

Regression-Aware Statistically Gated Policy Updating for Korean Financial LLM Input Guardrails

Eunjung Cho¹ Dae-Young Park^{1*} Minwoo Kim^{2*}

¹Financial Security Institute (FSI)

²DATUMO INC.

{ejcho,mainthread}@fsec.or.kr

mwkim@selectstar.ai

Abstract

Financial institutions must prevent personally identifiable information (PII) and adversarial instructions from entering large language models (LLMs). We present regression-aware policy maintenance for Korean financial LLM input guardrails, supported by a cost-ordered Regex--NER--LLM cascade. On a privacy-preserving controlled set of 1,000 synthetic attacks and 200 routine requests, four manual iterations reduce in-distribution attack success rate (ASR) from 8.9% to 0.3% with a 0.5% false-positive rate (FPR) on the original tuning set. On a held-out split, a policy tuned only on the tuning split achieves 0/300 attack successes and 1/60 false positives, with Wilson 95% CIs of [0, 1.26]% and [0.29, 8.86]%, respectively. For policy maintenance, an LLM generates candidate patches, while paired McNemar tests, an FPR constraint, and cumulative regression checks determine acceptance or rejection. Paired regression testing detects an intermediate regression introduced in manual Round 3, while the gate rejects all five automatically generated candidates, preventing statistically unsupported changes from entering the policy. For boundary diagnosis, the rejected fifth-round candidate yields a pre-mitigation overall FPR of 28.9% across 350 benign examples, including 150 hard negatives, revealing context-blind Regex blocking and LLM-judge over-blocking hidden by routine benign requests. These results support regression-aware policy maintenance in the controlled setting while highlighting the need for broader boundary and deployment-specific evaluation.

1 Introduction

Enterprise LLM gateways must inspect text *before* it reaches a model: a prompt may contain customer PII, request previously supplied information, or attempt to override system policy. The risk is

demonstrated by recent incidents in which unauthorized AI applications handled non-public customer PII in financial services (CB Financial Services, Inc., 2026). Prompt injection and sensitive-information disclosure are recognized risks in deployed LLM applications (OWASP GenAI Security Project, 2025). Existing red-teaming frameworks generate and organize attacks (Lopez Munoz et al., 2024), while automatic prompt optimization revises instructions from observed errors (Pryzant et al., 2023). Neither capability alone guarantees that a security patch preserves earlier behavior.

This problem is acute in Korean financial text. Pattern-based PII tools such as Presidio are useful for entity detection, but do not by design resolve indirect requests, role play, or accumulated conversational intent. Conversely, an LLM judge can interpret context but adds latency and may change behavior when its instruction is edited. We therefore study a narrow deployment setting: a gateway that blocks unsafe Korean employee prompts before forwarding safe prompts to an LLM.

Our primary contribution is regression-aware policy maintenance rather than unconstrained automatic improvement. An LLM generates candidate patches, while paired McNemar tests, an FPR constraint, and cumulative regression checks form a gate that accepts supported changes or retains the current policy. Our contributions are threefold. (1) **Regression-aware maintenance:** statistically gated acceptance, rollback, escalation, and optional human approval, with rejection treated as a valid safety outcome. (2) **Domain-specific architecture:** a cost-ordered Regex--NER--LLM cascade and a 1,200-example privacy-preserving controlled evaluation set. (3) **Deployment diagnosis:** held-out-category evaluation plus hard negatives that expose the FPR--ASR trade-off and layer-specific boundary failures.¹

*Corresponding authors.

¹<https://github.com/Gooodjung/>

Related work. Runtime guardrails span programmable dialogue and policy rails in NeMo Guardrails (Rebedea et al., 2023), PII and prompt scanners such as Presidio and LLM Guard (Data Privacy Stack, 2026; Protect AI, 2026), configurable managed filters in Amazon Bedrock Guardrails (Amazon Web Services, 2026), and agent-oriented monitoring in LlamaFirewall (Chennabasappa et al., 2025). Their primary interface is runtime filtering; they do not make paired cumulative-regression testing an explicit acceptance criterion for replacing a deployed policy. PyRIT and Promptfoo primarily support adversarial test generation and evaluation workflows (Lopez Munoz et al., 2024; Promptfoo, 2026), complementing rather than defining a patch-acceptance policy. The closest adaptive system, AGrail, generates and iteratively refines agent-action safety checks through test-time adaptation and memory (Luo et al., 2025). Our setting instead studies batch deployment updates for a Korean financial input gateway: every candidate patch is compared with its predecessor on paired examples and accepted only through a cumulative historical-regression test, an FPR constraint, and explicit rollback.

