Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-26)
Demonstrations Track

TraceBrain: An Open-Source Framework for Agentic Trace Management

Quy Minh Le[1], **Oscar Cao**[2], **Hoang Quoc Viet Pham**[1], **Hoang Thanh Lam**[3*] and **Hoang D. Nguyen**[2†]

[1]ToolBrain Research, Ireland
[2]University College Cork, Ireland
[3]IBM Research Lab, Dublin, Ireland
t.l.hoang@ie.ibm.com, hn@cs.ucc.ie

Abstract

As Large Language Model (LLM) agents scale toward real-world deployment, they generate large volumes of fragmented, non-standardized execution traces. Many existing observability platforms treat these traces primarily as passive logging artifacts, lacking the unified infrastructure to operationalize them for active governance and agent adaptation across heterogeneous single-agent and multi-agent workflows. To address this gap, we introduce **TraceBrain**, an open-source infrastructure for autonomous agent trace management. TraceBrain adopts a framework-agnostic architecture built on a delta-based OpenTelemetry (OTLP) schema, which mitigates context explosion and supports on-demand reconstruction of long-horizon execution trajectories. For runtime governance, TraceBrain implements uncertainty-driven supervision, where an internal Trace Evaluator prioritizes ambiguous trajectories for human review, thereby reducing manual annotation workload. Moving beyond passive observation, the platform incorporates a hybrid semantic-lexical retrieval engine that combines dense vector similarity and exact keyword matching for operational memory retrieval. Furthermore, an automated curriculum mechanism continuously synthesizes failure patterns into structured training artifacts. Empirical evaluations demonstrate a $\sim$100$\times$ reduction in storage overhead together with high precision in uncertainty-guided trace supervision. Ultimately, TraceBrain transforms the execution history into a reusable operational memory substrate, bridging runtime observability with retrieval-driven agent adaptation. The system is publicly available at https://github.com/ToolBrain/TraceBrain.

1 Introduction

Recent advances in Large Language Model (LLM) agents have enabled long-horizon workflows involving reasoning, tool use, and multi-agent interaction [Yao *et al.*, 2023; Wu *et al.*, 2024]. As these agents transition to real-world deployments, they generate large volumes of execution traces, aligning with emerging observability standards [OpenTelemetry Authors, 2026; Arize AI, 2023]. However, in practice, these traces are often fragmented across heterogeneous frameworks. As a result, practitioners lack a unified infrastructure to systematically manage, analyze, and operationalize trace data for continual agent adaptation.

This gap presents a major challenge for the operational governance of autonomous agents. Many existing observability systems primarily treat traces as passive logging artifacts, providing limited support for active supervision and trace-driven adaptation. Although recent work explores self-reflection and experiential learning in autonomous agents [Shinn *et al.*, 2023; Wang *et al.*, 2023], there remains insufficient operational support to transform execution traces into structured supervision and learning assets. To address these limitations, we introduce **TraceBrain**, a unified infrastructure for trace collection, governance, and adaptation, making the following contributions:

- **Standardized ingestion and hybrid retrieval.** TraceBrain adopts a framework-agnostic, delta-based OpenTelemetry (OTLP) [OpenTelemetry Authors, 2026] schema. This design mitigates context duplication, enabling efficient trajectory reconstruction. Furthermore, TraceBrain incorporates a hybrid semantic-lexical retrieval engine based on Reciprocal Rank Fusion (RRF) [Cormack *et al.*, 2009] to support operational trace querying.
- **Uncertainty-driven supervision.** TraceBrain implements a semi-automated feedback pipeline where an internal LLM-based Trace Evaluator generates annotations and requests expert input only when uncertainty arises. Conceptually similar to uncertainty-based active learning [Settles, 2009], this design improves the efficiency of human-in-the-loop monitoring by prioritizing critical or borderline cases.
- **Trace-driven adaptation.** TraceBrain continuously mines failure patterns from operational traces and synthesizes them into structured curricula and retrieval-ready trajectories, supporting downstream fine-tuning and retrieval-augmented adaptation [Bengio *et al.*, 2009; Lewis *et al.*, 2020].

*Corresponding authors.

