



(RESEARCH ARTICLE)



TELEOS: VALUE-AWARE ADAPTIVE TELEMETRY FOR PRIVACY-CONSTRAINED AND CAUSALLY USEFUL OBSERVABILITY IN AGENTIC AI SYSTEMS

Prudvi Saisaran Ponduru ^{1,*}, Pavani Priya Vyshnavi Nandanavanam ² and Sai Kesav Kumar Ponduru ³

¹ Department of Computer Science, University of the Potomac, Washington, D.C., USA.

² Department of Computer Science, California State University, Northridge, CA, USA.

³ Department of Business Analytics, University of Massachusetts, Amherst, MA, USA.

* Corresponding Author

ORCID Details

Prudvi Saisaran Ponduru: <https://orcid.org/0009-0009-5930-8265>

Pavani Priya Vyshnavi Nandanavanam: <https://orcid.org/0009-0003-1342-4291>

Sai Kesav Kumar Ponduru: <https://orcid.org/0009-0002-9400-5410>

World Journal of Advanced Research and Reviews, 2025, 27(03), 2043–2058

Publication history: Received on 28 July 2025; revised on 23 September 2025; accepted on 29 September 2025

Article DOI: <https://doi.org/10.30574/wjarr.2025.27.3.3148>

Copyright © 2025 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Agentic artificial intelligence systems generate model, prompt, retrieval, memory, tool, policy, security, application, and infrastructure telemetry, yet current observability practice largely treats collection as a static configuration problem. This study introduces TELEOS, a decision-theoretic framework that treats telemetry acquisition as an online resource-allocation decision. TELEOS defines Telemetry Value of Information (TVoI) from expected entropy reduction, expected reduction in diagnostic decision loss, provenance/reconstruction value, operational consequence, and penalties for collection cost, privacy exposure, latency, security sensitivity, and retention burden. A four-level escalation policy selects fidelity under hard governance constraints and explicitly excludes inaccessible proprietary chain-of-thought. A provenance graph links user intent, planner decisions, model calls, retrieval, memory, authorization, tools, infrastructure, verifiers, and outcomes. To test the decision rule without fabricating production evidence, we implemented a controlled synthetic fault-diagnosis experiment with 14 latent states, 16 telemetry channels, 20 random seeds, and 800 episodes per seed for each of five policies. TELEOS achieved 72.27% fault-localization accuracy (95% CI 71.49–73.05%) at mean normalized telemetry cost 0.492, versus 81.04% at cost 1.910 for static-maximal collection. Against threshold adaptation, TELEOS improved accuracy by 8.16 percentage points while reducing cost by 8.38%. The results establish proof-of-concept for value-aware telemetry selection, while real-agent validation remains future work.

Keywords: Agentic AI; Adaptive telemetry; AI observability; Value of information; Causal provenance; Privacy-aware logging

1 INTRODUCTION

Large-language-model agents increasingly combine model inference with planning, retrieval, memory, external tools, authorization checks, and multi-step interaction. AgentOps work has therefore broadened observability beyond model latency and token accounting toward lifecycle artifacts and agent behavior [1]. OpenTelemetry's 2025 agent-observability work likewise recognized that an agent is not a single model call but a workflow whose tasks, actions, framework events, and feedback need interoperable telemetry [2,3]. These developments improve the data plane, but they leave a control-plane question unresolved: which telemetry is worth collecting now, at what fidelity, and for how long?

The question matters because agentic systems create a tension that conventional 'collect more' observability handles poorly. Dense traces increase storage and network load, may add latency on critical paths, and can expose prompts, retrieved records, memory contents, credentials, tool parameters, or user information. Conversely, sparse instrumentation can make an incident unreconstructable. Recent forensic work makes the same point in a different domain: logging quality depends on collecting causally useful information, not merely increasing volume [4]. Causal network telemetry similarly argues that visibility is determined by what information and causal relationships are captured [5].

TELEOS addresses this tension by treating telemetry as an active diagnostic action. At each decision point, the system maintains uncertainty over possible operational states or root causes and estimates the expected value of candidate telemetry. It escalates only when expected diagnostic or assurance benefit exceeds the marginal resource and exposure burden. The proposal is compatible with ordinary tracing systems—Dapper-style request tracing, X-Trace, dynamic causal monitoring such as Pivot Tracing, end-to-end systems such as Canopy, and workflow-centric tracing—but adds a policy layer that decides what to observe rather than assuming a fixed observation plan [6–10].

The manuscript makes four contributions. First, it formalizes Telemetry Value of Information (TVoI), combining information gain and decision value with consequence-sensitive cost, privacy, latency, security, and retention terms. Second, it specifies an adaptive escalation controller with hysteresis and hard governance constraints. Third, it defines an agent-specific provenance model that distinguishes recorded execution lineage from stronger causal-effect claims. Fourth, it provides a reproducible controlled simulation comparing static-minimal, static-maximal, random, threshold-adaptive, and TELEOS policies. Because no production agent deployment was executed for this paper, all numerical results are explicitly simulation results.

2 RELATED WORK AND BACKGROUND

Agentic AI complicates observability because execution is sequential and partially observable. ReAct interleaves reasoning-visible-to-the-orchestrator and action, Toolformer learns API use, and AutoGen composes interacting agents [11–13]. Benchmarks such as AgentBench, WebShop, ALFWorld, WebArena, BrowserGym, WorkArena, τ -bench, τ^2 -Bench and AgentBoard expose long-horizon failures, inconsistent tool use, rule-following errors, and environment-state mistakes [14–22]. Safety-oriented environments such as ToolEmu, AgentDojo, and AgentHarm further show that tool-capable agents introduce failures whose consequences cannot be inferred from final text alone [23–25]. These works motivate trajectory-level telemetry but do not solve telemetry selection under cost and privacy constraints.

Evaluation frameworks provide signals that can serve as TELEOS triggers or verifier outputs. HELM emphasizes multi-metric evaluation; G-Eval and LLM-as-a-judge methods support model-mediated evaluation but introduce evaluator bias; RAGAS addresses retrieval-augmented generation; SelfCheckGPT targets black-box hallucination; and TruthfulQA probes factuality [26–31]. Retrieval-augmented generation itself introduces a distinct provenance boundary between model output and retrieved evidence [32]. TELEOS treats such evaluators as optional outcome or uncertainty signals, not as privileged ground truth.

Distributed-systems diagnosis provides complementary techniques. MicroRCA, MicroRank, CloudRanger, Sage, and causal-discovery approaches rank root causes from metrics, traces, topology or learned causal structure [33–37]. Mystery Machine, Magpie, Pinpoint and request-flow comparison show the longstanding diagnostic value of request context and cross-component correlation [38–41]. Log-mining systems such as DeepLog, Drain, LogAnomaly, robust log anomaly detection, experience-based log analysis, console-log mining and LogBERT extract patterns from existing logs [42–48]. TELEOS differs in a key respect: it decides which observations to acquire before and during diagnosis rather than operating only on a fixed corpus.