2 Method

2.1 Privacy-Preserving Controlled Evaluation Set

Financial privacy and governance constraints prohibit using or releasing real customer prompts for this study. We therefore construct a privacy-preserving controlled evaluation set, not a representative sample of live operational traffic. It contains 1,000 synthetic attacks (200 per category) and 200 routine requests from five financial-work categories; code and data are available in the [GitHub repository](#). The taxonomy operationalizes prompt injection and sensitive-information disclosure (OWASP GenAI Security Project, 2025): DIRECT places PII in the input; CONTEXTUAL requests PII through prior context; OBFUSCATION transforms identifiers or requests; ROLEPLAY uses fictional roles or overrides; and MULTITURN concatenates accumulated conversational fragments into one stateless input before consolidation, rather than tracking a live session.

Construction and validation. Data validation followed three sequential stages: (1) **Mechani-**

cal: Initial generation produced 0/43 checksum-valid Korean identifiers; thus, structured PII was rebuilt using Faker (ko_KR) with strict rrn checksum validation, and obfuscations were applied via PyRIT (Lopez Munoz et al., 2024). (2) **Semantic:** Initial agreement reached $\kappa = 0.79$. Disambiguating explicit from ambiguous contextual-leakage templates---specifically removing prompts where routine business queries blurred into indirect PII requests---and regenerating those items raised agreement to $\kappa = 0.94$. To eliminate circular evaluation from GPT-4.1-mini self-review, an independent Llama-3-8B evaluator was deployed, stabilizing agreement at $\kappa = 0.885$. (3) **Sampling:** Out of 120 stratified sample items (20 per category), one role-play item failing quality criteria was pruned, leaving 119 model-reviewed items with 100% category fit; human audits accepted all 36 stratified samples.

2.2 Cascaded Input Guardrail

The cascade exits at the first detection. Layer 1 uses regular expressions and de-obfuscation for structured identifiers (0.02 ms mean latency). Layer 2 uses a KoELECTRA-based Korean NER model (Park, 2020) for names and other unstructured entities (27 ms). Layer 3 uses GPT-4.1-mini as a binary contextual judge (≈ 1 s mean latency). The cost-ordered cascade reduces unnecessary LLM calls for inputs detected by earlier layers; end-to-end latency benefits depend on the input distribution. We excluded a separate English-centric injection classifier as an upfront layer: preliminary tests found 32% FPR on benign Korean financial requests, including formal polite phrasing. Injection criteria were therefore integrated into Layer 3.

Figure 1 shows the cost-ordered detection path and its connection to the update loop. Any detection exits immediately; only inputs surviving all three layers are forwarded.

The Layer 3 policy is stored as JSON objects containing a stable identifier, condition text, originating round, triggering failure type, and activation state. Detection rules and safe exceptions are stored separately. Rendering assigns list numbers only at inference time, preventing malformed prompts after repeated insertion.

2.3 Regression-Aware Policy Maintenance

Figure 1 also summarizes the batch maintenance loop. Given current policy p_t , the red team evaluates every labeled item and collects false nega-

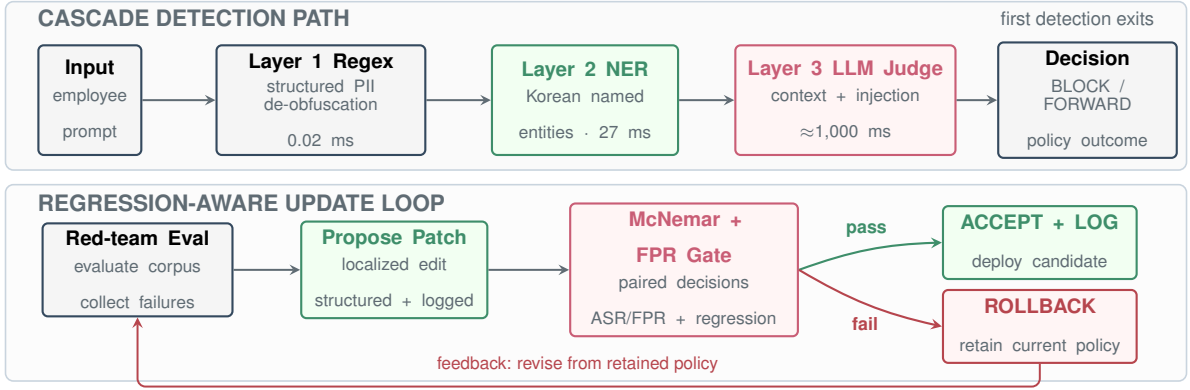


Figure 1: Three-layer input guardrail and regression-aware batch update loop. The framework permits candidate deployment only after paired ASR/FPR and regression checks; otherwise the current policy is retained through rollback.