- **Comprehensive system evaluation.** We provide quantitative assessments demonstrating TraceBrain's scalability and operational trade-offs. Empirical results show a $\sim$100$\times$ reduction in storage overhead compared to naive logging, stable hybrid retrieval latency (median 96.9 ms, P_{95} 122.6 ms), and high precision in uncertainty-guided supervision.

We demonstrate TraceBrain through representative multi-agent coordination scenarios involving real-time trace inspection, uncertainty-driven supervision, and trace-driven adaptation. By integrating observability, governance, and retrieval-driven learning within a unified infrastructure, TraceBrain provides a practical foundation for scalable autonomous agent systems.

2 Background and Related Work

The growing deployment of autonomous LLM agent systems [Yao *et al.*, 2023; Wu *et al.*, 2024] has increased the need for scalable infrastructure to manage execution traces. Existing systems primarily treat traces as passive debugging artifacts rather than reusable operational data. TraceBrain addresses these limitations through observability, supervision, and trace-driven adaptation.

Agent Observability and Tracing. Platforms such as *LangSmith* [LangChain, 2026] and OTLP-based observability systems support instrumentation and monitoring, but largely focus on passive logging. Frameworks such as *AgentTrace* [Anonymous, 2026] extend scalable observability toward security monitoring, while *OpenInference* [Arize AI, 2023] standardizes LLM telemetry on top of OTLP but still relies on cumulative prompt logging. TraceBrain instead introduces a **delta-based OTLP schema** for incremental state transitions together with hybrid semantic-lexical trace retrieval.

Model Evaluation and Supervision. LLM-based evaluators have emerged as an effective approach for automated assessment [Zheng *et al.*, 2023]. While models such as *Prometheus* [Kim *et al.*, 2024] provide specialized evaluation capabilities, operational systems often lack uncertainty-aware human feedback mechanisms. *Agent Trajectory Explorer* [Desmond *et al.*, 2025] supports human review but remains limited to manual binary feedback. TraceBrain introduces an uncertainty-driven supervision pipeline in which an internal *LLM-based Trace Evaluator* selectively escalates low-confidence trajectories for expert review, improving supervision efficiency [Settles, 2009].

Continuous Agent Adaptation. Prior work has shown that agents can improve through self-reflection and experiential learning [Shinn *et al.*, 2023; Wang *et al.*, 2023]. However, most observability systems treat traces as terminal artifacts. TraceBrain extends observability through automated curriculum synthesis inspired by curriculum learning [Bengio *et al.*, 2009], continuously mining historical failure patterns to generate structured adaptation signals for downstream refinement workflows.

Framework	Trace Storage	Feedback	Retrieval	Adaptation
LangSmith	Proprietary	Manual UI	×	×
Langfuse	Relational DB	Manual labels	×	×
Prometheus	Eval records	Automated only	×	Eval synthesis
Agent Traj. Exp.	Proprietary	Manual binary	×	×
TensorZero	Prop. tables	Code annotations	×	×
TraceBrain (Ours)	**Delta-OTLP**	**Uncertainty-Driven**	**Hybrid**	**Curriculum**

