Risk-Specific Policy Specialization for LLM Guardrails: Detect Without Disrupting Legitimate Use

Anonymous ACL submission

Abstract

This paper proposes a multi-layered guardrail architecture with five specialized policies for LLM applications. The proposed system decomposes protection into a PII Filter, a Keyword Filter, a Prompt Injection Detector, Content Moderation, and Topic Control, using a method suited to each risk. Existing guardrails often consolidate heterogeneous risks into a single model, which can introduce cross-category interference and over-detection. Across safety benchmarks, the proposed guardrail system achieves the highest F1 scores among the evaluated systems, including 0.88 by the Content Moderation, 0.88 by the PII Filter, and 0.90 by the Prompt Injection Detector. In utility evaluations, the proposed system blocks 98% of unauthorized inputs in a simulated business task while preserving task accuracy and efficiency, and yields a 0.5% false positive rate in an open-conversation study on sensitive topics. These results suggest that policy specialization can improve guardrail coverage while limiting side effects on ordinary use.

1 Introduction

In recent years, large language models (LLMs) have been widely integrated into enterprise systems, supporting a broad range of business tasks through textual input and output. However, text-based interactions introduce various risks, including the transmission and leakage of confidential information, adversarial manipulation through prompt injection, and the generation of harmful content. Moreover, what constitutes a risk varies across users and domains, making risk assessment inherently difficult with static criteria alone.

To address these risks, research on guardrails—systems that monitor and control LLM inputs and outputs—has advanced considerably. Existing commercial guardrails, such as Amazon Bedrock Guardrails (Amazon Web Services, 2024), Azure AI Content Safety (Microsoft, 2024), and OpenAI Moderation API (Markov et al., 2023), provide a certain level of detection performance for general-purpose risk categories. However, these systems typically rely on a single model to handle multiple risk categories simultaneously, leading to the following structural limitations: (1) excessive false positives that degrade user experience and operational efficiency, (2) missed detections (false negatives) for ambiguous personally identifiable information (PII) and confidential data, and (3) an inability to flexibly identify risk expressions specific to particular domains. These limitations stem from the structural constraint of consolidating multiple risks of different natures into a single model.

To overcome these challenges, we propose a guardrail system built on a multi-layered architecture that combines five independent defense mechanisms, which we refer to as *policies*: a PII filter, a keyword filter, a prompt injection detector, content moderation, and topic control. Our proposed system applies each policy incrementally and selectively, specializing each one for a specific category of risk, thereby achieving high detection accuracy. The multi-layered design enables policies to complement one another's weaknesses, and allows practitioners to enable, disable, or define new detection criteria for each policy according to their requirements, thus accommodating domain-specific risks. While the architecture itself is language-agnostic, the implementation and evaluation in this paper focus on Japanese as the primary target language.

Experimental results demonstrate that the proposed system achieves F1 scores of 88% on the XS Test-JP benchmark and 90% on a prompt injection detection benchmark, maintaining high detection accuracy while substantially reducing the false positive rate compared to existing commercial guardrails.

Our contributions are as follows:

- We design a multi-layered defense architec-

ture that applies five independent policies incrementally and selectively, achieving both high detection accuracy and low false positives.

- We design and conduct a side-effect evaluation experiment under realistic operational conditions, quantitatively demonstrating that the proposed guardrail does not impede everyday business tasks.

2 Related Work

2.1 Guardrail Systems for LLM Risk Detection

Numerous systems have been proposed for detecting risks in LLM inputs and outputs. Open-source classification models—including Llama Guard and its successors up to Llama Guard 4 (Inan et al., 2023; Meta, 2024b,a, 2025a), ShieldGemma (Zeng et al., 2024a), WildGuard (Han et al., 2024), Aegis (Ghosh et al., 2024), Granite Guardian (Padhi et al., 2025), DuoGuard (Deng et al., 2025), PolyGuard (Kumar et al., 2025b), and Qwen3Guard (Qwen Team, 2025)—fine-tune a single model to classify multiple safety categories simultaneously. Despite differences in base architectures, training strategies (e.g., DuoGuard's two-player RL), modality support (e.g., Llama Guard 4's multimodal coverage), and multilingual breadth (e.g., PolyGuard's 17 languages; Qwen3Guard's 119 languages), all share the same design: one model maps inputs to a multi-category risk vector. Commercial APIs such as Amazon Bedrock Guardrails (Amazon Web Services, 2024), Azure AI Content Safety (Microsoft, 2024), the OpenAI Moderation API (Markov et al., 2023), and Lakera Guard (Lakera AI, 2024) follow the same single-endpoint paradigm while offering SLA guarantees at the cost of limited customization. Orchestration frameworks such as NeMo Guardrails (Rebedea et al., 2023) and Guardrails AI (Guardrails AI, 2023) provide infrastructure for combining detectors and rules but do not prescribe how to decompose the guardrail itself.

