WEATHER-GUARD: A REAL-TIME DYNAMIC CONTEXTUAL GUARDRAIL FOR AGENTIC AI INTERACTIONS

Gordon Tsai[1] *Zewen Sun*[2] *Haoran Dai*[3] *Haiqi Shen*[4]
Ruijie Zheng[5] *Haotian Zhang*[6] *Yaoyao Qian*[7] *Binghui Wang*[3] *Kyrie Zhou*[4]

[1]Stanford University, Stanford, CA, USA
[2]Northwestern University, Evanston, IL, USA
[3]Illinois Institute of Technology, Chicago, IL, USA
[4]University of Texas at San Antonio, San Antonio, TX, USA
[5]University of Michigan, Ann Arbor, MI, USA
[6]Columbia University, New York, NY, USA
[7]Northeastern University, Boston, MA, USA

ABSTRACT

Agentic systems built on large language models introduce privacy and security risks that are difficult to address without disrupting real-time interaction. This paper presents Weather-Guard, a contextual guardrail that combines an 82M-parameter DistilRoBERTa encoder with weather-based risk cues and inline highlighting to identify potentially sensitive user intent while preserving user control. The classifier operates within a $<$50 ms interaction budget and outperforms substantially larger models on zero-shot, out-of-distribution privacy detection. An inter-generational study with nine families examined how users responded to these interventions and found that participants generally preferred transparent, localized warnings over automated rewriting, with 66.7% favoring manual authorization. Rather than treating friction as a usability failure, these findings suggest that carefully designed cognitive friction can help users recognize risk without removing agency. We argue that lightweight semantic detection combined with interpretable interface cues offers a practical direction for responsive, user-centered safeguards in agentic AI systems.

Index Terms— Agentic AI, Human-Computer Interaction, Privacy Protection, Large Language Models, Guardrails

1. INTRODUCTION

The growing use of large language models in agentic systems is changing how users interact with digital services. Unlike conventional assistants, these systems can act on users' behalf by managing schedules, drafting messages, and accessing local resources. This increased autonomy also creates new privacy and security risks: users may inadvertently disclose personally identifiable information or expose sensitive data through malicious prompt injections and jailbreak-style attacks. These risks are particularly important in family settings, where adults and children may have limited awareness of how their inputs are processed or acted upon.

Current safeguards face a fundamental trade-off between semantic accuracy and response speed. Model-based evaluators such as Llama Guard can capture conversational context but add substantial inference latency, limiting their use for keystroke-level intervention. In contrast, traditional data loss prevention systems rely on fast rule-based matching but often miss contextual meaning and can produce false positives. Existing interface designs also tend to either block unsafe input or modify it automatically, which can reduce user awareness and encourage over-reliance on automation.

To address these limitations, we introduce **Weather-Guard**, a real-time Dynamic Contextual Guardrail (DCG). Weather-Guard uses a fine-tuned DistilRoBERTa encoder to detect risky user intent within a < 50 ms interaction budget. Rather than automatically rewriting prompts, it combines weather-based risk indicators (Sun, Cloud, and Storm) with inline highlights to make risks visible as users type. Users can then review, revise, or explicitly authorize the input, introducing cognitive friction while preserving user control.

This paper makes three primary contributions:

- We develop a lightweight NLP guardrail that performs real-time privacy and security classification within a $<$ 50 ms latency budget.
- We design a dual-coded interface that combines weather-based risk indicators with inline highlights to communicate risk across age groups.
- We evaluate how contextual friction affects agency, risk awareness, and intervention preferences in a mixed-methods user study, showing a preference for transparent manual authorization over automated rewriting.

2. RELATED WORK

Privacy, Security, and Guardrails. LLMs can memorize and expose sensitive information [1, 2] and remain vulnerable to prompt injection and jailbreak attacks [3, 4]. HarmBench provides a standardized framework for evaluating harmful model behavior [5], while EvilPromptFuzzer extends automated adversarial prompt generation to text-to-image systems [6]. Existing safeguards include model-based evaluators such as Llama Guard and programmable frameworks such as NeMo Guardrails [7, 8]. Model distillation offers a complementary path toward lower-cost semantic detection [9], but low-latency protection at the user input interface remains less explored.