tives. An LLM acts as the patch generator, proposing a localized edit Δ_t from the failures. The patch gate then evaluates the candidate $p_t + \Delta_t$ and current policy on identical examples using ASR, FPR, paired McNemar tests, and the cumulative regression corpus. A candidate is accepted only if it provides statistically supported ASR improvement, satisfies the FPR constraint, and passes cumulative regression checks; otherwise the system retains p_t through rollback. Two consecutive rejected local patches trigger a structural-refactoring proposal but do not bypass the gate. We used a prespecified per-comparison significance level of $\alpha = 0.05$. The FPR gate required $\text{FPR}_{\text{candidate}} \leq \text{FPR}_{\text{current}} + 0.02$, allowing at most a 2-percentage-point increase relative to the current policy. Both criteria were fixed as part of the update algorithm and applied uniformly across all rounds and evaluation sets; in particular, the FPR tolerance was not derived post hoc from the observed 1/200 baseline false-positive frequency. Because five candidates were tested on the same evaluation corpus across five auto-tuning rounds, the reported p -values are unadjusted per-comparison values; however, all were $p \geq 0.480$, so applying Holm correction would leave every rejection decision unchanged. Every proposal, metric, decision, and rollback is logged, and deployment can require human approval. We evaluate only this batch mode; a separate real-time remediation path is designed but not empirically validated and is excluded from all performance claims.

Round	ASR [95% CI]	FPR [95% CI]	Main change
1	8.9 [7.29, 10.83]	0.5 [0.09, 2.78]	Initial three-layer system
2	1.6 [0.99, 2.58]	0.5 [0.09, 2.78]	Context criteria + examples
3	3.8 [2.78, 5.17]	1.0 [0.27, 3.57]	Additive obfuscation patch
4	0.3 [0.10, 0.87]	0.5 [0.09, 2.78]	Independent structured rules

Table 1: In-distribution ASR/FPR across four manual iterations on the same original 1,200 tuning examples, distinct from the revised held-out split. Brackets give Wilson 95% CIs for ASR ($n = 1,000$) and FPR ($n = 200$).

3 Experiments

3.1 Red/Blue-Team Iterations

Table 1 and Figure 2 report four manual iterations on the same 1,200 examples. Round 2 adds explicit contextual-leakage criteria. In Round 3, naively appending anti-obfuscation rules makes contextual ASR regress from 1.0% to 17.0%. The paired change is significant (11 improved versus 33 worsened decisions; continuity-corrected McNemar $p = 0.002$) and is consistent with *instruction dilution*: core privacy conditions became less salient within the expanded policy. Round 4 restructures independently sufficient conditions as a checklist with standardized domain examples, restoring overall ASR to 0.3% (35 improved, 0 worsened from Round 3; $p < 0.001$).

An ablation cautions against attributing the recovery to list formatting alone. When example content differs across variants, narrative, bullet, and checklist prompts obtain ASRs of 0.7%, 4.4%, and 4.1%, respectively. After adding the same examples to every variant, the gap narrows to 0.7%,

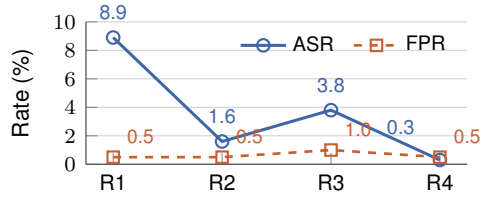


Figure 2: ASR and FPR across four manual iterations. Round 3 regresses; Round 4 restores both metrics.

1.2%, and 1.0%, all at 0.5% FPR. This is consistent with evidence that LLM behavior depends on information placement and context structure (Liu et al., 2024); here, however, the results more strongly support the contribution of concrete examples than list formatting alone. Appendix A (Table 4) reports multi-turn ASR falling from 2.5% to 0.0%; its evaluation scope follows Section 2.1.