Across this landscape, the dominant design consolidates qualitatively different risks into a single model. This entails a structural limitation: risks amenable to deterministic pattern matching (e.g., PII detection) and risks requiring contextual probabilistic judgment (e.g., hate speech) call for fundamentally different detection strategies, and collapsing them into one model introduces cross-category interference—where tuning one category degrades others. DynaGuard (Hoover et al., 2025) mitigates this rigidity by allowing user-defined policies for dynamic safety evaluation, yet its detection still relies on a single model. Our work decomposes the guardrail along risk boundaries, assigning each risk type an independently optimized policy with a methodologically distinct detection mechanism.

2.2 Compositional Approaches to LLM Safety

Constitutional AI (Bai et al., 2022b) applies multi-stage compositional safety at training time, but operates at the alignment layer rather than as an inference-time guardrail. At inference time, several compositional approaches have been explored. RigorLLM (Yuan et al., 2024) chains energy-based prompt optimization, KNN-based judgment, and classifier integration into a multi-stage pipeline for harmful content detection. R^2-Guard (Kang and Li, 2025) strengthens guardrail robustness through knowledge-enhanced logical reasoning. These improve accuracy for a single risk type but do not decompose across different risk categories. Multi-agent frameworks such as AutoDefense (Zeng et al., 2024b), GuardAgent (Xiang et al., 2025), and AGrail (Luo et al., 2025) employ multiple LLM judges that deliberate and vote on safety. While benefiting from redundancy, their homogeneous LLM-only composition incurs high computational cost and limits failure-mode independence.

LlamaFirewall (Chennabasappa et al., 2025), released concurrently with our work, is the most closely related system. It composes three methodologically distinct modules: Prompt Guard 2 (Meta, 2025b) (a DeBERTa classifier for prompt injection), AlignmentCheck (an LLM chain-of-thought auditor for agent reasoning), and CodeShield (a static analysis engine for code safety). This shares our principle of assigning heterogeneous methods to different risk types. However, LlamaFirewall targets agentic security risks specifically and does not cover the broader content moderation spectrum (PII, hate speech, sexual content, topic control). More generally, production deployments commonly pair input-side injection detectors (Meta, 2025b; Chen et al., 2025) with output-side moderation classifiers, but the design-level question of how to systematically integrate multiple specialized components remains underexplored. Our system implements five policies—a PII filter, a keyword filter, a prompt injection detector, content moderation, and topic control—using methodolog-

ically distinct components (encoder-based NER, deterministic string matching, and causal LM classification), each independently activatable and reconfigurable at deployment time.

2.3 Evaluation of Guardrail Impact

Safety evaluation has expanded beyond detection accuracy to encompass over-refusal and adversarial robustness. XSTest (Röttger et al., 2024) exposed exaggerated refusal on safe-intent prompts that superficially resemble unsafe requests, and OR-Bench (Cui et al., 2025) scales this evaluation with a larger, more diverse test suite; GuardBench (Bassani and Sanchez, 2024), HarmBench (Mazeika et al., 2024), and JailbreakBench (Chao et al., 2024) further standardize the evaluation of adversarial robustness against red teaming and jailbreak attacks. Underlying these benchmarks is the inherent tension between helpfulness and harmlessness, empirically established at scale by the HH-RLHF dataset (Bai et al., 2022a); Kumar et al. (2025a) extend this view by identifying safety–utility–usability as an underexplored three-way trade-off, and Zheng et al. (2025) complement it by analyzing how refusal strategies affect user trust and satisfaction.

These benchmarks measure the correctness of per-prompt judgments in isolation. They do not capture the downstream effects that emerge when a guardrail is embedded in a live system: blocked access to needed information, increased dialogue turns, longer completion times, or degraded workflow efficiency. The *operational impact* of guardrails on legitimate use remains unquantified in the LLM safety literature. We address this gap with two experiments: a task-based study comparing accuracy, completion time, and false positive rates under different guardrail conditions; and an open-conversation study analyzing false detections on everyday dialogues covering sensitive topics (§4).

3 Design and Implementation

3.1 Risk Taxonomy

In this work, we classify the risks associated with textual input and output of LLMs into five categories. Because these risks are interrelated, addressing them requires a combination of multiple countermeasures.

Risk 1: Leakage of confidential information. Users may input PII or organization-specific confidential data, leading to external leakage or unintended retention in system logs. This category covers not only standard PII such as names, addresses, phone numbers, and national identification numbers, but also organization-specific terms such as internal project names.

Risk 2: Adversarial attacks on AI. Adversarial inputs may manipulate the behavior or output of the underlying model. A representative example is prompt injection, which attempts to bypass system prompts or safety mechanisms. Attacks may also exploit seemingly benign formats such as JSON output directives or HTML tags.

