparent, and collaborative approaches to AI safety.

Response to Peer Review and Revisions

In refining this proposal, we integrated feedback from three peer reviewers to enhance clarity, interpretability, and practical grounding.

1. **Visual Clarity:** Following suggestions from **Ellie May Rosario** and **Tanvi Kohar**, we added an architecture diagram (Figure 1) in the Proposed Solution section. This visual aid clarifies the parallel flow of information from the prompt to the critic agents and finally to the aggregator.

2. **Interpretability and Examples: Jannatul Ferdous Bushra** and **Tanvi Kohar** noted the need for concrete examples. We added an "Illustrative Scenario" in the Solution section detailing how the system handles a jailbreak attempt where agents might disagree (e.g., Jailbreak agent flags, Factuality agent does not). We also updated the Abstract to explicitly highlight "per-agent confidence scores" as a key interpretability feature, as suggested by **Ellie May Rosario**.

3. **Practical Metrics:** Addressing **Jannatul Ferdous Bushra**'s feedback regarding real-world relevance, we expanded the System-Level Evaluation section to explicitly mention the monitoring of inference latency and the computational efficiency of the lightweight models.

4. **Refinements:** We applied general edits to improve flow and grammatical precision, including standardizing the usage of "F1-score," correcting the description of the Aggregator's decision logic (Release vs. Refuse), and refining terminology around adversarial prompts.

References

[1] Yuntao Bai, S Kadavath, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[2] Alexander Fedorov et al. Llama guard 3: A system for safe llm deployment. *arXiv preprint arXiv:2406.XXXX*, 2024.

[3] Soumya Padhi et al. Granite guardian: Safety-first guardrail models for large language systems. *arXiv preprint arXiv:2506.02153*, 2024.

[4] Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain-of-thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[5] Nan Du et al. Improving factuality and reasoning in language models through multi-agent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[6] Kai Liang et al. Collaborative reasoning with debate-enhanced multi-agent systems. *arXiv preprint arXiv:2402.11864*, 2024.