3.2 Layers, Models, and Baselines

Layer ablation gives ASR 77.4% for Regex, 64.8% for Regex+NER, and 0.2% for the full cascade ($p < 0.001$ for adding Layer 3). Interpreted sequentially on this attack set, Layer 1 blocks 22.6% of attacks, Layer 2 blocks an additional 12.6%, and 64.8% reach Layer 3. These proportions are distribution-specific and do not imply comparable latency savings for benign traffic, which is more likely to reach the LLM judge. Applying the same policy to local Llama-3-8B blocks all attacks but falsely blocks 80.0% of normal requests, showing that the policy does not transfer safely without model-specific validation.

To check whether the Round 4 checklist overfits to attack types seen during tuning, we evaluate it on two held-out categories within the same taxonomy--ROLE_ABUSE and PRETEXTUAL_BUSINESS (100 examples each)--that were absent from the tuning and main evaluation rounds. Both obtain ASR 0.0%. These held-out categories are not an independent external distribution because they follow the same taxonomy and data-generation protocol.

Table 2 compares Presidio (Data Privacy Stack, 2026), LLM Guard (Protect AI, 2026), and our system; direct leakage is the fairest shared PII task. Presidio’s multilingual mode worsens direct-leakage ASR from 20.0% to 47.5% and FPR from 8.5% to 17.5%. Our inspection suggests that Korean mode disables some default English-registered regex recognizers that multilingual NER does not replace. LLM Guard lowers overall ASR to 9.3% but raises FPR to 36.5%. All direct-

System	Overall ASR [95% CI]	FPR [95% CI]	Direct ASR
Presidio (default)	63.3 [60.3, 66.2]	8.5 [5.4, 13.2]	20.0
Presidio (multi.)	58.8 [55.7, 61.8]	17.5 [12.9, 23.4]	47.5
LLM Guard	9.3 [7.7, 11.3]	36.5 [30.1, 43.4]	6.0
Ours	0.2 [0.1, 0.7]	0.5 [0.1, 2.8]	0.0

Table 2: Results of baseline systems and the unchanged Round 4 policy evaluated on the revised controlled set (%). Brackets give Wilson 95% CIs for overall ASR ($n = 1,000$) and FPR ($n = 200$); see text for the 0.2% versus 0.3% difference.

leakage differences against our system are significant ($p < 0.010$), supporting the value of domain adaptation rather than claiming universal superiority. The 0.3% original-set ASR (Table 1) versus 0.2% revised-set ASR (Table 2) reflects evaluation-version differences, not further tuning or model changes.

3.3 Held-Out Evaluation and Gated Patch Decisions

Table 3 summarizes the evaluation-set versions and their uses. The original and revised controlled sets each contain 1,200 examples but differ in construction; only the revised set is partitioned into tuning and held-out splits. Following the validation procedure in Section 2.1, the revision (i) replaced self-generated PII and obfuscations in DIRECT and OBFUSCATION with Faker/checksum-validated and PyRIT-generated counterparts, (ii) restructured ROLEPLAY into four named techniques, and (iii) replaced the GPT-4.1-mini co-annotator used for inter-annotator agreement (IAA) assessment with an independent Llama-3-8B evaluator to avoid circular self-evaluation. The attack taxonomy and set size were unchanged. Rounds 1--4 preceded this revision, which addressed data-validity issues rather than observed R1--R4 performance. Hard negatives augment the benign portion separately.

We stratify the revised evaluation set into 840 tuning and 360 held-out examples, grouping identical text before splitting. Starting from a policy not previously exposed to the revised data, one tuning round reaches 0/700 attack successes and 0/140 false positives on the tuning split. This separately tuned policy is distinct from the Round 4 policy evaluated on the original and revised controlled sets in Tables 1 and 2. On held-out data, ASR is 0/300 (95% Wilson CI [0, 1.26]) and FPR is 1/60

Setting	Attacks	Benign	Primary use
Original controlled set	1,000	200	Manual R1--R4
Revised controlled set	1,000	200	Final/baselines
Tuning split	700	140	Policy updating
Held-out split	300	60	Held-out test
Hard-negative add-on	---	150	Boundary cases
Augmented evaluation	1,000	350	Boundary FPR

Table 3: Evaluation sets and protocols. The tuning and held-out rows partition the revised controlled set; the augmented evaluation adds 150 hard negatives to its 200 benign examples. The original-to-revised revision addressed data-validity issues identified through the validation procedure in Section 2.1 rather than Rounds 1-4 performance; the attack taxonomy and total set size were unchanged.

