Input-side Traps for Governing Undisclosed LLM Reviewing

Xun Yuan [*]
Independent Researcher
Cambridge, MA, USA
yuanxun2000@outlook.com

Abstract

Peer review at major AI venues is increasingly affected by low-quality, LLM-generated reviews, while existing output-side measures lack enforceability or cannot distinguish LLM substitution from benign assistance. This paper elevates what is often an individual, illicit use of input-side injection into a venue-controlled, adversarially structured governance framework. It also introduces a PDF-based trap architecture that embeds adversarially robust signals to preserve a structural cost asymmetry against undisclosed LLM misuse.

1 Introduction

Recent evidence indicates that major AI venues are increasingly facing low-quality, LLM-generated peer reviews [1–3], although such practices are formally prohibited. Existing countermeasures predominantly focus on output-side measures. [1] However, self-reported AI disclosures and reviewer attestations lack any actual enforcement mechanism. By contrast, post-hoc detectors evaluate only the final text and therefore cannot distinguish substantive delegation of the reading and evaluation to an LLM from benign uses such as grammatical polishing. [4, 5] This limitation motivates shifting attention to the input side, leveraging the fact that LLMs are susceptible to prompt injection. In this paper, we develop this observation into a systematic, evolving input-side governance framework for constraining undisclosed LLM-generated reviews. This paper also formalizes a PDF-based trap architecture that is adversarially robust.

2 Input-side Intuition

Recent studies have shown that some authors have embedded hidden instructions directly into submission PDFs to influence LLM-based reviewers, exploiting the fact that these models indiscriminately ingest non-visible document content. [6, 7] While such hidden-prompt manipulation is clearly prohibited and constitutes academic misconduct, its very capacity reveals an input-side leverage point rather than a mere vulnerability. At its core, this input-side influence implicates a public–private authority boundary, one that institutional design conventionally assigns to institutional authority rather than individual discretion. In the peer-review context, the most coherent use of such input-side influence is to implement it in a venue-wide, standardized, and transparent manner rather than leave it to individual actors.

[*] The author was affiliated with the Massachusetts Institute of Technology when this work was conducted. This work was carried out independently and is not affiliated with, nor endorsed by, the Massachusetts Institute of Technology.

3 Governance Framework

The framework does not aim to prohibit the use of LLMs in peer review, but to prevent pure LLM reviewing, defined here as evaluation that relies entirely on the LLM rather than on any human reading or assessment of the paper. To address this, the venue embeds traps into submission files so that the output of pure LLM reviewers can be identified. The venue also formally warns reviewers that traps are present in submissions, a disclosure that preserves the procedural legitimacy of this measure without revealing details that would accelerate its circumvention. To maintain fairness and avoid arbitrary intervention, only the venue may insert traps through a standardized programmatic process. Authors are not permitted to introduce such traps on their own. Automatic detection results should be treated only as preliminary indications, and reviewers must have an accessible and timely avenue for appeal to avoid mistaken classifications, as the aim is to support responsible reviewing rather than to cause undue alarm.

This framework operates as a dynamic adversarial game. Its aim is to maintain a structural cost asymmetry in which evading these traps consistently requires more effort than conducting a human review of the submission. Reviewers are likely to develop and circulate methods for removing such traps, a pattern that lies beyond the venue's effective influence. The framework's dynamic character is reflected in the fact that once particular traps are bypassed, the venue must update and rotate its techniques to sustain the cost structure described above. Consequently, the framework relies not on voluntary reviewer restraint but on a cost structure that deters pure LLM reviewing.

4 Trap Design Principles

The core design principle of input-side traps is that the embedded signal must be invisible to humans yet reliably ingested by the LLM. LLMs operate on tokenized text, so any PDF is processed by a PDF-to-LLM input pipeline that first converts it into text. This pipeline exposes a variety of non-visible document surfaces such as metadata fields, alt-text entries, invisible character spans, and other non-visible encodings. [6, 7] Taken together, these properties make input-side traps technically feasible.