The provenance and causality literature also shapes TELEOS. Causal reasoning can rank provenance by responsibility [49], while database provenance formalizes why/where lineage and its general abstractions [50,51]. Structural causal models distinguish association, intervention and counterfactual reasoning [52–55]. TELEOS deliberately uses the

phrase causal/provenance telemetry conservatively: execution edges such as 'invoked', 'read-from', 'authorized-by' and 'verified-by' are recorded lineage. Claims about intervention effects require additional causal assumptions and are not inferred merely because two spans are connected.

Finally, TELEOS draws on information theory and sequential decision making. Shannon entropy and mutual information provide a natural measure of uncertainty reduction [56,57]; Lindley and Bayesian experimental design formalize the expected information supplied by an experiment [58,59]. A telemetry controller is naturally partially observable [60] and can be cast as a constrained Markov decision process [61], while contextual-bandit and active-learning methods provide practical approximations for sequential acquisition [62–64]. Privacy must also be explicit: differential privacy formalizes disclosure control [65,66], and NIST guidance on logging, privacy, AI risk, and generative-AI risk emphasizes governance, minimization, access control and documentation [67–70].

Table 1: Conventional versus TELEOS-style telemetry.

Dimension	Conventional/static observability	TELEOS adaptive observability
Selection rule	Configured once; collect fixed spans/logs/metrics	Re-evaluated at each decision point using expected value and budgets
Fidelity	Usually fixed or sampled	Multi-level; can escalate, redact, aggregate or de-escalate
Diagnostic objective	Implicit	Explicit reduction in uncertainty and decision loss
Risk awareness	Often external to telemetry	Risk/consequence multiplies the benefit of diagnostic evidence
Privacy	Filter or redact after capture, depending on stack	Privacy exposure enters optimization and hard allow/deny policy
Causality	Trace-parent correlation or post-hoc graph	Agent-specific provenance graph with cautious causal interpretation
Retention	Global retention tiers	Per-item retention/summarization decision
Meta-telemetry	Rare	Records what was collected, omitted, and why

Table 2: Novelty matrix against closest related work.

Work/system	Primary contribution	Adaptive telemetry selection?	Agent-specific?	Privacy/value objective?
AgentOps [1]	Agent observability taxonomy	No	Yes	No
OpenTelemetry agent work [2,3]	Interoperable semantic conventions	Configurable instrumentation, not value optimization	Yes	No
Logging sufficiency [4]	Causality-aware forensic logging	No online value policy	No	Partial
Causal Network Telemetry [5]	Causal relations in network telemetry	No	No	No
Pivot Tracing [8]	Dynamic cross-component queries	Dynamic queries, not TVoI	No	No
Canopy [9]	End-to-end causal performance tracing	No	No	No
MicroRCA/MicroRank [33,34]	Root-cause ranking from existing telemetry	No	No	No

ToolEmu/AgentDojo [23,24]	Agent-risk environments	test	No	Yes	Safety evaluation
TELEOS (this work)	Decision-theoretic telemetry acquisition + provenance		Yes	Yes	Yes

2.1 Research gap and contributions

The reviewed literature contains mature tracing, log analysis, root-cause analysis, AI-agent evaluation, and emerging agent semantic conventions, but we found no work that jointly defines (i) per-observation value for agent telemetry, (ii) multi-budget acquisition, (iii) consequence-aware escalation, (iv) privacy-sensitive fidelity, and (v) provenance-aware diagnostic utility. TELEOS is therefore not another tracing schema. It is a policy and optimization layer above instrumentation standards.

The core hypothesis is that the best diagnostic telemetry set is episode-dependent. A retrieval-corruption episode may make retrieval content highly informative but an infrastructure-degradation episode may make that same content costly noise. Conversely, a policy-denial event may justify complete authorization provenance while still prohibiting raw user content. TELEOS operationalizes these conditional values and makes the trade-off inspectable through meta-telemetry.

2.2 TELEOS conceptual architecture

Figure 1 separates the agent data plane from the TELEOS control plane. Hooks in the agent runtime expose candidate signals from models, prompts/responses, planner state, tools, retrieval, memory, inter-agent messages, policy engines, applications, infrastructure, security controls, user feedback, and verifiers. The control plane estimates current uncertainty, operational consequence and anomaly evidence, evaluates candidate telemetry, and passes selected data through a policy/redaction gate. Selection decisions are themselves logged so that an investigator can distinguish 'not observed' from 'observed and normal'.

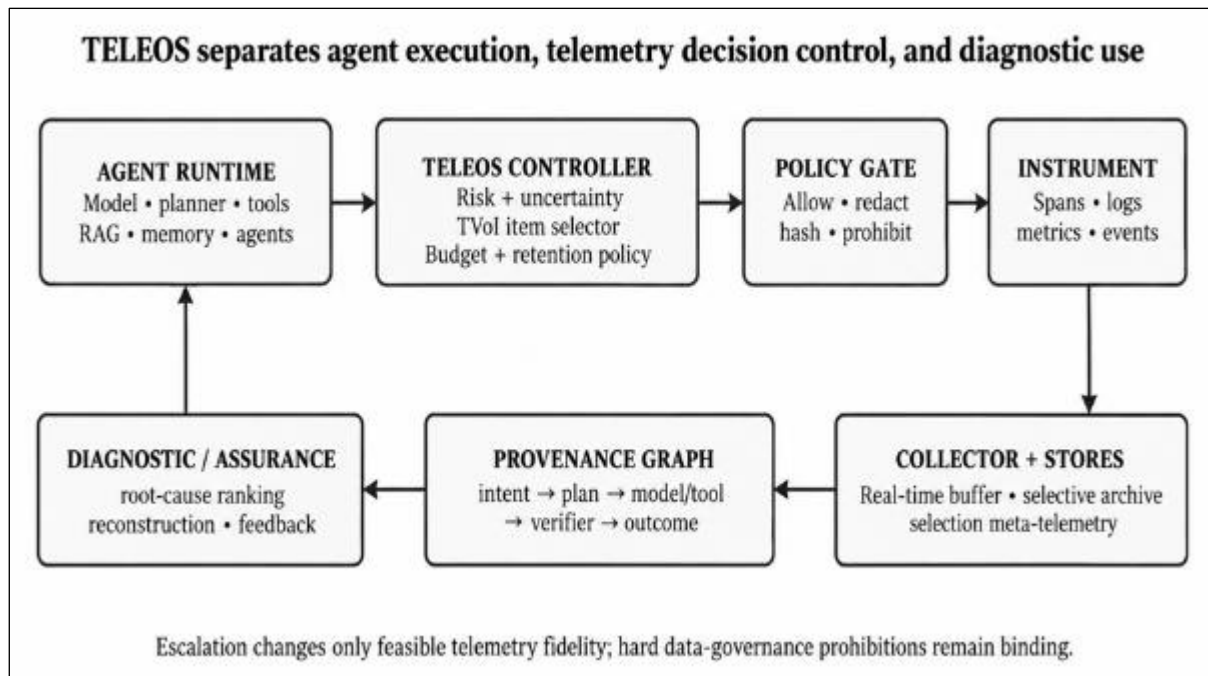


Figure 1: TELEOS end-to-end architecture. The decision layer changes telemetry fidelity while a policy gate enforces non-negotiable data-governance constraints.