Table 1: Comparison of Agent Observability Infrastructure

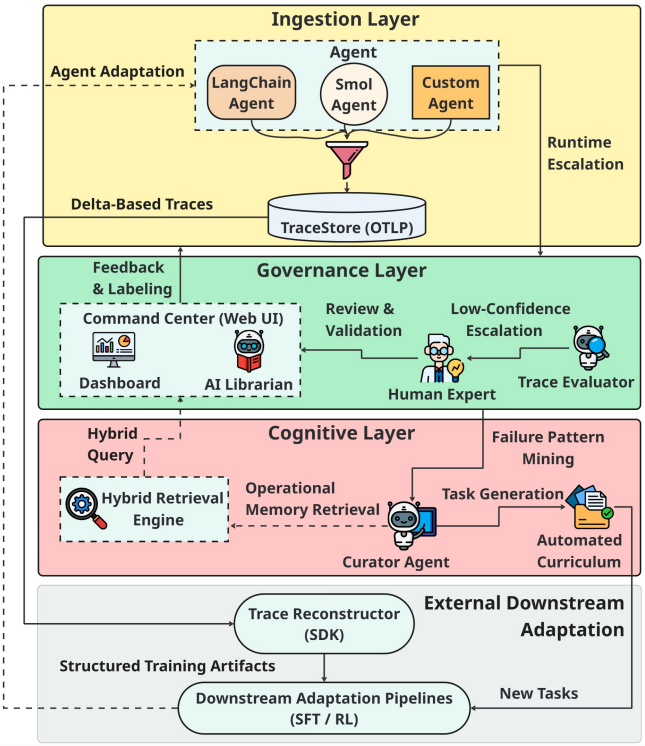

Figure 1: TraceBrain operates as a closed-loop infrastructure for trace-driven governance and adaptation. The Ingestion Layer converts heterogeneous agent executions into reconstructable delta-based OTLP trajectories. The Governance Layer performs runtime supervision through uncertainty-driven evaluation and human validation. The Cognitive Layer supports hybrid semantic-lexical retrieval and automated curriculum synthesis, enabling downstream adaptation through reusable operational traces and structured training artifacts.

3 System Architecture

TraceBrain organizes trace management through three layers (Figure 1): ingestion, governance, and cognitive adaptation.

3.1 Ingestion Layer: Structuring Execution

Agent frameworks such as LangChain [LangChain, 2026] and SmolAgents [Roucher *et al.*, 2025] produce heterogeneous execution logs. The Ingestion Layer standardizes these activities through a delta-based *OpenTelemetry (OTLP)* schema [OpenTelemetry Authors, 2026], recording incremental state updates rather than cumulative prompt histories. Conversational state is reconstructed on demand through

hierarchical parent-child execution links. The **TraceClient SDK** provides consistent instrumentation across heterogeneous frameworks.

3.2 Governance Layer: Enabling Runtime Supervision

The **TraceStore** exposes runtime governance interfaces for:

- **Runtime Intervention.** TraceBrain supports runtime escalation for anomalous or unstable execution behaviors, enabling supervisory intervention beyond passive post-hoc debugging.
- **Uncertainty-Driven Supervision.** An internal *LLM-based Trace Evaluator* assigns confidence-aware evaluations to trajectories using previously validated expert feedback. Only low-confidence traces are escalated for human review, improving supervision efficiency.

3.3 Cognitive Layer: Supporting Trace-Driven Adaptation

The Cognitive Layer transforms stored traces into structured learning assets.

- **Experience Retrieval.** Successful trajectories are indexed through a hybrid semantic-lexical retrieval engine, enabling runtime retrieval of operational memory for retrieval-augmented adaptation [Lewis *et al.*, 2020].
- **Automated Curriculum.** An LLM-based curator mines clustered failure patterns to synthesize structured remediation tasks inspired by curriculum learning [Bengio *et al.*, 2009].

Curated traces and synthesized curricula are exportable in standardized formats (e.g., JSONL), enabling integration with downstream fine-tuning and evaluation pipelines.

4 System Demonstration and Evaluation

To demonstrate TraceBrain and evaluate its infrastructure scalability, we present a representative multi-agent scenario followed by quantitative system benchmarks. A demonstration video is available at https://youtu.be/rH3Nlsd4Ut8.

4.1 Scenario: Multi-Agent Coordination Failure

We demonstrate TraceBrain through a Collaborative Coding Environment implemented with LangChain [LangChain, 2026], featuring a Coder and a Reviewer agent. This scenario highlights TraceBrain's support for multi-agent observability.

Anomaly Detection and Semantic Diagnosis. During execution, the agents enter a cyclical negotiation loop caused by semantic misalignment. Using the TraceClient SDK, TraceBrain ingests these interactions through the delta-based schema, automatically flagging the trajectory as anomalous (internally categorized as `logic_loop`). The **Trace Explorer** visualizes the interaction as a hierarchical execution graph for localizing divergence points.

Uncertainty-Driven Supervision and Adaptation. Due to low evaluator confidence, the trajectory is prioritized for human review. The **Automated Curriculum** engine then synthesizes the labeled failure into a targeted remediation task (e.g., "Multi-Agent Conflict Resolution"). Reconstructed trajectories can be exported as JSONL artifacts for downstream adaptation workflows.

