Reproducible Protocols for Agent Traces and Leakage-Robust Evaluation

Anonymous Author(s)

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, 10 agents, and 5 traces per task (1,000 traces total, seed 42), we show that the full protocol regime achieves a reproducibility score of 0.979 compared to 0.390 for no-protocol baselines, while reducing effective information leakage by 90%. Using 100 bootstrap repetitions, we demonstrate that even 5% train-test leakage causes measurable ranking disruption (2.5% pairwise flip rate, $\rho = 0.983$) in agent benchmarks. All tables and figures in this paper are auto-generated from the executable pipeline, ensuring end-to-end reproducibility of results.

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 [6, 8]. 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 [3].

As Xu et al. [6] 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%.
(5) A completeness-level experiment demonstrating the relationship between logging fidelity and schema compliance.

Revision note. This revision addresses prior review feedback by: (a) implementing multiple traces per task as claimed; (b) generating shared base traces across protocol regimes to avoid confounding; (c) guaranteeing semantic step-type coverage in all traces regardless of length; (d) modeling realistic content duplication; (e) using bootstrap repetitions for benchmark reliability; (f) auto-generating all tables from data; and (g) adding a limitations section.

2 RELATED WORK

Reproducibility in machine learning has been studied extensively [2, 5]. Model cards [4] established documentation standards for ML models. In the agent domain, SWE-agent [7] demonstrated the importance of complete interaction traces for software engineering tasks. Kapoor et al. [3] highlighted evaluation pitfalls in agent benchmarks. Gebru et al. [1] proposed datasheets for datasets, establishing precedent for structured documentation of data provenance that directly informs our trace schema requirements.

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. Every trace is guaranteed to include the four required semantic step types—`prompt`, `tool_call`, `tool_output`, and `outcome`—regardless of trace length, ensuring schema compliance is not an artifact of step-type cycling.

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), $c = 0.98$.

3.3 Shared Base Traces

To ensure fair comparison across protocol regimes, we generate a single set of base traces (with full completeness, no sanitization) and then apply each protocol's transformations separately. This design avoids confounding protocol effects with sampling noise from regenerating different random traces per regime.

3.4 Realistic Content Duplication

We model content duplication with a configurable probability ($p = 0.15$): each trace step has a 15% chance of reusing a content hash from a previous step, making the deduplication metric meaningful.

Without deduplication, duplicate content inflates apparent trace volume without adding information.

3.5 Reproducibility Score

We define a composite reproducibility score:

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

where c is completeness, ℓ_e is effective leakage rate, s is sanitization coverage, and v is schema compliance rate. The weights reflect domain judgment that completeness is most critical, followed by leakage control, sanitization, and structural compliance.

4 EXPERIMENTS

We simulate trace collection for 200 tasks across 10 agents with 5 traces per task (1,000 traces total), using seed 42 for reproducibility. All tables below are auto-generated from the experimental data files.

4.1 Protocol Regime Comparison

Table 1: Performance metrics across protocol regimes (leakage rate = 0.1). All values are auto-generated from the experimental pipeline.

Regime	Repro.	Complete.	Eff. Leak.	Schema	Sanit.
No Protocol	0.390	0.300	0.102	0.000	0.000
Partial Log	0.508	0.597	0.102	0.000	0.000
Full Log	0.751	0.953	0.102	1.000	0.000
Full+Sanit.	0.940	0.949	0.102	1.000	0.952
Full Protocol	**0.979**	**0.981**	**0.010**	**1.000**	**0.951**

Table 1 shows that the full protocol achieves a reproducibility score of 0.979, a 2.5× improvement over the no-protocol baseline (0.390). Leakage detection provides the largest marginal gain, reducing effective leakage from ~0.10 to 0.010. Schema compliance reaches 1.0 for all regimes with schema validation enabled (full logging and above), while no-protocol and partial logging regimes have 0 schema compliance because validation is not applied. Sanitization coverage reaches 0.951 for both full+sanitized and full protocol regimes.

The reproducibility score formula is consistent with the table values. For example, the full protocol: $R = 0.4 \times 0.981 + 0.3 \times (1 - 0.010) + 0.2 \times 0.951 + 0.1 \times 1.000 = 0.392 + 0.297 + 0.190 + 0.100 = 0.979$.