Table 3: Agentic telemetry taxonomy and recommended treatment.

Telemetry family	Examples	Default TELEOS treatment
Model	model/version, latency, token counts, error status	Low-fidelity by default; escalate on model anomaly
Prompt/response	input/output text, structured messages	Sensitive; redact/hash/sample unless content is required and permitted
Planning artifacts	explicit plan nodes, action candidates, confidence	Capture only artifacts exposed by the application; never assume hidden chain-of-thought access
Agent state	task state, checkpoints, loop counters	Structured state hashes/IDs at low fidelity; snapshots on escalation
Tool invocation	tool name, parameters, status, result	High value near consequential actions; redact secrets
Retrieval/RAG	query, document IDs, rank, chunks	Metadata first; content only when value exceeds privacy cost
Memory	read/write keys, version, snapshot	Metadata by default; content under stricter policy
Inter-agent communication	sender/receiver IDs, message type, delivery	Capture lineage and delivery; content policy-dependent
Authorization/policy	decision, rule ID, subject/resource scope	Preserve provenance; high assurance value
Application/distributed trace	request/span IDs, events, errors	Use standards-compatible structured spans
Infrastructure/security	CPU, network, dependencies, alerts	Escalate when system symptoms implicate environment
User/outcome	feedback, verifier, success/failure	Retain compactly because it anchors diagnosis and learning

2.3 Formal mathematical model

Let Θ_t be a latent diagnostic state at decision time t , such as normal operation or one of several root-cause classes. The controller maintains belief $b_t(\theta) = P(\Theta_t = \theta \mid h_t)$, where h_t is the observable history. A telemetry action $a = (z, f, r)$ chooses modality z , fidelity f and retention/redaction treatment r , producing observation Y_a . The first value term is expected information gain (EIG):

$$\text{EIG}(a \mid b_t) = H(b_t) - E_Y[H(b_{t+1})] = I(\Theta_t; Y_a \mid h_t, a) \quad (1)$$

Information gain alone can overvalue signals that reduce uncertainty without changing an operational decision. TELEOS therefore also uses expected value of sample information (EVSI). If d is a diagnostic or assurance decision with loss $L(d, \theta)$, then:

$$\text{EVSI}(a \mid b_t) = \min_d E_b[L(d, \theta)] - E_Y[\min_d E[L(d, \theta) \mid Y_a]] \quad (2)$$

We add $Q(a, G_t)$, the expected contribution to provenance completeness and post-incident reconstruction, and multiply the benefit by consequence factor $\kappa_t \geq 1$. Each action incurs normalized collection cost C , privacy exposure P , latency L , security sensitivity S , storage/retention burden R , and network burden N . A practical one-step score is:

$$\text{TVol}_t(a) = \kappa_t [\alpha \cdot \text{EIG} + \beta \cdot \text{EVSI} + \gamma \cdot Q] - \lambda_C C - \lambda_P P - \lambda_L L - \lambda_S S - \lambda_R R - \lambda_N N \quad (3)$$

Hard constraints are not converted into soft penalties. If policy prohibits a field, a is infeasible regardless of score. For feasible actions A_t , a myopic controller solves a multi-budget selection problem; a full sequential formulation is a constrained POMDP/CMDP:

$$\max_{\pi} E_{\pi}[\sum_t \text{TVol}_t(a_t)] \text{ s.t. } \Sigma C \leq B_C, \Sigma P \leq B_P, \Sigma L \leq B_L, \Sigma R \leq B_R, a_t \in A_{\text{allowed}}(s_t) \quad (4)$$

The coefficients are interpretable shadow prices rather than universal constants. In deployment they should be calibrated from incident costs, data-classification policy, service-level objectives, and retrospective diagnosis outcomes. TELEOS is therefore a framework for making collection decisions explicit, not a claim that one set of weights is globally optimal.

2.4 Adaptive telemetry policy

The runtime controller separates escalation from item selection. The escalation score Φ_t combines normalized posterior entropy, estimated action consequence, anomaly evidence, and policy-risk indicators. Two thresholds implement hysteresis so that telemetry does not oscillate rapidly between modes:

$$\Phi_t = \eta_H \cdot H(b_t) / H_{\max} + \eta_R \cdot \text{Risk}_t + \eta_A \cdot \text{Anomaly}_t + \eta_C \cdot \text{Consequence}_t \quad (5)$$

TELEOS defines four operational modes: L0 minimal, L1 diagnostic, L2 provenance, and L3 forensic. Crossing an upper threshold permits higher-fidelity candidates; falling below a lower threshold, reaching diagnostic confidence, or completing the protected action de-escalates collection. Importantly, escalation expands only the allowed candidate set. It never overrides a hard prohibition on sensitive content.

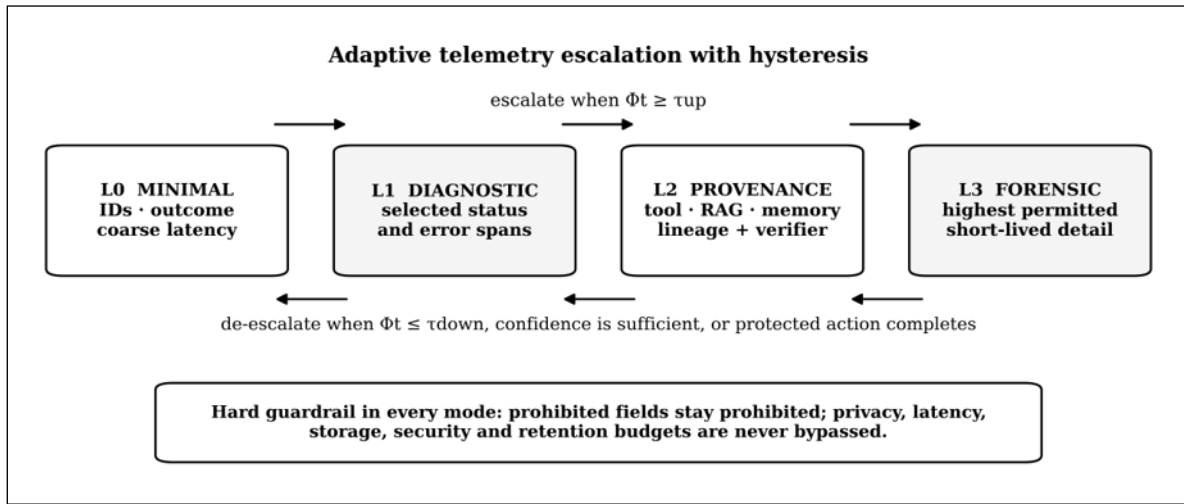


Figure 2: Adaptive telemetry escalation state machine with hysteresis and hard governance guardrails.