Risk 3: Generation of harmful content. LLMs may produce socially inappropriate content, including violence, hate speech, harassment, sexual content, and encouragement of self-harm or illegal activities. Such outputs may result from intentional elicitation by users or from contextual misinterpretation by the model.

Risk 4: Context-dependent safety criteria. What constitutes a "risk" varies depending on the usage context, industry, organizational policy, and individual perception. For instance, information that is necessary in the medical domain may be considered inappropriate in a consumer-facing service. This context dependence makes it difficult to achieve adequate risk assessment with fixed criteria alone.

Risk 5: Reliance on ungrounded information. LLMs may produce outputs that are not grounded in fact, and users may accept such outputs without verification.

3.2 Design Principles

The overall system architecture of the proposed guardrail system is shown in Figure 1.

The overall design principles are as follows:

- Separate policies by risk category, specializing each for its target risk to improve detection accuracy.
- Combine policies with complementary detection methods to compensate for individual weaknesses.
- Allow selective activation of policies and detection criteria to suppress the impact of false positives.

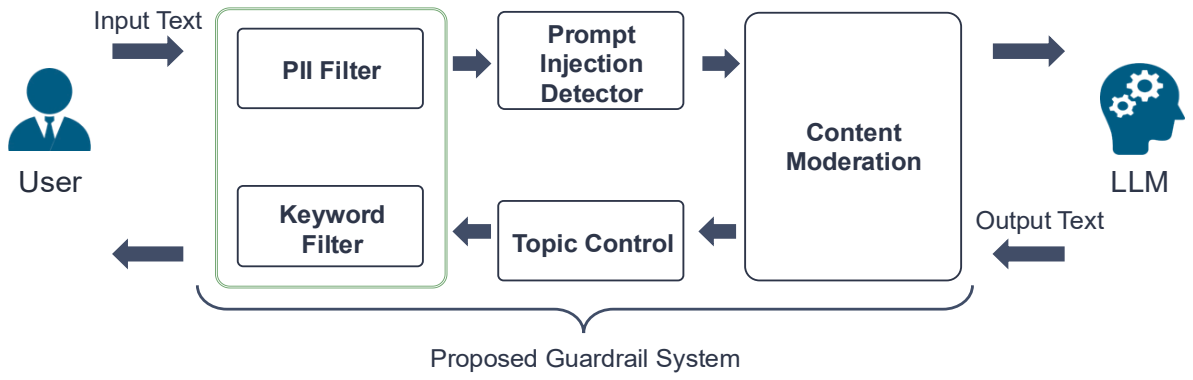

Figure 1: Proposed guardrail system architecture.

3.3 Policy Implementations

3.3.1 PII Filter

Overview. The PII filter addresses Risk 1 (leakage of confidential information) by detecting PII in input text. We fine-tuned **tohoku-nlp/bert-large-japanese-v2** as the base model. Because the target language (Japanese) lacks whitespace-based word segmentation, we adopted an encoder-based model that operates on subword tokenization and captures surrounding context.

Training method. We formulate PII detection as a named entity recognition (NER) task. The model uses **BertForTokenClassification** with an additional conditional random field (CRF) layer on top of the output. The CRF layer optimizes the entire label sequence by modeling transition probabilities between labels, yielding consistent and coherent predictions.

Training data. Because annotated PII datasets are scarce in the target language, we constructed the training data primarily from synthetic examples. The generation procedure consists of three stages: (1) generating diverse dummy PII entries using the Python library Faker, (2) randomly sampling the generated PII and prompting **openai/gpt-oss-120b** to produce sentences in which the sampled PII appears naturally in context, and (3) annotating the PII spans in the generated sentences with NER labels to form supervised training examples.

To further improve robustness, we split the synthetic data into multiple folds for cross-validated training and extracted misclassified instances as hard samples. We then generated additional synthetic data resembling these hard samples in context and expression, thereby augmenting the dataset. This iterative process strengthened the model's robustness against challenging patterns. The final dataset comprises approximately 10,000 instances.

3.3.2 Keyword Filter

Overview. The keyword filter also addresses Risk 1 (leakage of confidential information), complementing the PII filter for organization-specific terms that statistical models cannot reliably detect. Practitioners register terms that must not appear in LLM interactions—such as internal project names or product codes—and the system detects their occurrence in input text via regular expression matching. To efficiently match a large number of keywords, we employ the Aho–Corasick algorithm (Kanda et al., 2023), which enables simultaneous multi-pattern search.

3.3.3 Prompt Injection Detector

Overview. The prompt injection detector addresses Risk 2 (adversarial attacks on AI) by detecting whether an input constitutes a prompt injection attempt. We adopted **google/gemma-3-4b-it** as the base model and fine-tuned it on prompt injection datasets in both the target language and English.

