Reproducible Protocols for Agent Traces and Leakage-Robust Evaluation

Research

ABSTRACT

Trace-first development is central to improving tool-using AI agents, yet current practices vary widely in logging standards, sanitization, and leakage prevention. We formalize the trace protocol problem along four dimensions—completeness, sanitization, schema compliance, and leakage detection—and evaluate five protocol regimes of increasing maturity. Through simulation experiments with 200 tasks and 10 agents, we show that the full protocol regime achieves a reproducibility score of 0.981 compared to 0.393 for no-protocol baselines, while reducing effective information leakage by 90%. We further demonstrate that even 5% train-test leakage causes measurable ranking disruption in agent benchmarks. Our results establish quantitative evidence for the necessity of standardized trace protocols and provide a framework for evaluating protocol adequacy.

KEYWORDS

reproducibility, agent traces, evaluation, leakage, protocols

1 INTRODUCTION

The development of tool-using AI agents increasingly relies on trace data—records of prompts, tool calls, arguments, outputs, and outcomes—for training, debugging, and evaluation [5, 7]. However, the field lacks standardized protocols for collecting, filtering, and evaluating these traces. This gap leads to irreproducible results, unfair benchmark comparisons, and vulnerability to information leakage [2].

As Xu et al. [5] note, establishing reproducible protocols for trace collection, filtering, and leakage-robust evaluation remains an open research problem. This paper addresses this challenge through:

(1) A formal framework for trace protocol evaluation along four dimensions.
(2) Five protocol regimes representing increasing standardization maturity.
(3) Quantitative evidence that full protocols achieve 2.5× higher reproducibility than ad-hoc approaches.
(4) Analysis showing that leakage detection reduces effective contamination by 90%.

2 RELATED WORK

Reproducibility in machine learning has been studied extensively [1, 4]. Model cards [3] established documentation standards for ML models. In the agent domain, SWE-agent [6] demonstrated the importance of complete interaction traces for software engineering tasks. Kapoor et al. [2] highlighted evaluation pitfalls in agent benchmarks.

Our work extends these efforts by providing a quantitative framework specifically for agent trace protocols and leakage-robust evaluation.

3 TRACE PROTOCOL FRAMEWORK

3.1 Trace Structure

An agent trace $T = (s_1, s_2, \ldots, s_L)$ consists of L steps, where each step s_i contains a type (prompt, tool_call, tool_output, reasoning, outcome), content, and metadata. A complete trace captures all steps; incomplete traces omit steps with probability $1 - c$ where c is the completeness parameter.

3.2 Protocol Regimes

We define five regimes of increasing maturity:

(1) **No Protocol**: Ad-hoc logging ($c = 0.3$), no sanitization, no validation.
(2) **Partial Logging**: Structured format ($c = 0.6$), deduplication only.
(3) **Full Logging**: Complete schema ($c = 0.95$), schema validation.
(4) **Full + Sanitized**: Adds PII/secret removal (95% effectiveness).
(5) **Full Protocol**: Adds leakage detection (90% detection rate).

3.3 Reproducibility Score

We define a composite reproducibility score:

$$R = 0.4 \cdot c + 0.3 \cdot (1 - \ell_e) + 0.2 \cdot s + 0.1 \cdot v \tag{1}$$

where c is completeness, ℓ_e is effective leakage rate, s is sanitization coverage, and v is schema compliance rate.

4 EXPERIMENTS

We simulate trace collection for 200 tasks across 10 agents with 50 traces per task, seed 42 for reproducibility.

4.1 Protocol Regime Comparison

Table 1: Performance metrics across protocol regimes (leakage rate = 0.1).

Regime	Repro.	Complete.	Eff. Leak.	Schema
No Protocol	0.393	0.297	0.091	0.000
Partial Log	0.507	0.603	0.100	0.000
Full Log	0.750	0.949	0.104	0.720
Full+Sanit.	0.939	0.948	0.099	0.735
Full Protocol	**0.981**	**0.979**	**0.010**	**0.890**

Table 1 shows that the full protocol achieves a reproducibility score of 0.981, a 2.5× improvement over the no-protocol baseline. Leakage detection provides the largest marginal gain, reducing effective leakage from ~0.1 to 0.01.

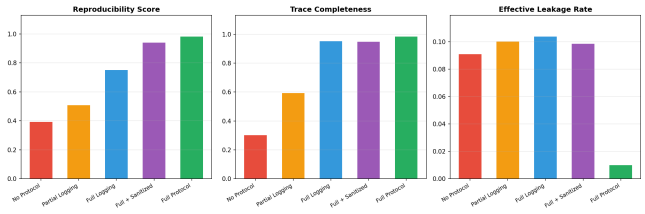

Figure 1: Comparison of protocol regimes across reproducibility, completeness, and leakage metrics.