Within a mode, candidate telemetry can be selected greedily by TVoI, by a contextual bandit trained on past incidents, or by approximate POMDP planning. The controlled experiment below uses a transparent greedy policy so every decision can be reproduced and audited. Production implementations can learn the value model but should preserve deterministic guardrails around prohibited data and high-consequence actions.

2.5 Causal and provenance telemetry model

TELEOS represents an episode as a directed provenance graph $G=(V,E)$. Node types include user intent, planner decisions, model invocations, retrievals, memory reads/writes, tool selection, policy decisions, tool execution, infrastructure responses, verifier events, and final outcomes. Edges are typed execution relations: spawned-by, read-from, invoked, authorized-by, returned-to, derived-from, and verified-by. A root-cause ranker can use the graph to restrict hypotheses to ancestors of a failed verifier or to score nodes whose local observations are unlikely under normal operation.

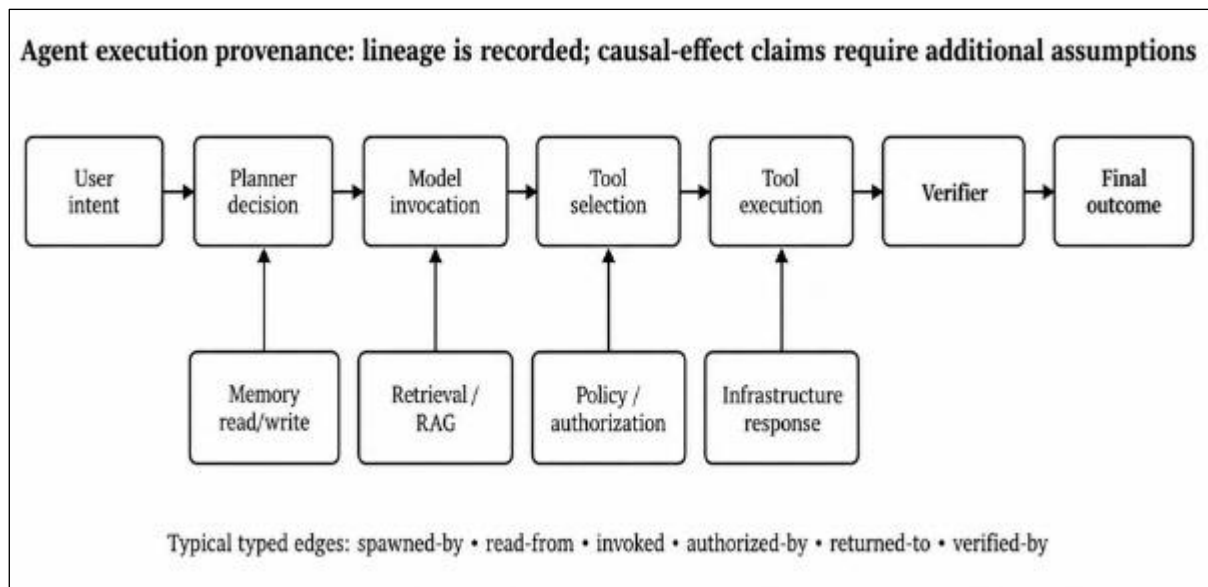


Figure 3: Agent-specific provenance graph. Recorded lineage is useful for diagnosis but is not automatically equivalent to an identified causal effect.

This distinction prevents a common overclaim. Trace parentage is causal in the engineering sense of execution lineage, but causal attribution in the Pearl/Granger sense requires assumptions about confounding, interventions, stationarity, or structural equations [52–55]. TELEOS therefore reports 'provenance-supported attribution' unless those assumptions are independently justified. Future deployments may layer structural causal models over the recorded graph.

3 EXPERIMENTAL METHODOLOGY

Sections 8–10 constitute the Materials and Methods for the controlled computational study. We deliberately used a synthetic probabilistic testbed rather than claiming unavailable production-agent measurements. The experiment isolates the telemetry-selection problem: a known latent fault generates stochastic observable signatures across candidate telemetry channels, and each policy chooses which channels to acquire before predicting the fault. This design provides ground truth, controllable budgets, repeated random seeds, and exact reproducibility.

The simulator was implemented in Python with NumPy. Fourteen latent classes were used: normal operation plus 13 fault classes. Sixteen binary telemetry channels were assigned class-conditional observation probabilities and normalized cost, privacy, latency, and security burdens. The prior probability of normal operation was 0.35; the remaining probability mass was distributed uniformly across fault classes. After each collected item, an exact Naive Bayes posterior was computed from the synthetic likelihood model. Posterior entropy measured residual uncertainty.

For each of 20 independent pseudo-random seeds, 800 episodes were generated and evaluated under each policy. Thus each policy was evaluated on 16,000 episodes, for 80,000 policy-episode evaluations. The same latent episodes and observations were reused across policies within each seed to support paired comparisons. Mean performance and 95% confidence intervals were calculated from seed-level means using the normal approximation; paired t-tests used the 20 paired seed means. We also computed non-parametric paired bootstrap confidence intervals for policy differences as a robustness check.

3.1 Datasets, environments and fault injection

The current evidence is a controlled simulation, not a benchmark run on ALFWorld, WebShop, BrowserGym or a Kubernetes cluster. Those environments are proposed for external validation because they can expose different segments of the TELEOS taxonomy [15–20]. The synthetic model includes the fault classes listed in Table 4. Their signatures were intentionally overlapping rather than deterministic, preventing a single channel from trivially identifying every fault.

Table 4: Controlled fault model and high-information telemetry channels.

Fault class	Illustrative operational meaning	Higher-likelihood diagnostic channels
Tool timeout	Tool exceeds deadline	aggregate latency; tool I/O; dependency trace
Wrong tool	Planner selects inappropriate capability	tool choice; planner state
Retrieval corruption	Retrieved evidence is wrong/noisy	retrieval metadata/content; verifier
Stale memory	Old agent state influences action	memory metadata/snapshot
Context truncation	Relevant context omitted	prompt/response artifact; planner state
Model failure	Model endpoint/status failure	model status; latency
API failure	External API failure	tool I/O; dependency trace
Authorization denial	Policy blocks action	authorization/policy event
Planner loop	No-progress repeated planning	planner state; latency
Bad tool parameter	Hallucinated/malformed argument	tool I/O; planner state
Dependency failure	Downstream service fault	dependency trace; infrastructure metrics
Message loss	Inter-agent delivery failure	message-delivery event
Infrastructure degradation	Resource/network degradation	infrastructure metrics; latency

3.2 Baselines

Table 5: Telemetry policies evaluated in the controlled experiment.