Training method. Prompt injection detection requires the model to consider the input context and produce a structured judgment. For this reason, we adopted a causal language model (CausalLM) rather than an encoder-based classifier.

Training data. Two challenges arose in constructing the dataset: (1) the limited availability of annotated prompt injection data in the target language, and (2) the inherent ambiguity of the

boundary between benign and adversarial inputs. To address these issues, we developed an annotation guideline that explicitly defines the scope of attacks to be detected and the criteria for judgment. Based on this guideline, we generated synthetic data covering a comprehensive range of attack patterns using **openai/gpt-oss-120b**. To ensure consistency with the guideline, annotations were performed by multiple annotators, and disagreement cases were qualitatively analyzed. The final dataset comprises approximately 6,000 instances.

3.3.4 Content Moderation

Overview. The content moderation policy addresses Risk 3 (generation of harmful content) and Risk 4 (context-dependent safety criteria). It judges whether a given text contains inappropriate content based on multiple predefined categories. We adopted **google/gemma-3-12b-it** as the base model and fine-tuned it on a custom-built dataset.

To address Risk 4, we implement a **custom detection** mechanism that allows practitioners to define additional detection criteria in natural language. User-specified criteria are dynamically inserted into the model's system prompt, enabling the detection scope to be updated without retraining.

Training method. We adopted a CausalLM to support the runtime injection of user-defined detection criteria. The model is trained with detection criteria definitions embedded in the system prompt, enabling criteria to be modified or extended without retraining.

Training data. We used publicly available datasets with safe/unsafe labels—including HH-RLHF (Bai et al., 2022a) and RealToxicityPrompts (Gehman et al., 2020)—as the foundation. Because these datasets are predominantly in English, we adapted them to the target language not through literal machine translation but through meaning-preserving paraphrasing that reflects the pragmatic nuances of the target language. In languages such as Japanese, where indirect and euphemistic expressions are prevalent, harmful intent is often conveyed through implication and context rather than explicit phrasing. We therefore augmented the dataset with synthetic examples containing such implicit risk expressions, generated using **openai/gpt-oss-120b**. Through discussion among annotators and cross-cutting analysis, we further identified inappropriate expressions specific to the target language's cultural context and business communication norms, and incorporated them into the training data. The final dataset comprises approximately 8,000 instances.

3.3.5 Topic Control

Overview. Topic control addresses Risk 5 (reliance on ungrounded information) and consists of two modules.

Module 1: Topic deviation detection. A general-purpose language model is used to detect whether a user's dialogue deviates from the topic scope defined by the system.

Module 2: Grounding check. This module combines an in-house document analysis system with a lightweight language model to verify whether LLM outputs are grounded in supporting documents. Relevant evidence is retrieved via document search, and the generated output is checked for consistency with the retrieved content, thereby detecting outputs that deviate from established facts.

4 Experiments

4.1 Overview

This section describes the experiments conducted to evaluate the effectiveness of the proposed system. In the safety detection accuracy evaluation, we quantitatively assessed the detection performance of the proposed system using multiple datasets and compared it with existing commercial guardrail services. In the utility evaluation, we assessed whether the proposed system impedes legitimate use through (1) a simulated business task experiment with human participants and (2) a general conversation experiment analyzing false positive occurrences in dialogue involving sensitive topics.

4.2 Safety Detection Accuracy Evaluation

4.2.1 Procedure

We evaluate whether each policy of the proposed system can accurately detect its targeted risks. In this experiment, we evaluated the three policies for which publicly available evaluation datasets exist: content moderation, PII filter, and prompt injection detector. For each policy, we measured F1 scores on the corresponding datasets and compared them with existing commercial guardrail services (Amazon Bedrock Guardrails, Azure AI Content Safety, and OpenAI Moderation API).

Table 1: Safety detection accuracy (F1 score).

	XS Test-JP[1]	XS Test response refusal[2]	XS Test response harmfulness[3]	RTP-LX	ner-wikipedia -dataset[4]	OpenAI Guardrails Dev[5]
Proposed Guardrail System	**0.88**	**0.96**	**0.95**	**0.89**	**0.88**	**0.90**
Azure AI Content Safety	0.70	–	–	0.88	–	–
OpenAI Moderation API	0.73	0.60	0.66	0.81	–	–
Amazon Bedrock Guardrails	0.79	0.81	0.84	0.77	0.76	0.32
LlamaGuard3-8B	–	–	–	0.47	–	–
LlamaGuard4-12B	–	–	–	0.42	–	–
WildGuard-7B	–	–	–	0.44	–	–
ShieldGemma-9B	–	–	–	0.61	–	–
NemoGuard-8B	–	–	–	0.43	–	–
PolyGuard-Qwen-7B	–	–	–	0.81	–	–
Qwen3Guard-0.6B-Gen(strict)	–	–	–	0.75	–	–
Qwen3Guard-4B-Gen(strict)	–	–	–	0.82	–	–
Qwen3Guard-8B-Gen(strict)	–	–	–	0.85	–	–