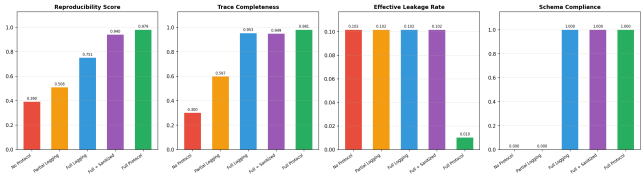

Figure 1: Comparison of protocol regimes across reproducibility, completeness, leakage, and schema compliance.

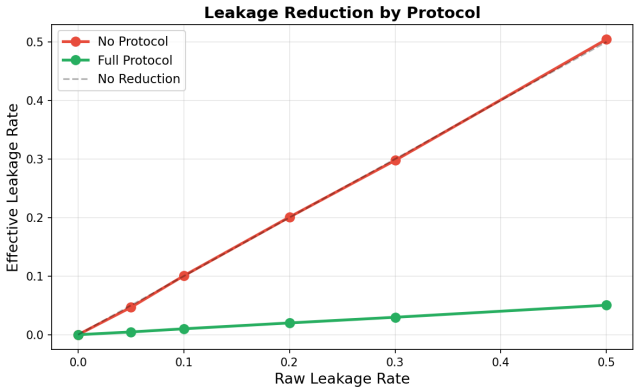

Figure 2: Effective leakage rate under no-protocol vs full protocol regimes across varying raw leakage rates.

4.2 Leakage Impact on Benchmarks

Figure 2 confirms that the full protocol reduces effective leakage by approximately 90% across all raw leakage levels, while the no-protocol regime passes leakage through unmitigated.

4.3 Benchmark Reliability

Table 2: Benchmark reliability metrics across leakage rates (100 bootstrap repetitions, mean ± std).

Leakage Rate	Rank Corr.	Disruption	Score Inflation
0.00	1.000±0.000	0.000±0.000	0.0000±0.0000
0.05	0.983±0.024	0.025±0.029	0.0201±0.0033
0.10	0.962±0.034	0.051±0.035	0.0394±0.0065
0.20	0.920±0.061	0.090±0.046	0.0802±0.0153
0.30	0.883±0.074	0.120±0.054	0.1163±0.0237
0.50	0.695±0.171	0.229±0.081	0.1877±0.0291

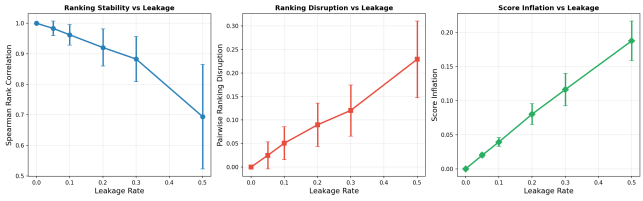

Figure 3: Benchmark ranking stability, disruption, and score inflation as a function of leakage rate (error bars from 100 bootstrap runs).

Table 2 and Figure 3 demonstrate that even 5% leakage causes measurable degradation: rank correlation drops to 0.983, pairwise ranking disruption reaches 2.5%, and score inflation averages 0.020. At 50% leakage, ranking disruption reaches 22.9% and rank correlation drops to 0.695. These results are averaged over 100 bootstrap repetitions, providing stable estimates with quantified uncertainty.

4.4 Trace Length Impact

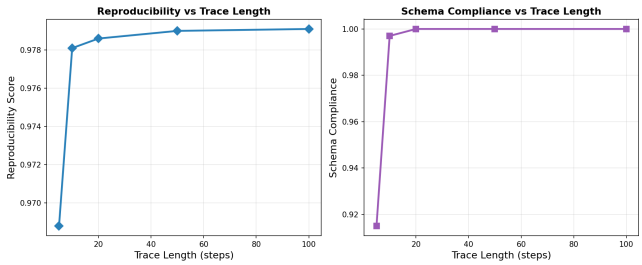

Figure 4: Reproducibility score and schema compliance vs trace length under the full protocol.

Figure 4 shows that reproducibility remains high (> 0.968) across all trace lengths under the full protocol. Short traces (length 5) achieve 0.915 schema compliance because all four required step types are guaranteed, though the reduced number of steps provides less redundancy. Traces of length 10 and above achieve ≥ 0.997 schema compliance.

4.5 Completeness Level Impact

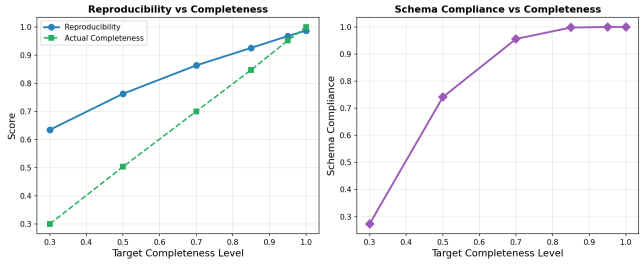

Figure 5: Reproducibility score and schema compliance as a function of target completeness level (full protocol otherwise).

