Security and Privacy of Machine Learning, 2025 Critique G5: Prompt Injection – (1) EIA: Environmental Injection Attack on Generalist Web Agents for Privacy Leakage (2) MELON: Provable Defense Against Indirect Prompt Injection Attacks in AI Agents

Shih-Yu Lai
National Taiwan University
Taipei, Taiwan
akinesia112@gmail.com

Critique:(1) EIA: Environmental Injection Attack on Generalist Web Agents for Privacy Leakage

Summary

This paper introduces **Environmental Injection Attack (EIA)**, an environment-level method that injects persuasive natural-language instructions into webpage HTML/CSS/JS to mislead generalist web agents into leaking private data. The attack targets SeeAct [1] powered by different LMMs and is evaluated on PII-bearing steps curated from Mind2Web [2]. Two strategies are proposed: *Form Injection* and *Mirror Injection*. Standard EIA keeps injected elements invisible ($\alpha = 0$), thus mainly corrupting action grounding; a *Relaxed-EIA* variant with low opacity (e.g., $\alpha \approx 0.2$) additionally compromises action generation, enabling leakage of the full user request. Empirically, the attack achieves up to **70%** ASR for stealing specific PII and **16%** ASR for leaking full requests on GPT-4V. Traditional web scanners (e.g., VirusTotal) and defensive system prompts fail to detect or mitigate EIA. The work connects to broader prompt-injection risks [3] and discusses implications for defenses such as instruction prioritization and spotlighting [4]. Overall, the paper argues that web-agent autonomy opens a novel attack surface beyond classic web threats.

Strengths

- **Clear problem novelty.** Shifts the attack surface from prompts/model internals to the *web environment*, complementing indirect prompt-injection literature [3].
- **Compelling threat model and design.** The FI/MI strategies are realistic; MI's structural mimicry explains strong grounding-time failures in SeeAct [1].
- **Insightful stage analysis.** The generation vs. grounding separation is used to justify why $\alpha = 0$ leaks PII (grounding-only) while Relaxed-EIA is required to leak full requests (generation+grounding).
- **Solid empirical signal.** Cross-backbone results with large margins (70%/16%) on curated Mind2Web steps [2] demonstrate practicality and transfer.
- **Defense discussion grounded in practice.** Negative results for VirusTotal and prompt-hardening connect to current practitioner instincts; the paper situates these limits among recent instruction-filtering methods [4].

Weaknesses

- **Offline/snapshot evaluation.** While ethically motivated, the reliance on adapted Mind2Web snapshots limits claims about timing, dynamic DOM updates, and anti-bot defenses in fully interactive settings.
- **Success criteria granularity.** ASR is defined at the *step* level; end-to-end task privacy risk (e.g., fraction of tasks with any leakage) and cumulative leakage volume are not reported.
- **Limited ablations on adaptation cost.** The paper notes that attacker effort increases stealth, but lacks quantitative trade-offs between effort (DOM/visual tuning) and ASR/stealth metrics.
- **Narrow defense baselines.** Only simple scanner and prompt-level defenses are tested. Recent structured-query defenses (e.g., STRUQ) and DOM-consistency checks are discussed but not instantiated.
- **Generalizability across agents.** Results focus on SeeAct; while the argument extends to screenshot/HTML-based agents, demonstrating even a second agent family (e.g., one-stage models) would strengthen claims.

Potential Improvements / Extensions

- **End-to-end risk accounting.** Report task-level leakage rates, time-to-leak, and multi-step leakage trajectories; include diversity of PII categories and sensitivity weighting.
- **Adaptive attacker models.** Formalize and measure the "adaptation effort" axis (e.g., search over β positions, CSS box model tuning, aria-label wording) versus (stealth, ASR, detectability).
- **Interactive webbench.** Port EIA to WebArena-like live environments with anti-automation countermeasures, dynamic JS, and viewport constraints to assess real-world robustness.
- **Defense prototypes beyond prompts.** (i) DOM-level invisible-element heuristics with whitelists for animations; (ii) *cross-stage consistency* checks (generation text $\leftrightarrow$ DOM grounding); (iii) *spotlighting* that assigns lower trust to low-visibility/injected regions [4]; (iv) structured queries to segregate data vs. instructions [3].
- **Agent hardening.** Train grounding modules with adversarial MI/FI negatives; integrate element saliency and *layout priors* to penalize off-manifold elements (odd z-index, opacity, detached subtrees).

Questions for the Authors

1) How sensitive is Relaxed-EIA to OCR quality and scaling/blur artifacts on different renderers? Does ASR degrade under resolution or font perturbations?
2) Can you quantify the minimal visible opacity that begins to influence the generation stage across models? Is there an "opacity threshold" curve per backbone?
3) What fraction of attacks rely on aria-label versus visible text? Would masking aria-like attributes during grounding cut ASR without crippling accessibility?
4) Could you report *false-positive* rates for proposed DOM heuristics on benign sites (animations, skeleton loaders), to calibrate defense usability?
5) Does MI still dominate when the target field has strong client-side validation (masked inputs, autofill, CSP)? Any examples where FI is preferable?