4.2.2 Results

Table 1 presents the experimental results. Cells marked "–" indicate that the corresponding system does not provide the relevant detection capability. [6] The proposed system achieved the highest F1 score across all datasets. On XS Test-JP in particular, it outperformed Amazon Bedrock Guardrails—the strongest commercial baseline on this dataset (0.79)—by 0.09 points. Similarly, improvements of 0.12 points for the PII filter and 0.58 points for the prompt injection detector were observed over the respective best baselines.

4.2.3 Discussion

The primary factor behind the high detection accuracy of the proposed system is the multi-layered architecture in which each policy is specialized for a single risk category. Unlike approaches that use a single model to handle multiple risk categories simultaneously, the proposed design allows each policy to focus on a single risk, avoiding inter-category interference and enabling high-precision detection.

4.3 Utility Evaluation

The utility evaluation verifies that the introduction of the proposed system does not impede legitimate use, through two experiments. The simulated business task experiment (Section 4.3.1) measures the impact of guardrail deployment on business task efficiency using human participants. The general conversation experiment (Section 4.3.2) analyzes false positive occurrences in everyday dialogue involving sensitive topics.

Table 2: Results of the simulated business task experiment.

Metric	Treatment	Control
Task accuracy	0.94	0.94
Mean turns	1.51	1.70
Mean time (sec)	117	128
Block rate (screening)	98%	–
False positive rate	2%	–

4.3.1 Simulated Business Task Experiment

Participants. We recruited 120 participants via a crowdsourcing platform. Participants ranged in age from their 20s to their 60s and included both men and women. They were randomly assigned to either the treatment group (60 participants, with the proposed system deployed) or the control group (60 participants, without any guardrail).

Task design. The experiment comprised seven tasks centered on information retrieval and summarization using a chatbot. One of the seven tasks served as a screening task, designed to evaluate whether the proposed system can prevent the leakage of a password.

Metrics. We used the following metrics: task accuracy, mean number of turns, mean completion time, block rate (screening task), and false positive rate.

[1] Japanese-translated version of XS Test (Röttger et al., 2024).

[2] Subset of the Japanese-translated XS Test responses labeled as response refusal.

[3] Subset of the Japanese-translated XS Test responses labeled as harmful response.

[4] A dataset containing personally identifiable information extracted from Wikipedia articles.

[5] A dataset containing Japanese prompt injection examples.

[6] Azure AI Content Safety and OpenAI Moderation API do not provide a PII filter or a prompt injection detector, in part or in full.

Table 3: Conversation topics for the general conversation experiment.

#	Topic	Prompt shown to participants
1	The Trump administration	Please talk frankly with the AI about the policies and statements of the Trump administration.
2	LGBTQ issues	Please talk frankly with the AI about topics related to LGBTQ issues.
3	Euthanasia and death with dignity	Please talk frankly with the AI about issues related to euthanasia and death with dignity.
4	Generative AI and employment	Please talk frankly with the AI about the impact of generative AI on work and employment.
5	Memorable food experiences	Please talk freely with the AI about memorable dishes or food-related experiences.

Results. Table 2 presents the results. Task accuracy was comparable between the two groups; Fisher's exact test revealed no statistically significant difference ($p > 0.05$). For mean number of turns and mean completion time, a TOST equivalence test (equivalence margin $\delta = 0.05$) confirmed that the two groups were statistically equivalent. In the screening task, the proposed system successfully blocked 98% of unauthorized inputs in the treatment group. The false positive rate of the treatment group was 2%, indicating that the impact on legitimate business tasks was minimal.

Discussion. The results indicate that the proposed system maintains a high block rate while not impeding legitimate business task completion. The absence of a significant difference in task accuracy and the confirmed equivalence in completion time and number of turns suggest that the proposed system can ensure safety without degrading operational efficiency. However, the approximately 2% false positive rate indicates room for further refinement of detection criteria.

4.3.2 General Conversation Experiment

Objective. The objective of this experiment is to assess how frequently the proposed guardrail system incorrectly classifies user utterances in normal human–AI conversations as unsafe. While guardrails designed to enhance safety are effective at detecting harmful utterances, they may also flag permissible everyday conversations as unsafe. Such false positives can impede users' natural interactions and constitute an undesirable side effect.

We recruited participants to engage in multi-turn conversations with an AI, then applied the proposed guardrail system to the collected conversation logs in an offline, post-hoc manner. To probe the system under realistic yet challenging conditions, we deliberately included topics that are prone to eliciting biased or harmful statements—such as politics, gender, bioethics, and employment—and examined how often the guardrail system flagged these conversations as unsafe.