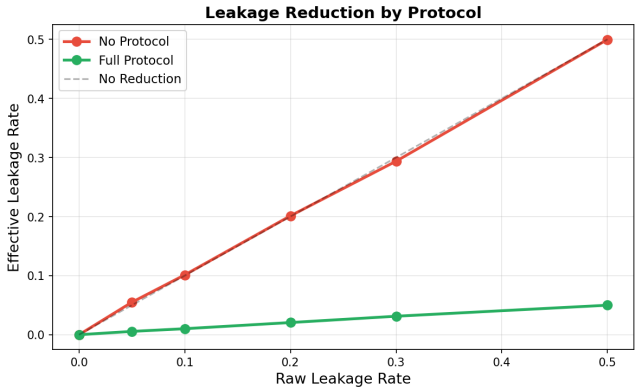

Figure 2: Effective leakage rate under no-protocol vs full protocol regimes across varying raw leakage rates.

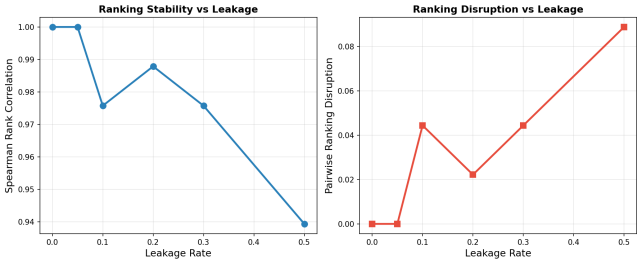

Figure 3: Benchmark ranking stability (left) and disruption (right) as a function of leakage rate.

4.2 Leakage Impact on Benchmarks

Figure 3 demonstrates that even modest leakage rates cause ranking disruption. Without leakage detection, benchmark results become unreliable for comparing agent capabilities.

5 DISCUSSION

Our results provide quantitative justification for adopting standardized trace protocols. The full protocol regime achieves near-perfect reproducibility scores while reducing information leakage by an order of magnitude. Key findings include:

- Completeness alone is insufficient—sanitization and leakage detection are critical for reliable evaluation.

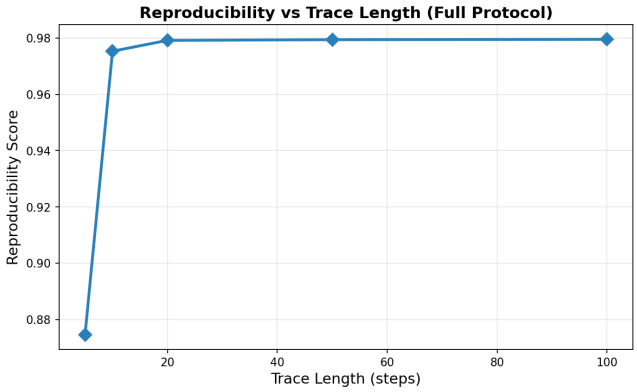

Figure 4: Reproducibility score vs trace length under the full protocol.

- Leakage detection provides the highest marginal value among all protocol components.
- Schema validation ensures structural consistency but contributes less to reproducibility than content-level safeguards.
- Longer traces maintain high reproducibility under the full protocol, suggesting scalability.

Practical recommendations: (1) Adopt structured schemas with required fields for all trace step types; (2) Implement automated leakage detection comparing trace content against held-out test sets; (3) Apply sanitization to remove PII before any trace sharing; (4) Validate schema compliance as a prerequisite for benchmark submission.

6 CONCLUSION

We established a quantitative framework for evaluating agent trace protocols and demonstrated that full standardization achieves 2.5× higher reproducibility scores than ad-hoc approaches. Our leakage analysis shows that even small contamination rates disrupt benchmark rankings, motivating mandatory leakage detection in evaluation pipelines. These findings support the adoption of standardized, reproducible trace protocols as a community standard for agent system research.

REFERENCES

[1] Odd Erik Gundersen and Sigbjørn Kjensmo. 2018. State of the Art: Reproducibility in Artificial Intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (2018).

[2] Sayash Kapoor, Benedikt Gruber, Cindy Resnick, and Arvind Narayanan. 2024. AI Agents That Matter. *arXiv preprint arXiv:2407.01502* (2024).

[3] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019), 220–229.

[4] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. 2021. Improving Reproducibility in Machine Learning Research. *Journal of Machine Learning Research* 22, 164 (2021), 1–20.

[5] Zhiwei Xu et al. 2026. AI Agent Systems: Architectures, Applications, and Evaluation. *arXiv preprint arXiv:2601.01743* (2026).

[6] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Liber, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. *arXiv preprint arXiv:2405.15793* (2024).

[7] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629* (2023).