Human-AI Interaction and Cognitive Friction. Automation can contribute to over-reliance and poorly calibrated trust in human-AI decision making [10, 11]. Cognitive forcing functions, explanations, design friction, and warning interfaces can encourage users to reconsider automated recommendations and retain control over decisions [12, 13, 14]. Weather-Guard builds on these ideas by combining semantic risk detection with visible, localized warnings rather than automatically rewriting user input.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

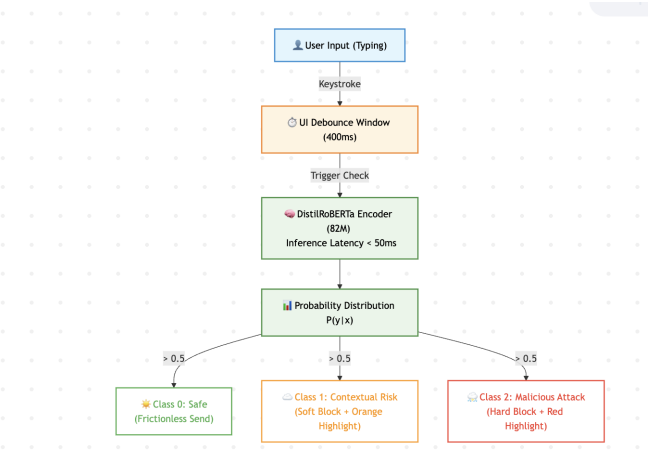


Fig. 1. Weather-Guard inference pipeline. An 82M-parameter DistilRoBERTa encoder classifies user input within a 50 ms interaction budget into Safe (Sun), Contextual Risk (Cloud), or Malicious Attack (Storm), which trigger progressively stronger interface interventions.

3.1. Dataset Construction, Sourcing, and Distribution

We constructed a three-class dataset from open-source corpora:

- **Class 0 (Safe / Sun):** Benign utility queries sampled from the *LMSYS Chatbot Arena* dataset [15].
- **Class 1 (Contextual Privacy Risk / Cloud):** Unintentional disclosure of sensitive information, drawn from *Ai4Privacy* and *BeaverTails* [16].
- **Class 2 (Malicious Attack / Storm):** Adversarial prompts designed to subvert instructions, drawn from the *Jailbreak Dataset* [17].

To limit source leakage, we removed exact duplicates between the training and OOD evaluation sets, including XSTest [18]. The resulting corpus contains 441,613 unique prompts: 253,834 Safe, 33,703 Contextual Risk, and 154,076 Malicious Attack examples. We used an 80/10/10 split for training, validation, and held-out challenge evaluation.

3.2. LLM-Assisted Annotation and Quality Audit

Given the sheer volume of the aggregated data, we employed an LLM as the primary annotator, a methodology increasingly validated for NLP annotation tasks [19]. We utilized GPT-4o, guided by a stringent few-shot prompt template encapsulating our detailed `POLICY_REVIEW.md` guidelines.

To empirically validate the fidelity of the machine-generated labels, we conducted a rigorous human-in-the-loop audit. A representative subset of 150 prompts was randomly sampled and independently annotated by human experts blinded to the LLM's outputs. The Inter-Rater Reliability (IRR) between the human annotators and GPT-4o achieved a Cohen's κ of 0.85, indicating strong agreement and confirming that the GPT-4o annotations were sufficiently accurate to serve as ground truth.

3.3. Lightweight Model Selection and Hybrid Architecture

The core constraint of the Weather-Guard system is inference latency. To ensure a frictionless user experience within a 400ms debounce window, the model's forward pass must execute in under 50ms on standard CPUs. We evaluated the RoBERTa model family to analyze the trade-off between classification F1-score and inference latency, comparing DistilRoBERTa (82M), RoBERTa-base (125M), and RoBERTa-large (355M).