Policy	Acquisition rule
Static minimal	Always collect episode outcome, verifier signal and aggregate latency.
Static maximal	Collect all 16 channels.
Random/ad hoc	Start with minimal set; randomly add feasible channels under cost and privacy budgets.
Threshold adaptive	Start minimal; use fixed ordered escalation while confidence is low or failure signals are present.
TELEOS	Start with outcome; greedily acquire the feasible channel maximizing consequence-weighted information gain minus cost/privacy/latency/security penalties; stop at confidence ≥ 0.84 or non-positive marginal value.

The TELEOS simulation used cost budget 0.78 and privacy budget 0.52. For candidate j , the implemented score was $EIG_j(1+0.65 \cdot \text{expected severity}) - 0.30C_j - 0.38P_j - 0.12L_j - 0.10S_j$. This is a restricted instantiation of Equation 3: EVSI and graph reconstruction terms were not separately parameterized in the main simulation. The simplification is intentional and is addressed as a limitation.

4 RESULTS

Static-maximal collection produced the highest diagnostic accuracy, 0.8104, but consumed normalized cost 1.910 and privacy exposure 1.680 per episode. TELEOS reached 0.7227 accuracy at cost 0.492 and privacy exposure 0.269. Relative to static maximal, this is a 74.2% reduction in collection cost, an 84.0% reduction in privacy exposure, and a 65.5% reduction in telemetry-item count, with an 8.78-percentage-point loss in accuracy. The result does not show that maximal telemetry is unnecessary; it shows that under this synthetic model a large fraction of maximal telemetry had low marginal diagnostic value.

The more relevant comparison is against adaptive and ad hoc baselines at similar budget. TELEOS exceeded threshold adaptation by 8.16 percentage points in accuracy while reducing cost by 8.38% and reducing the mean number of

acquired items by 0.84. Against random acquisition, TELEOS improved accuracy by 11.62 percentage points while reducing cost by 33.5% and privacy exposure by 40.9%. These paired comparisons support the value-aware selection mechanism under the assumed likelihood model.

Table 6: Main controlled-simulation results; values are mean (95% CI) across 20 seed means.

Policy	Accuracy	Cost	Privacy	Items	Steps to explanation
Static minimal	0.425 (0.416–0.434)	0.160 (0.160–0.160)	0.020 (0.020–0.020)	3.000 (3.000–3.000)	3.297 (3.282–3.311)
Random	0.607 (0.599–0.614)	0.740 (0.739–0.741)	0.455 (0.453–0.456)	8.147 (8.133–8.161)	6.608 (6.554–6.663)
Threshold	0.641 (0.634–0.648)	0.537 (0.533–0.541)	0.276 (0.273–0.278)	6.366 (6.333–6.399)	6.538 (6.488–6.588)
TELEOS	0.723 (0.715–0.730)	0.492 (0.488–0.496)	0.269 (0.266–0.272)	5.528 (5.488–5.568)	6.125 (6.078–6.171)
Static maximal	0.810 (0.804–0.817)	1.910 (1.910–1.910)	1.680 (1.680–1.680)	16.000 (16.000–16.000)	11.173 (11.080–11.266)

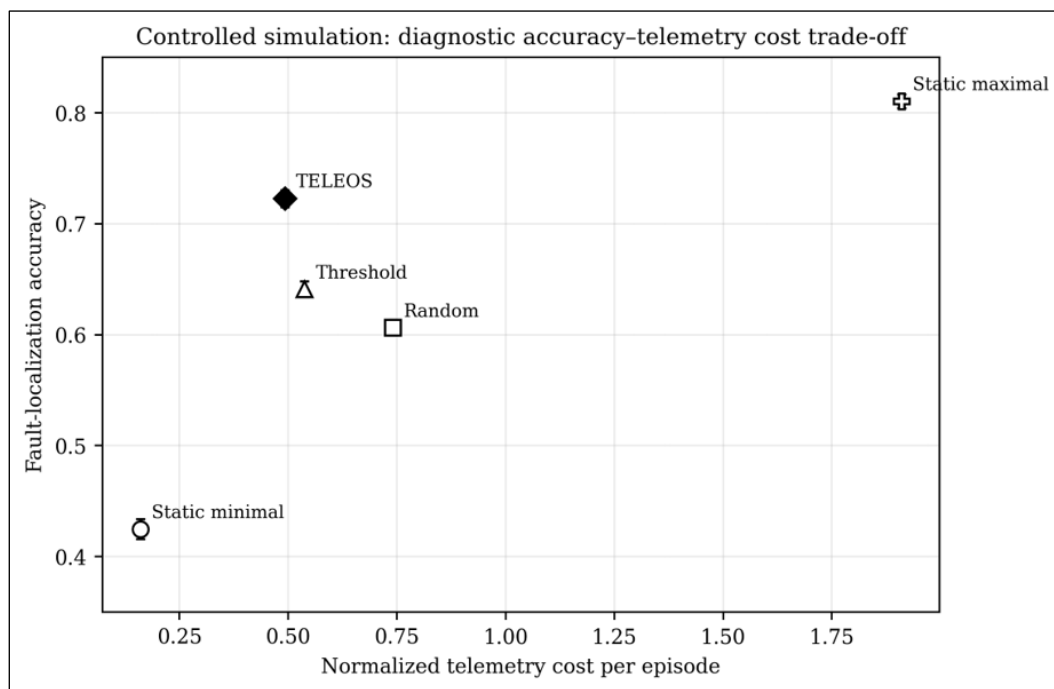


Figure 4: Fault-localization accuracy versus normalized telemetry cost in the controlled simulation. Error bars show 95% confidence intervals across seed-level means.

4.1 Statistical analysis

Paired seed-level comparisons were used because each policy evaluated the same generated episodes for a given seed. TELEOS versus threshold adaptation had a mean accuracy difference of +0.0816 (paired bootstrap 95% CI +0.0747 to +0.0888; paired $t_{19}=22.53$, $p<0.001$). Mean cost difference was -0.0450 (bootstrap 95% CI -0.0470 to -0.0431 ; $p<0.001$), and privacy difference was -0.00708 (95% CI -0.00837 to -0.00582 ; $p<0.001$). TELEOS versus random acquisition had accuracy difference +0.1162 (95% CI +0.1081 to +0.1238) and cost difference -0.2481 (95% CI -0.2524 to -0.2434), both $p<0.001$.