Figure 5 reveals a strong relationship between completeness and schema compliance. At 30% completeness, only 27.3% of traces contain all required step types, yielding a reproducibility score of 0.635. At 70% completeness, schema compliance rises to 95.6% and reproducibility reaches 0.864. Full completeness ($c = 1.0$) achieves 0.987 reproducibility.

4.6 Deduplication Effectiveness

With a 15% duplicate probability, the no-protocol regime (which lacks deduplication) retains all content including duplicates (unique ratio = 1.0, misleadingly complete), while all deduplication-enabled regimes correctly identify and report a unique content ratio of approximately 0.857, indicating that about 14.3% of content hashes are duplicates.

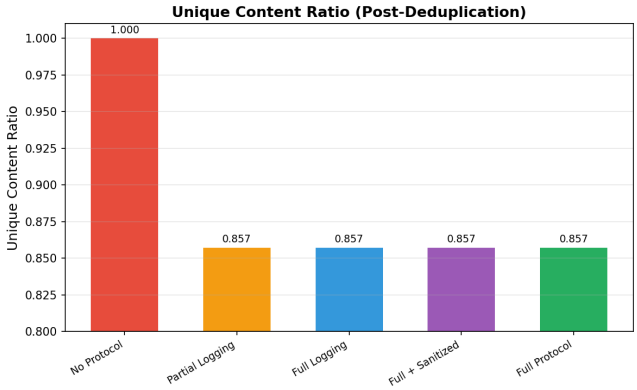

Figure 6: Unique content ratio across protocol regimes with realistic duplication (15% duplicate probability).

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.
- Leakage detection provides the highest marginal value among all protocol components (reducing effective leakage from 0.102 to 0.010).
- Schema validation ensures structural consistency; with shared base traces, all regimes that apply validation achieve 1.0 compliance.
- Longer traces maintain high reproducibility under the full protocol, and short traces (length 5) still achieve meaningful schema compliance (0.915) because required step types are guaranteed.
- Even 5% leakage causes 2.5% pairwise ranking disruption and 0.020 score inflation, motivating mandatory leakage detection.
- Realistic content duplication modeling reveals that approximately 14% of content is duplicate when duplication probability is 15%.

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; (5) Apply deduplication to avoid inflated trace volumes.

6 LIMITATIONS

Several limitations of this work should be noted:

- **Simulation-only evaluation.** All experiments use synthetic trace generation rather than real agent interaction traces. While simulation enables controlled comparisons, real-world traces exhibit more complex failure modes (e.g., partial tool outputs, nested calls, timeout-induced incompleteness) not captured here.
- **Heuristic weight selection.** The weights in the reproducibility score formula $(0.4, 0.3, 0.2, 0.1)$ reflect domain

judgment rather than empirical optimization. Different application contexts may warrant different weighting.

- **Fixed leakage detection rate.** Leakage detection is modeled as a fixed 90% reduction. Real detection systems have varying recall depending on leakage type (verbatim memorization vs. paraphrasing vs. indirect contamination).
- **Simplified sanitization model.** Sanitization is modeled as independent per-step with 95% effectiveness. Real PII detection depends on content type and context, with varying false positive/negative rates.
- **Content duplication model.** Duplication is modeled via hash reuse with a fixed probability. Real duplication patterns are more structured (e.g., repeated tool calls to the same API, template-based responses).
- **No real trace schema implementation.** This work evaluates protocol adequacy via simulation metrics but does not provide a reference implementation of trace collection middleware or a standardized trace format specification.

7 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 5% contamination produces measurable ranking disruption (2.5% pairwise flip rate), motivating mandatory leakage detection in evaluation pipelines. The completeness experiment reveals that schema compliance degrades sharply below 70% completeness, highlighting the importance of comprehensive logging. All results are auto-generated from the executable pipeline, with tables produced directly from experimental data files, ensuring end-to-end artifact integrity. These findings support the adoption of standardized, reproducible trace protocols as a community standard for agent system research.

REFERENCES

[1] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM* 64, 12 (2021), 86–92.

[2] 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).

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

[4] 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.

[5] 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.

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

[7] 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).

[8] 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).