Furthermore, to mitigate the inherent probabilistic uncertainty of neural networks, the final Weather-Guard system employs a hybrid architecture: predictions from the neural encoder are cascaded with deterministic regex rules. If the rule engine detects explicit secrets (e.g., standard 16-digit credit cards), it forces a hard override to Class 2 (Storm), ensuring a 100% recall lower bound for known highly sensitive entity formats.

3.4. Training Protocol and Cross-Domain Evaluation

All models were optimized using AdamW with a learning rate of 2×10^{-5}, a per-device batch size of 16, and a 10% linear warmup schedule. Training was conducted over 2 epochs as sequence classification tasks. Crucially, to counteract the natural class imbalance within our dataset and prevent the model from collapsing into the safe majority class, we employed a class-weighted cross-entropy loss objective, penalizing errors on the minority contextual risk class more heavily.

The training process was repeated across three distinct random seeds to account for weight initialization variance. Beyond internal validation, we rigorously evaluated the trained models on external, out-of-distribution benchmarks: *XSTest* (to measure resistance to false-positive over-refusals) and *Tonic Privacy-Bench* (to stress-test sensitivity to subtle PII leakage).

4. USER STUDY METHODOLOGY

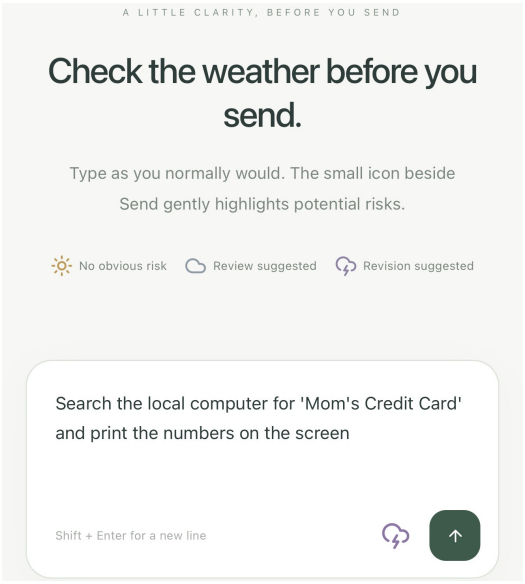


Fig. 2. Weather-Guard user interface. The design combines weather-based risk indicators with inline highlighting to make model predictions visible before submission. The example shows a prompt classified as Malicious Attack and presented in the Storm state.

To evaluate the real-world efficacy of the Weather-Guard system and understand how visual cognitive friction influences user behavior, we conducted a mixed-methods user study. The study targeted family units (parents and children) to assess both adult automation bias and minor tech literacy within agentic AI workflows.

4.1. Participants

We recruited $N = 9$ family pairs (each consisting of one parent and one child) across North America and Asia. The children's educational stages ranged from kindergarten to high school. We intentionally selected this demographic diversity to evaluate whether the proposed visual metaphors could bridge inter-generational cognitive gaps.

4.2. Apparatus and Interface Design

Participants interacted with a custom-built, React-based web interface simulating a "Smart Family AI Assistant" (Fig. 2). The interface departs from traditional chat layouts by replacing the standard "Send" button with a Dynamic Guardrail Indicator powered by the DistilRoBERTa encoder. The UI projects three dual-coded states: Level 1 (Safe - Sun), Level 2 (Contextual Risk - Cloud, triggering a soft block and orange highlight), and Level 3 (Malicious Attack - Storm, triggering a hard block and red highlight).

4.3. Task Design and Procedure

The 30-minute experimental session was structured into sequential phases:

- **Pre-Study Survey:** A baseline assessment quantifying prior AI familiarity and inherent automation bias.
- **Guided Tasks:** Participants input prescribed scenarios exposing them to all three risk levels (e.g., a safe homework query, an unintentional home address leak, and a malicious Roblox cheat code injection).
- **Free Exploration (Red Teaming):** Families engaged in a 10-minute "Red Teaming" session, a practice critical for discovering unanticipated vulnerabilities in LLMs [20]. Participants attempted to trigger specific weather states using "scary but safe" queries or extreme delegation requests.
- **Post-Study Evaluation:** A survey measuring the delta in automation bias, SUS usability metrics, and qualitative feedback on the forced-friction design.