4.2 System Evaluation

We conducted quantitative benchmarks to evaluate TraceBrain's infrastructure scalability and supervision performance. The key results are summarized in Table 2.

Metric	Result & Impact
Storage Efficiency	Delta-based schema mitigates context explosion associated with naive cumulative logging. For a 200-step trajectory, storage dropped from 16.10 MB to **0.15 MB**, achieving a $\sim$100$\times$ reduction.
Reconstruction	Backward traversal overhead is minimal. Rebuilding a 500-span trajectory context requires only **3.64 ms** and 0.82 MB peak memory on the client side.
Ingestion vs. Retrieval	Hybrid indexing introduces write-time overhead (**41.8 RPS** under a throttled **20-concurrency** load) while maintaining stable retrieval latency (median 96.9 ms, P_{95} 122.6 ms) at 100k traces.
Labeling Efficiency	Evaluated on 75 complex traces from the TRAIL benchmark [Deshpande *et al.*, 2025]. The *LLM-based Trace Evaluator* achieved an **89% automatic acceptance rate** (> 0.8 confidence) and **0.93 priority precision**, indicating strong potential for reducing human annotation workload.

Table 2: Summary of TraceBrain System Evaluation

Table 2 demonstrates scalable retrieval and uncertainty-driven supervision with low storage overhead. The evaluator automatically processes routine trajectories while escalating severe anomalies for expert review.

5 Conclusion

We presented **TraceBrain**, an open-source infrastructure for trace-driven observability, supervision, and adaptation in autonomous LLM agents. TraceBrain introduces a delta-based OTLP schema that mitigates context explosion caused by cumulative prompt logging, while supporting uncertainty-driven supervision and retrieval-driven adaptation using historical execution traces. Empirical evaluations demonstrate substantial reductions in storage overhead together with scalable hybrid retrieval, efficient uncertainty-aware supervision, and robust trajectory reconstruction. Future work will explore automated workflow mining and fine-grained error attribution across long-horizon execution trajectories.

Ethics Statement

TraceBrain may process sensitive operational data. The platform supports self-hosted deployment and trace sanitization to mitigate privacy risks.

Acknowledgements

This publication has emanated from research supported in part by a grant from Research Ireland under Grant number [12-RC-2289-P2] which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

Contribution Statement

Quy Minh Le and Oscar Cao contributed equally to this work. All authors contributed to the design, implementation, and writing of this paper.

References

[Anonymous, 2026] Anonymous. Agenttrace: A structured logging framework for agent system observability. In *AAAI 2026 Workshop on Trust and Control in Agentic AI (TrustAgent)*, 2026.

[Arize AI, 2023] Arize AI. Openinference. https://github.com/Arize-ai/openinference, 2023. Accessed: 2026-05-14.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery.

[Cormack *et al.*, 2009] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA, 2009. Association for Computing Machinery.

[Deshpande *et al.*, 2025] Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization, 2025.

[Desmond *et al.*, 2025] Michael Desmond, Ja Young Lee, Ibrahim Ibrahim, James M. Johnson, Avirup Sil, Justin MacNair, and Ruchir Puri. Agent trajectory explorer: Visualizing and providing feedback on agent trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pages 29634–29636, 2025.

[Kim *et al.*, 2024] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[LangChain, 2026] LangChain. Langchain overview. https://docs.langchain.com/oss/python/langchain/overview, 2026. Accessed: 2026-02-12.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[OpenTelemetry Authors, 2026] OpenTelemetry Authors. OpenTelemetry Documentation. https://opentelemetry.io/, 2026. Accessed: 2026-02-12.

[Roucher *et al.*, 2025] Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 'smolagents': a smol library to build great agentic systems. https://github.com/huggingface/smolagents, 2025.

[Settles, 2009] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Shinn *et al.*, 2023] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc., 2023.

[Wang *et al.*, 2023] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. In *Second Agent Learning in Open-Endedness Workshop*, 2023.

[Wu *et al.*, 2024] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, Chi Wang, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling (COLM)*, 2024.

[Yao *et al.*, 2023] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023.