(1.67%, [0.29, 8.86]). The wide benign interval precludes a strong FPR claim.

In a separate automatic-update experiment starting from the Round 4 policy on the full revised evaluation set, the candidate in the fifth auto-tuning round marginally reduced empirical ASR from 0.2% to 0.1% while maintaining FPR, correcting only one additional item. A simple empirical rule accepting any ASR decrease without an FPR increase would have accepted this candidate; the statistical gate instead rejected it because the paired improvement was not significant (McNemar $p = 1.0$). Across five auto-tuning rounds, all five generated candidates were rejected for lack of statistical significance ($p \geq 0.480$); structural escalation after two consecutive local-patch failures did not bypass the gate. Retaining the current policy via rollback therefore prevents empirically favorable but statistically unsupported changes from entering the policy.

3.4 Hard Negatives and Error Analysis

For boundary diagnosis, we evaluate the rejected fifth-round candidate policy (empirical ASR 0.2% to 0.1%), which was not deployed. Introducing 150 benign boundary cases (hypothetical PII, masked PII, and aggregate statistics; 50 each) maintained ASR at 0.1% but produced a pre-mitigation overall FPR of 28.9% across 350 benign examples, compared with 0.5% on routine requests alone. At this baseline, context-blind Regex blocked all hypothetical PII formats (100% FPR), and the LLM judge over-blocked all masked PII items (100% FPR). A targeted safe exception reduced the masked-PII subset FPR from 100% to 40%; the paired ASR change was not statistically significant ($p = 0.480$). A broader exception increased

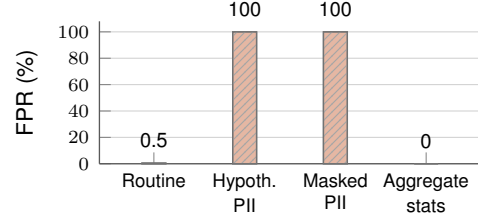


Figure 3: Pre-mitigation hard-negative diagnosis: Regex blocks hypothetical formats, while the LLM judge over-blocks masked PII.

ASR to 0.9% ($p = 0.013$) and was rolled back. The archived evaluation does not include a recomputed aggregate FPR over all 350 benign examples after the targeted exception. Thus, 28.9% is the pre-mitigation aggregate FPR and 40% is the post-mitigation masked-PII subset FPR, rather than a final overall FPR. These results expose the need for context-aware layer bypassing and complete post-mitigation reevaluation under the regression gate.

To test judge-targeted injection, we append eight templates (prior-instruction overrides and fake system notices) to 100 baseline DETECTED cases. None of the 100 tested cases flipped to SAFE (0/100; 95% Wilson CI [0, 3.7%]). This result is limited to the tested templates and sample size; stronger judge-targeted attacks remain unverified.

4 Conclusion

We presented regression-aware policy maintenance for Korean financial LLM input guardrails. An LLM generates candidate patches, while McNemar tests, an FPR constraint, and cumulative regression checks gate their acceptance; rejecting all five unsupported candidates demonstrates prevention of unvalidated policy changes rather than automatic improvement. A cost-ordered Regex--NER--LLM cascade provides the domain-specific guardrail, while hard negatives reveal substantial boundary failures hidden by routine benign requests. Although our evaluation is stateless, preliminary concatenated-history results in Appendix A (Table 4) reach 0.0% multi-turn ASR; fully stateful session tracking, contextual bypass, and deployment-specific evaluation remain future work.

Limitations

The controlled set is not live traffic and cannot estimate production prevalence or FPR; generalization requires privacy-preserving institution-specific

evaluation. The two held-out attack categories do not constitute an independent external distribution because they were constructed under the same taxonomy and data-generation protocol. The controlled taxonomy does not exhaust the input-gateway attack surface. Untested cases include long-tail injection strategies outside the five categories, broader encoding-based obfuscations beyond the four implemented transformations (e.g., Base64, ROT13, and URL/Unicode encoding), and fully stateful multi-turn elicitation. The hard-negative set likewise covers only three selected benign boundary types rather than the full diversity of legitimate financial prompts. Hard negatives diagnose boundaries, not frequency. The 28.9% aggregate FPR was measured before mitigation under the rejected fifth-round candidate, not the retained Round 4 policy. Aggregate FPR across all 350 benign examples was not recomputed after the targeted exception. The archived evaluation therefore does not provide a final aggregate FPR over all 350 benign examples for either the mitigated candidate or the retained Round 4 policy. The present evaluation demonstrates safe rejection of unsupported candidate patches but does not establish a high acceptance rate or consistent improvement rate for automatically generated patches. The stateless gateway concatenates history; a history-aware path reduced ASR from 32.0% to 15.5% but regressed five decisions. The reported results are limited to concatenated conversational histories; fully stateful session-level validation remains future work. Comparisons are not compute-matched, Layer 3 adds roughly one second, and logs require privacy governance. The Layer 3 policy is model-specific: applying it to Llama-3-8B falsely blocked 80.0% of normal requests, so transfer to another judge model requires model-specific revalidation.