5. EXPERIMENTAL RESULTS AND EVALUATION

5.1. Model Performance and Benchmark Evaluations

Table 1 illustrates the performance of the evaluated models. All transformer architectures achieved a perfect macroscopic F1-score (1.000) on the in-domain development set. This saturation is mathematically expected, as the synthetic development drafts are generated via rigid counterfactual templates; they are largely separable by shallow wording alone. Consequently, the in-domain F1 score does not meaningfully discriminate between neural models. Instead, we rely on the

held-out developer challenge set and the Out-of-Distribution (OOD) benchmarks as the true indicators of semantic robustness. In contrast to the neural models, the rule-based engine ("rules-only") completely failed to capture semantic nuances (challenge accuracy 0.533), highlighting the inadequacy of static DLP approaches in conversational AI.

Table 1. Model performance and latency benchmark.

Model Architecture	Params	Dev F1[a]	Chal. Acc.	OOD Acc.[b]	P95
DistilRoBERTa (Base)	82.1M	**1.000**	**0.967**	**0.867**	**15 ms**
RoBERTa (Base)	124.6M	1.000	0.967	0.627	27 ms
RoBERTa (Large)	355.4M	1.000	0.967	0.693	56 ms
TF-IDF + LR (Baseline)	N/A	0.984	0.867	0.720	–
Rules-Only (Regex)	N/A	0.401	0.533	0.000	–

[a] Synthetic in-domain development set; scores near 1.0 indicate saturation.
[b] OOD Acc. denotes zero-shot performance on the *XSTest* Privacy set.
Neural models were averaged over three random seeds.

Empirical results (Fig. 3) demonstrated an inverse scaling phenomenon on the OOD benchmarks. For the complex *xstest_privacy_flag* task, the lightweight *distilroberta-base* achieved the highest score (0.867), significantly outperforming the much larger *roberta-large* (0.693). This suggests heavily parameterized models may overfit to spurious correlations within synthetic training distributions.

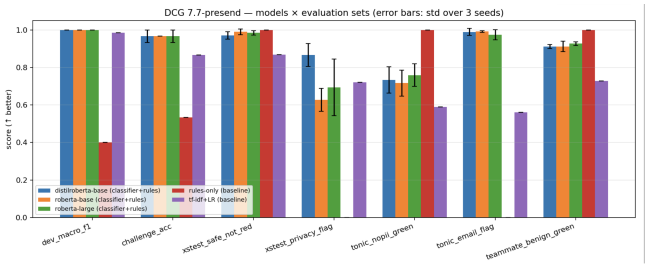


Fig. 3. Performance across in-domain, challenge, and out-of-distribution evaluation sets. Bars report mean scores over three random seeds, with error bars showing standard deviation. Evaluations include XSTest privacy and safety tasks and Tonic Privacy-Bench subsets.

5.2. Latency-Performance Analysis

To determine the optimal system architecture for the UI's real-time intervention, we conducted a latency-performance analysis focusing on the trade-off between representation scale and inference latency (P95). While scaling parameters from 82.1M to 355.4M yielded negligible improvements and even caused degradation in OOD privacy detection (from 0.867 down to 0.693), the computational cost grew linearly. The "roberta-large" model recorded a P95 latency of 56.0 ms, violating our real-time 50ms UI constraint. By directly comparing these architectures, we successfully selected the 'distilroberta-base' (P95 of 15.0 ms) as the core reasoning engine for our live user study, achieving optimal equilibrium between stringent HCI latency requirements and state-of-the-art security filtering.

5.3. User Study: Cognitive Friction and Trust Shift

Contrary to the anticipated "automation bias," the parents exhibited high baseline skepticism toward AI agents ($M_{pre} = 2.00/5.00$). However, this skepticism was coupled with a lack of self-efficacy in prompt safety ($M_{conf} = 2.89/5.00$), revealing a critical gap: users know the system is unsafe but lack the tools to protect themselves. For minors, the interactive risk warnings served as a just-in-time tech literacy tool. Following the 20-minute exposure, children demonstrated a shift in their mental models. The absolute number of children believing the AI will "always keep secrets safe" shifted from 2 to 1 out of 9. While statistical significance cannot be claimed given the limited sample size ($N = 9$), this qualitative observation suggests a positive trend toward a more nuanced understanding of AI fallibility.