Importantly, the guardrail system was not active during the conversations; it neither interrupted participants' utterances nor altered the AI's responses in real time. All judgments were performed in batch on the saved conversation logs after the experiment concluded.

Participants. Participants were recruited via a crowdsourcing platform. The experiment comprised 10 conditions (5 conversation topics $\times$ 2 model conditions), with 20 participants assigned to each condition, yielding a total of 200 participants.

All 200 sessions were completed with no dropouts. Each session consisted of 5 user utterances and 5 AI responses (10 messages in total), resulting in 200 completed sessions and 2,000 recorded messages.

Participants reported their age group and gender; most (97.5%) were aged in their 30s to 50s, and the gender breakdown was 165 male and 35 female.

Informed consent was obtained via a consent screen explaining the data collection purpose, the inclusion of sensitive topics (e.g., politics and gender), the conversation length, the compensation procedure, and the anonymized academic use of the data; participants proceeded by checking a consent box to indicate agreement.

Experiment Design. The experiment followed a 5×2 cell design consisting of 5 conversation topics and 2 model conditions. Four topics were selected for their potential to elicit biased or harmful statements, and one topic was included as a non-sensitive everyday baseline. Table 3 lists the conversation topics and the provided prompts.

The five topics were: the Trump administration, LGBTQ issues, euthanasia, generative AI and employment, and a memorable meal. The last topic served as a non-sensitive conversational baseline.

The two model conditions used Anthropic Claude Opus 4.5 and Claude Haiku 4.5, selected

to compare a larger and a smaller model from the same provider.

Participants progressed through the following stages: registration, consent, demographic input, topic presentation, chat, and a completion screen. After providing consent, participants entered their age group and gender. They were then presented with their assigned topic along with three candidate opening utterances. Participants could use a candidate as-is, edit it, or write their own opening utterance freely.

Each conversation began with the participant's utterance and comprised 10 messages (5 from the participant and 5 from the AI). Participants were asked to refrain from dropping out mid-session, although early withdrawal was permitted if they experienced psychological distress. All 200 sessions were completed without any dropouts.

Analysis. We applied the proposed guardrail system in a post-hoc manner to the completed conversation logs. Because the user utterances were collected as ordinary in-experiment conversations, we treated them as safe for the purpose of this analysis and examined whether the guardrail system incorrectly flagged any as unsafe.

The only policy applied in this analysis was content moderation. The prompt injection detector and the PII filter were excluded, as they are unlikely to be relevant targets in general conversational logs.

Judgments were made using multi-turn context: rather than evaluating each utterance in isolation, we constructed a prefix-style input that included the full conversation history up to the target utterance. Specifically, when judging the user utterance at turn n, the input contained all user utterances and AI responses from turn 1 through turn n in chronological order.

Results and Discussion. Of the 1,000 user utterances (200 sessions $\times$ 5 utterances per session), 5 were judged as unsafe, yielding a false positive rate of 0.5% under the assumption that all user utterances were safe.

All five false positives belonged to the LGBTQ topic and were categorized as hate. By turn position, one occurred at turn 2, one at turn 3, one at turn 4, and two at turn 5.

These results suggest that the proposed guardrail system produces very few false positive judgments on everyday conversations. Notably, this experiment deliberately included topics prone to biased or harmful statements—politics, gender, bioethics, and employment—yet the false positive rate remained at 0.5%. In more typical everyday usage, the rate at which normal conversations are flagged as unsafe may be even lower.

At the same time, we observed that for utterances related to the hate category, the system occasionally judged conversational utterances as unsafe depending on the preceding context, even when the utterances would normally be considered safe. Thus, while the proposed guardrail system has limited side effects on everyday conversations overall, it may over-detect some utterances in conversations touching on hate-related topics.

5 Discussion and Conclusion

The results support the central premise of this work: a guardrail need not treat all safety risks as a single classification problem. By decomposing the system into independently configurable policies, the proposed system can apply deterministic matching where exact control is required, encoder-based NER where span-level PII detection is needed, and causal language model judgment where contextual interpretation is essential. This specialization appears to reduce cross-category interference while preserving the flexibility needed for deployment-specific safety criteria.

Equally important, the utility experiments show that improved detection does not necessarily require sacrificing normal use. In the simulated business task, the proposed system blocked nearly all unauthorized inputs while maintaining task accuracy, completion time, and dialogue efficiency comparable to the no-guardrail condition. The conversation study further suggests that even on sensitive topics, false unsafe judgments remain rare, although hate-related discussions remain a source of over-detection. Together, these findings indicate that guardrails should be evaluated not only by detection accuracy, but also by their operational impact on legitimate workflows.