We avoid interpreting the very large standardized paired effect sizes from this simulation as real-world magnitudes. The synthetic generator is fixed, seeds are numerous, and the value model is aligned with the same likelihood family used for diagnosis; consequently, between-seed variance is small. Confidence intervals therefore quantify Monte Carlo uncertainty within this testbed, not external validity across models, tasks, organizations or incident distributions.

4.2 Ablation and sensitivity analysis

Table 7: Ablation results from a separate 10-seed × 400-episode simulation.

Variant	Accuracy	Cost	Privacy	Items	Final entropy
TELEOS	0.728	0.490	0.268	5.51	1.343
No stage-specific provenance proxy	0.489	0.536	0.197	7.25	2.318
No privacy penalty	0.719	0.497	0.290	5.48	1.366
No risk weighting	0.727	0.482	0.258	5.43	1.348

The 'no provenance' ablation deliberately coarsened stage-specific telemetry identities, making observations less attributable to the component that generated them. Accuracy fell from 0.728 to 0.490 while more telemetry items were acquired. This is a proxy for loss of provenance resolution, not evidence that a particular causal-discovery algorithm causes the improvement. Removing the privacy penalty increased exposure from 0.268 to 0.290 without improving accuracy. Removing risk weighting produced little accuracy change in this testbed, suggesting that information gain dominated when fault severities were only moderately differentiated.

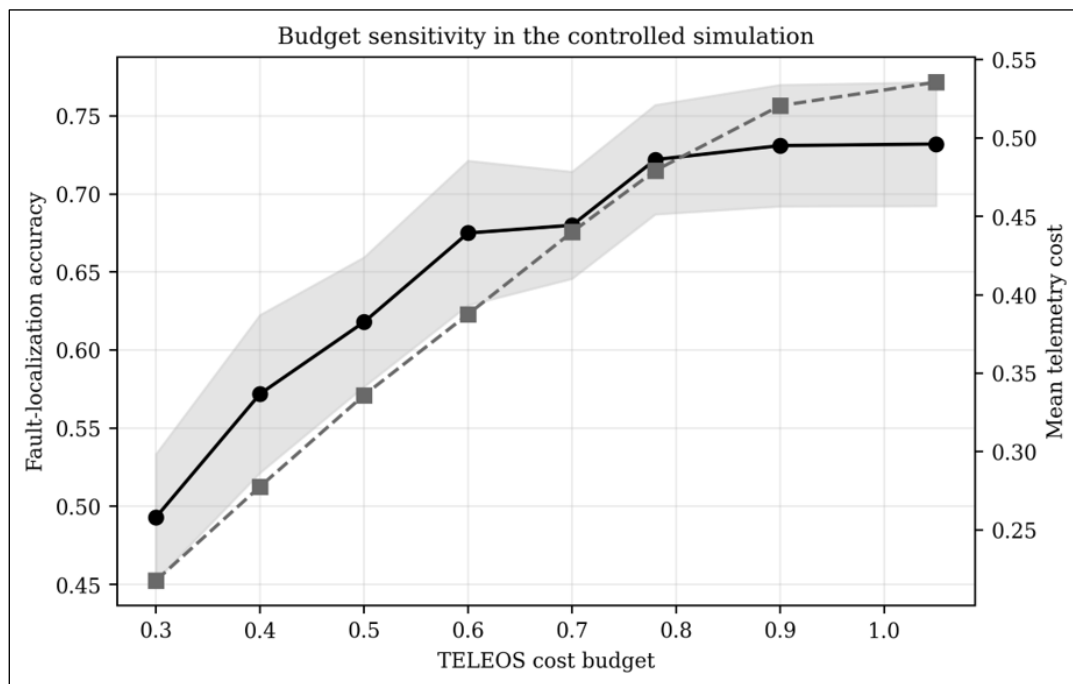


Figure 5: TELEOS budget sensitivity. Accuracy rose with the telemetry budget and showed diminishing returns beyond approximately 0.9 normalized cost in this synthetic model.

Budget sensitivity showed the expected frontier: accuracy rose from 0.493 at a cost budget of 0.30 to 0.722 at 0.78 and 0.732 at 1.05. The curve flattened after roughly 0.9, indicating diminishing marginal diagnostic value under the synthetic channel model. The location of this knee is testbed-specific and should not be transferred to production deployments.

5 DISCUSSION

The controlled results support the central TELEOS claim: observability can be treated as a decision problem in which information is acquired selectively rather than as a binary choice between sparse and exhaustive logging. In the simulation, TELEOS occupied a useful part of the accuracy–cost frontier because it acquired channels whose likelihoods were discriminative for the current belief state and avoided channels with low marginal information or high exposure.

This framing changes the role of observability engineering. A semantic convention still matters because candidate telemetry must be consistently named, typed and correlated [2,3]. However, semantic standardization is the substrate; it does not tell the runtime whether raw retrieval content is necessary for a specific episode. Similarly, root-cause algorithms still matter, but a perfect ranker cannot recover evidence that was never captured. TELEOS makes evidence acquisition and diagnosis part of one closed loop.

The method also provides a principled way to discuss privacy. The model does not equate privacy with a single scalar in deployment; the scalar penalty is only one component. Hard data-classification constraints remain first-class, and prohibited fields stay prohibited even in forensic mode. Thus 'forensic escalation' means the highest permitted fidelity, not unconditional collection.

5.1 Practical implementation architecture

A practical implementation can sit above OpenTelemetry-compatible instrumentation. Agent-framework hooks emit candidate events with stable identifiers, sensitivity classes, estimated byte/latency costs, and parent provenance IDs. A TELEOS controller maintains the belief state and budgets, returns a sampling/fidelity decision, and writes a compact selection-decision event. An instrumentation adapter then emits the chosen attributes and content through the existing collector pipeline. The architecture is compatible with head-, tail-, or hybrid sampling, but the decision variable is richer than 'keep/drop': it can choose metadata-only, structured summary, redacted content, full permitted payload, or retention duration.

For high-consequence tool calls, the controller should be placed before the action boundary so provenance can be escalated before an irreversible side effect. Examples include authorization result, tool schema/version, parameter hashes, selected unredacted fields when policy permits, dependency identifiers, and verifier requirements. The proposing model need not and should not expose hidden chain-of-thought. TELEOS operates on observable execution artifacts, explicit planner state created by the application, model I/O that policy allows, tool calls, memory interfaces and outcome signals.

5.2 Security and privacy implications

Adaptive telemetry can reduce exposure, but the controller itself becomes a security-sensitive component. Attackers might deliberately trigger forensic mode to exfiltrate sensitive prompts, or suppress anomaly signals to prevent escalation. Countermeasures include signed policy bundles, separate authorization for sensitive telemetry, rate limits on forensic activation, immutable meta-telemetry, collector-side redaction, envelope encryption, retention TTLs, and independent monitoring of the controller. Differential-privacy techniques may protect aggregate telemetry, but they are not a substitute for access control when exact event reconstruction is required [65,66].