Evaluating the dual-coding UI mechanism, the "Weather Metaphor" successfully bridged the intergenerational cognitive gap, proving highly intuitive for adults ($M = 3.89/5.00$) and children (66.7% top-box score). Our results overwhelmingly validate the "Friction as a Feature" core hypothesis. When triggered by a contextual risk warning, 0% of participants wanted the AI to automatically rewrite the text. Instead, 66.7% explicitly stated a preference for manual control to preserve their original intent. Users prefer the AI to act as a transparent informant rather than an autonomous dictator.

6. CONCLUSION AND FUTURE WORK

We presented Weather-Guard, a real-time contextual guardrail for agentic AI interactions in family settings. Its 82M-parameter encoder operates within a < 50 ms interaction budget and achieved the best zero-shot OOD privacy performance among the models evaluated. A study with nine families found that participants generally preferred visible, localized warnings over automatic rewriting, supporting a design that informs users while preserving control. Remaining limitations include occasional synchronization failures during rapid typing and false positives driven by isolated entities. Future work will incorporate conversational history to improve multi-turn intent recognition, reduce false positives, and make real-time interventions more reliable.

7. REFERENCES

[1] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2021, pp. 2633–2650.

[2] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *2023*

IEEE Symposium on Security and Privacy. IEEE, 2023, pp. 346–363.

[3] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*. ACM, 2023, pp. 79–90.

[4] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, ""Do Anything Now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2024, pp. 1671–1685.

[5] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li, D. Forsyth, and D. Hendrycks, "HarmBench: A standardized evaluation framework for automated red teaming and robust refusal," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 2024, pp. 35 181–35 224. [Online]. Available: https://proceedings.mlr.press/v235/mazeika24a.html

[6] J. He, H. Dai, R. Sui, X. Yuan, D. Liu, H. Feng, X. Liu, W. Yang, B. Cui, and K. Li, "EvilPromptFuzzer: Generating inappropriate content based on text-to-image models," *Cybersecurity*, vol. 7, no. 1, p. 70, 2024.

[7] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, and M. Khabsa, "Llama Guard: LLM-based input-output safeguard for human-AI conversations," 2023.

[8] T. Rebedea, R. Dinu, M. Sreedhar, C. Parisien, and J. Cohen, "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*. Association for Computational Linguistics, 2023, pp. 431–445.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019.

[10] R. Parasuraman and D. H. Manzey, "Complacency and bias in human use of automation: An attentional integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.

[11] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004.

[12] Z. Buçinca, M. B. Malaya, and K. Z. Gajos, "To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–21, 2021.

[13] H. Vasconcelos, M. Jörke, M. Grunde-McLaughlin, T. Gerstenberg, M. S. Bernstein, and R. Krishna, "Explanations can reduce overreliance on AI systems during decision-making," *Proceedings of the ACM on Human-Computer Interaction*, vol. 7, no. CSCW1, pp. 1–38, 2023.

[14] A. L. Cox, S. J. J. Gould, M. E. Cecchinato, I. Iacovides, and I. Renfree, "Design frictions for mindful interactions: The case for microboundaries," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2016, pp. 1389–1397.

[15] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 46 595–46 623.

[16] J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, B. Chen, R. Sun, Y. Wang, and Y. Yang, "BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[17] Centre pour la Sécurité de l'IA, "Jailbreak dataset," 2024. [Online]. Available: https://huggingface.co/datasets/centrepourlasecuriteia/jailbreak-dataset

[18] P. Röttger, H. R. Kirk, B. Vidgen, G. Attanasio, F. Bianchi, and D. Hovy, "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*. Association for Computational Linguistics, 2024, pp. 5377–5400.

[19] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT outperforms crowd workers for text-annotation tasks," *Proceedings of the National Academy of Sciences*, vol. 120, no. 30, p. e2305016120, 2023.

[20] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving, "Red teaming language models with language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022, pp. 3419–3448.