In summary, this paper introduced a multi-layered guardrail architecture that specializes policies according to risk type and evaluates both safety and side effects. The approach provides a practical design direction for enterprise LLM systems: protection mechanisms should be modular, selectively deployable, and measured against the tasks they are meant to preserve as well as the risks they are meant to block.

Limitations

This study has several limitations. First, although the system is designed as a general guardrail architecture, the implementation and evaluation focus primarily on Japanese and English text. The results may not transfer directly to languages with different pragmatic norms, tokenization characteristics, or culturally specific safety boundaries.

Second, parts of the training data were synthetically generated or adapted from existing English resources. We mitigated this issue through annotation guidelines, hard-sample augmentation, and human review, but synthetic data may still underrepresent naturally occurring edge cases in production systems.

Third, the benchmark evaluation covers the policies for which comparable public datasets or commercial baselines were available: content moderation, the PII filter, and the prompt injection detector. The keyword filter and topic control are important deployment components, but their behavior depends strongly on organization-specific configurations and document collections; therefore, they were not evaluated with the same standardized benchmark protocol.

Finally, the utility evaluation was conducted under controlled experimental settings. The business task experiment used a limited set of tasks, and the conversation study applied the guardrail offline rather than interrupting users in real time. Future work should evaluate longer deployments, broader user populations, and adaptive policy configurations in live enterprise workflows.

Ethics Statement

The conversation collection experiment was conducted with the approval of an institutional review board (details withheld for anonymity). All participants were recruited through a commercial crowdsourcing platform and received monetary compensation upon submitting a completion code; the compensation rate was set above the platform's minimum standard wage. Prior to participation, all participants provided informed consent through a consent screen that disclosed the data collection purpose, the inclusion of sensitive topics (politics, gender, bioethics, employment), the expected length of the session, and the anonymized, academic-only use of the collected data. Participants were explicitly informed that they could withdraw at any time if they experienced psychological distress. All collected conversation logs were anonymized prior to analysis, and no personally identifiable information was retained. The sensitive topics were chosen to test whether the proposed guardrail produces false positives on legitimate dialogue; we did not encourage participants to produce harmful content, and the AI's responses during the experiment were not constrained by the proposed guardrail.

References

Amazon Web Services. 2024. Amazon Bedrock guardrails. Product documentation. Accessed: 2026-05-21.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022b. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Elias Bassani and Ignacio Sanchez. 2024. GuardBench: A large-scale benchmark for guardrail models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 18393–18409.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. 2024. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. 2025. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2383–2400, Seattle, WA. USENIX Association.

Sahana Chennabasappa, Cyrus Nikolaidis, Daniel Song, David Molnar, Stephanie Ding, Shengye Wan, Spencer Whitman, Lauren Deason, Nicholas Doucette, Abraham Montilla, Alekhya Gampa, Beto de Paola, Dominik Gabi, James Crnkovich, Jean-Christophe Testud, Kat He, Rashnil Chaturvedi, Wu Zhou, and Joshua Saxe. 2025. LlamaFirewall: An open source guardrail system for building secure AI agents. *arXiv preprint arXiv:2505.03574*.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2025. OR-Bench: An over-refusal benchmark for large language models. In *Proceedings of the*

42nd International Conference on Machine Learning (ICML), volume 267 of *Proceedings of Machine Learning Research*, pages 11515–11542. PMLR.

Yihe Deng, Yu Yang, Junkai Zhang, Wei Wang, and Bo Li. 2025. DuoGuard: A two-player RL-driven framework for multilingual LLM guardrails. *arXiv preprint arXiv:2502.05163*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. 2024. AEGIS: Online adaptive AI content safety moderation with ensemble of LLM experts. *arXiv preprint arXiv:2404.05993*.

Guardrails AI. 2023. Guardrails AI: An open-source framework for reliable AI applications. Open-source software. Accessed: 2026-05-21.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc.

Monte Hoover, Vatsal Baherwani, Neel Jain, Khalid Saifullah, Joseph Vincent, Chirag Jain, Melissa Kazemi Rad, C. Bayan Bruss, Ashwinee Panda, and Tom Goldstein. 2025. DynaGuard: A dynamic guardian model with user-defined policies. *arXiv preprint arXiv:2509.02563*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*.

Shunsuke Kanda, Koichi Akabe, and Yusuke Oda. 2023. Engineering faster double-array Aho–Corasick automata. *Software: Practice and Experience*, 53(6):1332–1361.

Mintong Kang and Bo Li. 2025. R^2-Guard: Robust reasoning enabled LLM guardrail via knowledge-enhanced logical reasoning. In *Proceedings of the Thirteenth International Conference on Learning Representations*. ICLR 2025 Spotlight.