NIST log-management and privacy guidance supports explicit log governance and minimization [67,68]. AI RMF and its generative-AI profile further motivate risk measurement, documentation and controls tailored to generative systems [69,70]. TELEOS operationalizes these principles at acquisition time: governance constraints are checked before collection, and the reason for collecting sensitive telemetry can be recorded as an auditable decision.

5.3 Threats to validity

Table 8: Principal threats to validity and mitigation.

Threat	Implication	Mitigation / interpretation
Synthetic likelihood model	TELEOS and Bayesian diagnosis share the same modeled evidence family	Report as controlled proof-of-concept; require external benchmark validation
Simplified binary channels	Real telemetry is structured, continuous and high-dimensional	Use learned likelihood/value estimators in future work
Fixed fault prior	Production incident distribution may drift	Calibrate priors online; stress-test rare faults
Scalar privacy cost	Legal/data-governance constraints are not fully scalar	Retain hard allow/deny policies outside optimization
No live LLM/framework overhead	Wall-clock and collector effects not measured	Benchmark in LangGraph/AutoGen/BrowserGym/Kubernetes deployments
Naive Bayes independence	Telemetry channels may be correlated	Use graphical/learned posterior models where justified
Provenance proxy ablation	Does not establish causal-discovery benefit	Interpret only as value of stage-specific identity

5.4 Limitations

The strongest limitation is external validity. The present study validates the telemetry-selection mechanism under a designed probabilistic environment; it does not demonstrate performance on a production LLM agent, Kubernetes service mesh, or public agent benchmark. The TVoI weights and budgets are illustrative. Real deployments must estimate data sensitivity, cost, and diagnostic utility from their own incident histories and governance regimes.

Second, the main experiment uses information gain rather than the full EVSI and provenance-reconstruction terms in Equation 3. Third, uncertainty from proprietary LLMs may be poorly calibrated or unavailable; TELEOS therefore should combine model-independent signals such as verifier failures, repeated actions, tool exceptions, policy denials and infrastructure anomalies. Fourth, adaptive collection can make incident datasets missing-not-at-random. Selection decisions must therefore be retained so downstream analysis can account for the acquisition policy.

5.5 Future research

The next validation step is a preregistered experiment in at least two open agent environments and one distributed-systems testbed. A practical design would instrument BrowserGym/WorkArena and WebShop with OpenTelemetry-compatible spans, inject retrieval, memory, tool, authorization and infrastructure faults, and repeat the five policies under matched trajectories. A Kubernetes microservice application could separately test dependency and infrastructure faults. Primary outcomes should include root-cause top-k accuracy, wall-clock mean time to explanation, bytes exported, CPU overhead, privacy-class exposure, and outcome-reconstruction accuracy.

Methodologically, TVoI can be learned from historical incidents using contextual bandits, with off-policy evaluation to reduce deployment risk. Bayesian experimental design can provide stronger multi-step acquisition policies. Causal extensions should estimate intervention effects only where assumptions are testable. Another important direction is selective retention: once an episode is verified as benign, high-fidelity data could be summarized or cryptographically committed and then deleted, while incidents retain a policy-approved forensic subset.

6 CONCLUSION

TELEOS reframes observability for agentic AI from a passive data-collection problem into a constrained evidence-acquisition problem. Its Telemetry Value of Information links uncertainty reduction and diagnostic decision value to the real costs of collecting, exposing, transmitting and retaining telemetry. Adaptive escalation provides a controlled path from minimal monitoring to diagnostic, provenance and forensic modes, while hard governance rules remain inviolable. The controlled simulation shows that value-aware selection can substantially outperform random and threshold policies at comparable or lower telemetry burden, and can approach the diagnostic benefit of exhaustive telemetry at a fraction of its synthetic cost and privacy exposure. These results are proof-of-concept rather than production evidence. The central research agenda is therefore clear: standardize candidate telemetry, learn its context-dependent value, preserve provenance, and validate adaptive acquisition on reproducible real-agent and cloud-system fault benchmarks.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Acknowledgments

The author acknowledges the open research communities that maintain public agent benchmarks, OpenTelemetry specifications, and reproducible observability tooling. No external contributor is listed as an author solely for editorial or computational assistance.

Disclosure of conflict of interest

The author declares no conflict of interest.

Statement of ethical approval

This study used a synthetic computational fault model and did not involve human participants, animals, clinical data, or personally identifiable information. Institutional ethics approval was therefore not required for the reported experiment.

Data availability

The controlled experiment generates synthetic observations and contains no proprietary or human-subject data. The episode-level simulation output used for the reported analyses is supplied with the submission package as `teleos_sim_results.csv`.

Code availability

The Python scripts used to generate the controlled simulation, ablation analysis, and figures are available in the author working package and should be deposited in a persistent public repository before or at publication.

Author contributions

Prudvi Saisaran Ponduru: Conceptualization, methodology, formal analysis, software, investigation, validation, data curation, visualization, writing - original draft, writing - review and editing, and project administration. Pavani Priya Vyshnavi Nandanavanam: Methodology, investigation, validation, literature investigation, data interpretation, visualization, and writing - review and editing. Sai Kesav Kumar Ponduru: Methodology, formal analysis, statistical analysis, validation, data interpretation, and writing - review and editing. All authors reviewed and approved the final manuscript and accept responsibility for the integrity of the work.

