

TRICE: Deterministic Context Control for Live Software Agents

TraceRazor Research - 2026-06-21

Abstract

TRICE makes context assembly, evidence manifests, live verification, and acceptance gates deterministic. Verifier durations are normalized before hashing and wall-clock fields are excluded from evidence. On six bundled live repository tasks, TRICE reduces measured input context by 78.6% (95% bootstrap CI: 76.8% to 80.3%) while preserving executable task success.

Method

TRICE chooses keep, extract, summarize, mask-with-receipt, anchor-prefix, or lazy-recall actions under a token budget. Acceptance requires live verifier pass preservation and measured input savings above the user-conditioned target.

Task	Baseline	TRICE	Savings	Pass
csv-filter	1895	430	77.3%	yes
dedupe-helpers	1826	361	80.2%	yes
fix-imports	1806	341	81.1%	yes
fix-offby-one	2527	634	74.9%	yes
implement-median	1889	424	77.6%	yes
rename-api	1814	349	80.8%	yes
Mean	--	--	78.6%	yes

Replicated Suite Evidence

The public suite artifact fix-offby-one-suite contains 3 live replicate runs across 1 task cluster(s). Mean input-token savings is 76.6% with clustered-by-task 95% CI 76.6% to 76.6%; pass regressions 0. S-tier gate: not passed.

Portable Artifact Bundle

The public bundle trice_suite_evidence.trice.zip contains 35 hashed entries, root manifest trice_suite_evidence_manifest.json, and root result trice_suite_results.json. Verification replays aggregate and child-manifest checks after safe extraction.

Broad Smoke Evidence

The broad-smoke suite bundled-live-six-task-suite contains 6 live run(s) across 6 task cluster(s). Mean input-token savings is 81.5% with clustered-by-task 95% CI 79.0% to 83.5%; pass regressions 0. S-tier gate: not passed.

The broad-smoke bundle trice_broad_smoke_evidence.trice.zip contains 65 hashed entries.

Source Provenance

The suite source manifest records repo tree digest 604addf4070c... over 3 files and intervention provenance before execution. JSON patch tasks record patch SHA-256; command adapter tasks record command argv and timeout; adapter-profile tasks record profile SHA-256. Suite tasks may use either local paths or locked Git sources with URL, revision, optional subdirectory, and resolved commit recorded.

Product Contract

A TRICE result is accepted only when the public library emits a stable manifest, the live verifier passes, and replay is used only as preflight evidence. Generic users provide LiveTask plus a deterministic RepairAdapter; the bundled JSON patch and command repair adapters refuse test edits by default. Command adapters may be packaged as trice-adapter-profile/v1 files, and every live condition emits a trice-run-receipt/v1 artifact with adapter envelope, command hash, before/after workspace fingerprints, changed files, output hashes, and optional agent-reported token accounting. Receipts have a shipped JSON Schema and are validated during manifest and bundle verification. Identical real-run smoke executions must reproduce result and artifact hashes. Multi-repository evaluation uses trice-suite/v1 manifests, local or locked Git sources, verify-suite for aggregate plus child-manifest verification, and deterministic .trice.zip evidence bundles for handoff. The public CLI is tracerazor-trice and mirrors `python -m tracerazor.trice`.

Deterministic Claim Gate

Local smoke gate passed: yes. Broad claim allowed: no. Rationale: local deterministic smoke passed; broad claim still requires held-out provider runs with repeated trials and clustered confidence intervals

Selected References

- [agentsmatter2024] AI Agents That Matter. <https://arxiv.org/html/2407.01502v1>
- [acmartifact] ACM Artifact Review and Badging. <https://www.acm.org/publications/policies/artifact-review-badging>
- [mohammadi2025agentsurvey] Evaluation and Benchmarking of LLM Agents: A Survey. <https://arxiv.org/html/2507.21504v1>
- [complexmcp2026] ComplexMCP: Evaluation of LLM Agents in Dynamic Tool Sandboxes. <https://arxiv.org/html/2605.10787v2>
- [reasonbench2025] ReasonBENCH: Benchmarking the Instability of LLM Reasoning. <https://arxiv.org/html/2512.07795v2>
- [dfah2026] Replayable Financial Agents: A Determinism-Faithfulness Assurance Harness. <https://arxiv.org/html/2601.15322v1>
- [terminalagents2026] Building AI Coding Agents for the Terminal. <https://arxiv.org/html/2603.05344v1>
- [uncertainty2024] Towards Reproducible LLM Evaluation. <https://arxiv.org/html/2410.03492v2>
- [externalization2026] Externalization in LLM Agents. <https://arxiv.org/html/2604.08224v1>
- [sweuniverse2026] SWE-Universe: Scale Real-World Verifiable Environments. <https://arxiv.org/html/2602.02361v1>
- [swebench2023] SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. <https://arxiv.org/abs/2310.06770>
- [sweagent2024] SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. <https://arxiv.org/abs/2405.15793>
- [acon2025] Acon: Optimizing Context Compression for Long-Horizon LLM Agents. <https://arxiv.org/html/2510.00615v1>
- [personalizedagents2026] Learning Personalized Agents from Human Feedback. <https://arxiv.org/html/2602.16173v1>
- [nlharness2026] Natural-Language Agent Harnesses. <https://arxiv.org/html/2603.25723v1>
- [swtbench2024] SWT-Bench: Testing and Validating Real-World Software Fixes. <https://arxiv.org/html/2406.12952v3>
- [swebenchcl2025] SWE-Bench-CL: Continual Learning for Software Engineering Agents. <https://arxiv.org/abs/2507.00014>
- [rigorousagentbench2025] Establishing Best Practices for Building Rigorous Agentic Benchmarks. <https://arxiv.org/html/2507.02825v5>
- [sparkllmeval2026] Spark-LLM-Eval: A Distributed Framework for Statistically Rigorous LLM Evaluation. <https://arxiv.org/html/2603.28769v1>
- [agentloganalysis2026] Log Analysis is Necessary for Credible Evaluation of AI Agents. <https://arxiv.org/html/2605.08545v1>
- [topline2026] Topline Benchmark Performance. <https://arxiv.org/html/2605.28065v1>
- [setupbench2025] SetupBench: Assessing Software Engineering Agents' Ability to Bootstrap Development Environments. <https://arxiv.org/abs/2507.09063>
- [pairedbca2025] A Paired Bootstrap Protocol for Evaluating Small Improvements. <https://arxiv.org/pdf/2511.19794>
- [llmlingua2023] LLMLingua: Compressing Prompts for Accelerated Inference. <https://arxiv.org/abs/2310.05736>
- [longllmlingua2023] LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios. <https://arxiv.org/abs/2310.06839>
- [promptcache2023] Prompt Cache: Modular Attention Reuse for Low-Latency Inference. <https://arxiv.org/abs/2311.04934>
- [kvflow2025] KVFlow: Efficient Serving for LLM Agent Workflows. <https://arxiv.org/abs/2507.07400>
- [agentdiet2025] AgentDiet: Efficient Context Pruning for LLM Agents. <https://hf.co/papers/2509.23586>
- [react2022] ReAct: Synergizing Reasoning and Acting in Language Models. <https://arxiv.org/abs/2210.03629>
- [reflexion2023] Reflexion: Language Agents with Verbal Reinforcement Learning. <https://arxiv.org/abs/2303.11366>
- [sweevo2025] SWE-EVO: Benchmarking Coding Agents in Long-Horizon Software Evolution. <https://arxiv.org/html/2512.18470v6>