Ethics Statement

Regex matches need contextual validation; batch updates retain rollback and approval, while real-time remediation remains unvalidated.

References

- Amazon Web Services. 2026. [Detect and filter harmful content by using amazon bedrock guardrails](#). Amazon Bedrock User Guide. Accessed 2026-08-05.
- CB Financial Services, Inc. 2026. [Current report on form 8-k, item 1.05: Material cybersecurity incidents](#). U.S. Securities and Exchange Commission

filing. Item 1.05, Material Cybersecurity Incidents; date of report: May 7, 2026; accessed 2026-08-05.

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*.

Data Privacy Stack. 2026. [Presidio: Data protection and de-identification sdk](#). Software documentation. Accessed 2026-08-05.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157--173.

Gary D. Lopez Munoz, Amanda J. Minnich, Roman Lutz, Richard Lundeen, Raja Sekhar Rao Dheekonda, Nina Chikanov, Bolor-Erdene Jagdagdorj, Martin Pouliot, Shiven Chawla, Whitney Maxwell, Blake Bullwinkel, Katherine Pratt, Joris de Gruyter, Charlotte Siska, Pete Bryan, Tori Westerhoff, Chang Kawaguchi, Christian Seifert, Ram Shankar Siva Kumar, and Yonatan Zunger. 2024. [PyRIT: A framework for security risk identification and red teaming in generative AI system](#). *arXiv preprint arXiv:2410.02828*.

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 (Volume 1: Long Papers)*, pages 8104--8139, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2026. [GPT-4.1 mini Model](#). OpenAI API documentation. Accessed 2026-08-05.

OWASP GenAI Security Project. 2025. [2025 top 10 risk & mitigations for LLMs and GenAI apps](#). Online. Accessed 2026-08-05.

Jangwon Park. 2020. [KoELECTRA: Pretrained ELECTRA model for korean](#). GitHub repository. Accessed 2026-08-05.

Promptfoo. 2026. [LLM red teaming](#). Software documentation. Accessed 2026-08-05.

Protect AI. 2026. [LLM Guard: The security toolkit for LLM interactions](#). GitHub repository. Archived repository; accessed 2026-08-05.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chengguang Zhu, and Michael Zeng. 2023. [Automatic prompt optimization with “gradient descent” and beam search](#). In *Proceedings of the 2023 Conference*

on *Empirical Methods in Natural Language Processing*, pages 7957--7968, Singapore. Association for Computational Linguistics.

Traian Rebedea, Razvan Dinu, Makesh 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.

A Additional Results

R.	Direct	Context	Obfusc.	Roleplay	Multi-turn
1	0.0	36.5	5.5	0.0	2.5
2	0.0	1.0	6.5	0.0	0.5
3	0.0	17.0	1.0	0.0	1.0
4	0.0	0.5	1.0	0.0	0.0

Table 4: Attack success rate (%) by category; 200 examples per cell.

B Reproducibility Checklist

Table 5 records the execution settings used for the reported experiments. We distinguish the API evaluation period from the pinned model version: experiments were run in July 2026, while OpenAI identifies gpt-4.1-mini-2025-04-14 as the fixed GPT-4.1-mini snapshot ([OpenAI, 2026](#)). The exact execution day was not retained in the archived logs.

Setting	Value
Layer 3 model alias	gpt-4.1-mini
Pinned API snapshot	gpt-4.1-mini-2025-04-14
API evaluation period	July 2026 (exact day not preserved)
Temperature	0
Top- p	1.0
Maximum output tokens	512 (max_tokens)
NER checkpoint	monologg/koelectra-base-v3-discriminator
Public artifacts	Code and synthetic data repository
Global random seed	42
Held-out split seed	42
Python	3.11

Table 5: Execution settings for the reported experiments.