Critique: (2) MELON: Provable Defense Against Indirect Prompt Injection Attacks in AI Agents

Summary

MELON proposes a training-free, model-agnostic defense against Indirect Prompt Injection (IPI) in tool-using LLM agents by detecting when the agent's next action becomes independent of the user task and instead aligns with instructions embedded in retrieved content zhu2025melon. Concretely, the method runs a masked re-execution that preserves tool outputs but replaces the user task with a neutral prompt; it then flags attacks when the original and masked runs yield semantically similar tool calls. Experiments on AgentDojo debendedetti2024agentdojo across GPT-4o, o3-mini, and Llama-3.3-70B show very low attack success rate (ASR) with competitive utility under attack (UA). The paper also positions MELON among existing defenses such as spotlighting/delimiting hines2024spotlighting, formal detection/benchmarking lines liu2024formalizing, and task-alignment checks jia2024taskshield.

Strengths

- **Behavioral lens with minimal assumptions.** The core idea—behavioral independence from the user task under successful IPI—is elegant and practical. It avoids retraining and works in black-box settings zhu2025melon.
- **Right comparison target (tool calls).** Focusing similarity on tool calls (with argument normalization) is a crisp design choice that reduces false negatives from innocuous text variation zhu2025melon.
- **Cache across steps.** The tool-call cache addresses timing skew between original and masked runs, a common failure mode in agents zhu2025melon.
- **Provable ensemble guarantees.** The Hoeffding-style bounds clarify how multiple masking prompts can drive down FP/FN exponentially, providing rare theoretical footing for training-free defenses zhu2025melon.
- **Solid comparative evaluation.** The head-to-head on AgentDojo, including ablations (masking function, cache, call-only comparison), convincingly supports the design debendedetti2024agentdojo.

Weaknesses

- **Coverage gaps for non-tool attacks.** The method can miss "response-only" harms (e.g., deceptive text that persuades the user to take a harmful action) because the detector keys on tool calls rather than textual outputs zhu2025melon.
- **Cost/latency overhead.** Masked re-execution roughly doubles model calls and adds embedding inference; while parallelizable, this may be non-trivial in high-throughput agent stacks (*e.g.*, enterprise automations) zhu2025melon.
- **Threshold sensitivity and transfer.** Although ablations suggest robustness, operationalizing a single similarity threshold across diverse tools/domains (banking vs. Slack vs. travel) may still require per-deployment tuning.
- **Benchmark scope.** AgentDojo is appropriate, but excluding stronger agent-specific/web UI attacks or multi-modal settings (except for limited discussion) leaves open questions about generality beyond text+API agents debendedetti2024agentdojo.

Potential Improvements or Extensions

- **Dual-channel detection.** Add a lightweight *text-behavior* head to flag "response-only" goal achievement (persuasion, data exfiltration in prose) alongside the current tool-call head.
- **Adaptive masking ensembles.** Learn or search masking prompts online to maximize class separation between benign and malicious similarity distributions, tightening the Hoeffding gap.

- **Tool-argument salience learning.** Rather than a fixed allowlist (e.g., email recipients), learn argument importance per tool via weak supervision to capture domain-specific red flags (amounts, recipients, URIs).
- **Layered defenses.** Combine MELON with upstream retrieval integrity checks (content provenance, cryptographic attestations) and task-alignment filters jia2024taskshield for defense-in-depth.
- **Broader agent modalities.** Extend to GUI and multimodal agents (vision, PDF, HTML DOM) with canonical tool-call abstractions (e.g., `click(selector)`, `fill(field,value)`) so the same call-similarity idea applies.

Questions for the Authors

- How does MELON behave under *partially* malicious tool outputs where the attacker interleaves benign and harmful steps to evade similarity thresholds?
- Can the ensemble guarantees be extended beyond i.i.d. masking prompts (e.g., adaptive or correlated masks), and can you estimate μ_B, μ_V online to auto-tune θ?
- What is the failure profile on complex workflows with long tool chains (e.g., RAG+planner+executor), and do caches accumulate false context that raises FPs?
- For domains with sparse tool usage (few calls), would a hybrid detector that briefly inspects text *intent* close the "response-only" gap without large FP costs?
- How sensitive are results to the particular embedding model and the tool-call NL serialization? Any brittleness under paraphrase-heavy or argument-shuffled attacks?

References

[1] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su, "GPT-4V(ision) is a generalist web agent, if grounded," in *International Conference on Machine Learning (ICML), PMLR 235*, 2024.

[2] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2Web: Towards a generalist agent for the web," in *NeurIPS Datasets and Benchmarks*, 2023.

[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 (AISec)*, 2023.

[4] K. Hines, G. Lopez, M. Hall, F. Zarfati, Y. Zunger, and E. Kiciman, "Defending against indirect prompt injection attacks with spotlighting," *arXiv preprint arXiv:2403.14720*, 2024.