Traps can be injected as natural-language instructions, including override-style hidden prompts [6, 7] and pseudo-system or urgent-message instructions [8]. These instructions can embed local traces such as literal echoes of injected text [7, 8] and changes to specific fields [6]. They can also embed global traces such as lexical or structural patterns [6, 7] and token-level watermark patterns [4, 9]. Traps can also be injected via invisible characters, which are retained in the model's output and serve as detectable traces. [9]

To remain robust under adversarial attempts to sanitize or suppress these signals, input-side traps need to satisfy several design properties, including (i) human-invisible yet pipeline-stable, (ii) paraphrase-resistant, (iii) multi-surface redundancy, (iv) per-submission randomness, (v) easy to insert at scale, and (vi) upgradeable against evasion. Among these, multi-surface redundancy is particularly crucial. Prior studies [6–8] show that distributing injection signals across multiple document surfaces markedly increases trigger robustness and the likelihood that embedded instructions are realized in the model's output. In our framework, this redundancy sharply raises the effective cost of evasion. Consequently, in practice, deployments should combine the methods described above.

5 Extensions

Beyond peer review, the same framework applies to other settings where undisclosed LLM misuse is a concern, including online examinations, crowdsourced annotation, and hiring or coding assessments.

References

[1] Nicholas Lo Vecchio. "Personal Experience with AI-generated Peer Reviews: A Case Study". In: *Research Integrity and Peer Review* 10.1 (Apr. 7, 2025), p. 4. ISSN: 2058-8615. DOI: 10.1186/s41073-025-00161-3. URL: https://doi.org/10.1186/s41073-025-00161-3.

[2] Giuseppe Russo Latona et al. *The AI Review Lottery: Widespread AI-Assisted Peer Reviews Boost Paper Scores and Acceptance Rates*. May 3, 2024. DOI: 10.48550/arXiv.2405.02150. arXiv: 2405.02150 [cs]. URL: http://arxiv.org/abs/2405.02150. Pre-published.

[3] ICLR Conference. *Statement on LLM-generated Peer Reviews*. X (formerly Twitter). Nov. 16, 2025. URL: https://x.com/iclr_conf/status/1990204431959470099.

[4] Vishisht Srihari Rao et al. "Detecting LLM-generated Peer Reviews". In: *PLOS ONE* 20.9 (Sept. 22, 2025), e0331871. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0331871. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0331871.

[5] Weixin Liang et al. *Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews*. June 15, 2024. DOI: 10.48550/arXiv.2403.07183. arXiv: 2403.07183 [cs]. URL: http://arxiv.org/abs/2403.07183. Pre-published.

[6] Changjia Zhu et al. *When Your Reviewer Is an LLM: Biases, Divergence, and Prompt Injection Risks in Peer Review*. Sept. 12, 2025. DOI: 10.48550/arXiv.2509.09912. arXiv: 2509.09912 [cs]. URL: http://arxiv.org/abs/2509.09912. Pre-published.

[7] Matteo Gioele Collu et al. *Publish to Perish: Prompt Injection Attacks on LLM-Assisted Peer Review*. Aug. 29, 2025. DOI: 10.48550/arXiv.2508.20863. arXiv: 2508.20863 [cs]. URL: http://arxiv.org/abs/2508.20863. Pre-published.

[8] Reworr and Dmitrii Volkov. *LLM Agent Honeypot: Monitoring AI Hacking Agents in the Wild*. Feb. 10, 2025. DOI: 10.48550/arXiv.2410.13919. arXiv: 2410.13919 [cs]. URL: http://arxiv.org/abs/2410.13919. Pre-published.

[9] Yepeng Liu et al. *In-Context Watermarks for Large Language Models*. May 22, 2025. DOI: 10.48550/arXiv.2505.16934. arXiv: 2505.16934 [cs]. URL: http://arxiv.org/abs/2505.16934. Pre-published.