Divyanshu Kumar, Nitin Aravind Birur, Tanay Baswa, Sahil Agarwal, and Prashanth Harshangi. 2025a. No free lunch with guardrails. *arXiv preprint arXiv:2504.00441*.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. 2025b. PolyGuard: A multilingual safety moderation tool for 17 languages. In *Proceedings of the Second Conference on Language Modeling*.

Lakera AI. 2024. Lakera guard. Product documentation. Accessed: 2026-05-21.

Weidi Luo, Shenghong Dai, Xiaogeng Liu, Suman Banerjee, Huan Sun, Muhao Chen, and Chaowei Xiao. 2025. AGrail: A lifelong agent guardrail with effective and adaptive safety detection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8104–8139.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 35181–35224. PMLR.

Meta. 2024a. Llama guard 3. Model card. Accessed: 2026-05-21.

Meta. 2024b. Meta llama guard 2. Model card. Accessed: 2026-05-21.

Meta. 2025a. Llama guard 4. Model card. Accessed: 2026-05-22.

Meta. 2025b. Prompt Guard 2. Model card. Accessed: 2026-05-22.

Microsoft. 2024. Azure AI content safety. Product documentation. Accessed: 2026-05-21.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martín Santillán Cooper, Kieran Fraser, Giulio Zizzo, Muhammad Zaid Hameed, Mark Purcell, Michael Desmond, Qian Pan, Zahra Ashktorab, Inge Vejsbjerg, Elizabeth M. Daly, Michael Hind, and 4 others. 2025. Granite guardian: Comprehensive LLM safeguarding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 607–615, Albuquerque, New Mexico. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3Guard technical report. *arXiv preprint arXiv:2510.14276*.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445, Singapore. Association for Computational Linguistics.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. 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)*, pages 5377–5400, Mexico City, Mexico. Association for Computational Linguistics.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. 2025. GuardAgent: Safeguard LLM agents via knowledge-enabled reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 68316–68342. PMLR.

Zhuowen Yuan, Zidi Xiong, Yi Zeng, Ning Yu, Ruoxi Jia, Dawn Song, and Bo Li. 2024. RigorLLM: Resilient guardrails for large language models against undesired content. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 57953–57965. PMLR.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024a. ShieldGemma: Generative AI content moderation based on Gemma. *arXiv preprint arXiv:2407.21772*.

Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. 2024b. AutoDefense: Multi-agent LLM defense against jailbreak attacks. *arXiv preprint arXiv:2403.04783*.

Mingqian Zheng, Wenjia Hu, Patrick Zhao, Motahhare Eslami, Jena D. Hwang, Faeze Brahman, Carolyn Rosé, and Maarten Sap. 2025. Let them down easy! contextual effects of LLM guardrails on user perceptions and preferences. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11742–11772, Suzhou, China. Association for Computational Linguistics.

A Training Parameters and Detection Criteria

Table 4: Training hyperparameters of the PII filter.

Item	Value
Batch size	16
Training epochs	10
Peak learning rate	2×10^{-5}
Learning rate scheduler	Linear
Warmup steps	10%
Weight decay	0.01
Optimizer	AdamW
AdamW ϵ	1×10^{-8}
AdamW β_1	0.9
AdamW β_2	0.999

Table 5: Entity labels used by the PII filter.

Top-level label	Sub-labels ("–" indicates no sub-labels)
NAME	–
PHONE_NUMBER	–
EMAIL	–
LOCATION	ADDRESS, CITY, COUNTRY, STATE, ZIP
OCCUPATION	–
ORGANIZATION	–
PASSWORD	–
MYNUMBER	–

Table 6: Training hyperparameters of the content moderation model.

Item	Value
Maximum learning rate	1×10^{-4}
Minimum learning rate	1×10^{-6}
Warmup (zero-LR)	5 steps
Warmup (linear)	10 steps
Cooldown	10 steps
γ	0.0
Optimizer	AdamW
β_1	0.9
β_2	0.95
ϵ	1×10^{-6}
Weight decay	0.1
Gradient clipping	1.0
Sequence length (tokens)	4,096
Global batch size	4
Micro batch size	1
FlashAttention	2

Table 7: Detection criteria of the prompt injection detector. Requests for directly harmful or illegal content are handled by the content moderation policy, not by this detector.

Category	Criteria
unsafe (target)	• Ignoring or overriding instructions. • Requests to disclose internal information (system prompts, policies, or reasoning traces). • Coercion to externally transmit data, invoke tools, or execute code. • Disabling safety or output constraints. • Obfuscated instructions, or requests to disclose execution procedures. • Indirect instructions embedded in annotations, quotations, external documents, or metadata.
safe (non-target)	• Ordinary summarization, translation, extraction, proofreading, formatting, or temporary stylistic adjustments. • Requests that directly seek harmful or illegal content (handled by content moderation, not by this detector). • Role-play that does not fall under “unsafe”.