REFERENCES

- [1] Dong L, Lu Q, Zhu L. AgentOps: Enabling Observability of LLM Agents. arXiv [Preprint]. 2024. arXiv:2411.05285.
- [2] Liu G, Solomon S. AI Agent Observability - Evolving Standards and Best Practices. OpenTelemetry; 2025 Mar 6.
- [3] OpenTelemetry. Semantic Conventions for Generative AI Agentic Systems (gen_ai.*). GitHub issue 35; 2025 Aug 20.
- [4] Olegård J, Axelsson S, Li Y. When is logging sufficient? Tracking event causality for improved forensic analysis and correlation. *Forensic Sci Int Digit Investig*. 2025;52(Suppl):301877. doi:10.1016/j.fsidi.2025.301877.
- [5] Liu Y, Foster N, Schneider FB. Causal Network Telemetry. In: P4 Workshop in Europe (EuroP4 '22). New York: ACM; 2022. p. 1-7. doi:10.1145/3565475.3569084.
- [6] Sigelman BH, Barroso LA, Burrows M, Stephenson P, Plakal M, Beaver D, et al. Dapper, a Large-Scale Distributed Systems Tracing Infrastructure. Google Research; 2010.
- [7] Fonseca R, Porter G, Katz RH, Shenker S, Stoica I. X-Trace: A Pervasive Network Tracing Framework. In: 4th USENIX Symposium on Networked Systems Design and Implementation. 2007. p. 271-284.
- [8] Mace J, Roelke R, Fonseca R. Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems. *ACM Trans Comput Syst*. 2018;35(4):11:1-11:28. doi:10.1145/3208104.
- [9] Kaldor J, Mace J, Bejda M, Gao E, Kuropatwa W, O'Neill J, et al. Canopy: An End-to-End Performance Tracing and Analysis System. In: Proceedings of the 26th Symposium on Operating Systems Principles. 2017. p. 34-50. doi:10.1145/3132747.3132749.
- [10] Sambasivan RR, Shafer I, Mace J, Ganger GR. Principled Workflow-Centric Tracing of Distributed Systems. In: ACM Symposium on Cloud Computing. 2016. doi:10.1145/2987550.2987568.
- [11] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan KR, Cao Y. ReAct: Synergizing Reasoning and Acting in Language Models. *International Conference on Learning Representations*; 2023.
- [12] Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Hambro E, et al. Toolformer: Language Models Can Teach Themselves to Use Tools. *Adv Neural Inf Process Syst*. 2023;36. doi:10.52202/075280-2997.
- [13] Wu Q, Bansal G, Zhang J, Wu Y, Li B, Zhu E, et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *Conference on Language Modeling*; 2024.
- [14] Liu X, Yu H, Zhang H, Xu Y, Lei X, Lai H, et al. AgentBench: Evaluating LLMs as Agents. *International Conference on Learning Representations*; 2024.
- [15] Yao S, Chen H, Yang J, Narasimhan K. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. *Adv Neural Inf Process Syst*. 2022;35. doi:10.52202/068431-1508.
- [16] Shridhar M, Yuan X, Côté MA, Bisk Y, Trischler A, Hausknecht M. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. *International Conference on Learning Representations*; 2021.

- [17] Zhou S, Xu FF, Zhu H, Zhou X, Lo R, Sridhar A, et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. *International Conference on Learning Representations*; 2024.
- [18] de Chezelles TLS, Gasse M, Lacoste A, Caccia M, Drouin A, Boisvert L, et al. The BrowserGym Ecosystem for Web Agent Research. *Trans Mach Learn Res*. 2025.
- [19] Drouin A, Gasse M, Caccia M, Laradji IH, Del Verme M, Marty T, et al. WorkArena: How Capable are Web Agents at Solving Common Knowledge Work Tasks? *Proc Mach Learn Res*. 2024;235:11642-11662.
- [20] Yao S, Shinn N, Razavi P, Narasimhan K. τ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *International Conference on Learning Representations*; 2025.
- [21] Barres V, Dong H, Ray S, Si X, Narasimhan K. τ^2 -Bench: Evaluating Conversational Agents in a Dual-Control Environment. *arXiv [Preprint]*. 2025. arXiv:2506.07982.
- [22] Ma C, Zhang J, Zhu Z, Yang C, Yang Y, Jin Y, et al. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. *Adv Neural Inf Process Syst*. 2024;37. doi:10.52202/079017-2365.
- [23] Ruan Y, Dong H, Wang A, Pitis S, Zhou Y, Ba J, et al. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. *International Conference on Learning Representations*; 2024.
- [24] Debenedetti E, Zhang J, Balunović M, Beurer-Kellner L, Fischer M, Tramèr F. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. *Adv Neural Inf Process Syst*. 2024;37. doi:10.52202/079017-2636.
- [25] Andriushchenko M, Souly A, Dziemian M, Duenas D, Lin M, Wang J, et al. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. *International Conference on Learning Representations*; 2025.
- [26] Bommasani R, Liang P, Lee T. Holistic Evaluation of Language Models. *Ann N Y Acad Sci*. 2023;1525(1):140-146. doi:10.1111/nyas.15007.
- [27] Liu Y, Iter D, Xu Y, Wang S, Xu R, Zhu C. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In: *Proceedings of EMNLP 2023*. 2023. p. 2511-2522. doi:10.18653/v1/2023.emnlp-main.153.
- [28] Es S, James J, Espinosa Anke L, Schockaert S. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In: *Proceedings of EACL 2024 System Demonstrations*. 2024. p. 150-158. doi:10.18653/v1/2024.eacl-demo.16.
- [29] Zheng L, Chiang WL, Sheng Y, Zhuang S, Wu Z, Zhuang Y, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Adv Neural Inf Process Syst*. 2023;36.
- [30] Manakul P, Liusie A, Gales M. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In: *Proceedings of EMNLP 2023*. 2023. p. 9004-9017. doi:10.18653/v1/2023.emnlp-main.557.
- [31] Lin S, Hilton J, Evans O. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2022. p. 3214-3252. doi:10.18653/v1/2022.acl-long.229.
- [32] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv Neural Inf Process Syst*. 2020;33:9459-9474.
- [33] Li W, Tordsson J, Elmroth E, Kao O. MicroRCA: Root Cause Localization of Performance Issues in Microservices. In: *IEEE/IFIP Network Operations and Management Symposium*. 2020. doi:10.1109/NOMS47738.2020.9110353.
- [34] Yu G, et al. MicroRank: End-to-End Latency Issue Localization with Extended Spectrum Analysis in Microservice Environments. In: *Proceedings of The Web Conference 2021*. 2021. p. 3087-3098. doi:10.1145/3442381.3449905.
- [35] Wang P, et al. CloudRanger: Root Cause Identification for Cloud Native Systems. In: *18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 2018. p. 492-502. doi:10.1109/CCGRID.2018.00076.
- [36] Gan Y, Liang M, Dev S, Lo D, Delimitrou C. Sage: Practical and Scalable ML-Driven Performance Debugging in Microservices. In: *Proceedings of ASPLOS*. 2021. p. 135-151. doi:10.1145/3445814.3446700.
- [37] Ikram A, et al. Root Cause Analysis of Failures in Microservices through Causal Discovery. *Adv Neural Inf Process Syst*. 2022;35. doi:10.52202/068431-2259.
- [38] Chow M, Meisner D, Flinn J, Peek D, Wenisch TF. The Mystery Machine: End-to-End Performance Analysis of Large-Scale Internet Services. In: *11th USENIX Symposium on Operating Systems Design and Implementation*. 2014.

- [39] Barham P, Isaacs R, Mortier R, Narayanan D. Magpie: Online Modelling and Performance-Aware Systems. In: 9th Workshop on Hot Topics in Operating Systems. 2003.
- [40] Chen MY, Kiciman E, Fratkin E, Fox A, Brewer EA. Pinpoint: Problem Determination in Large, Dynamic Internet Services. In: International Conference on Dependable Systems and Networks. 2002. p. 595-604. doi:10.1109/DSN.2002.1029005.
- [41] Sambasivan RR, Zheng AX, De Rosa M, Krevat E, Whitman S, Stroucken M, et al. Diagnosing Performance Changes by Comparing Request Flows. In: 8th USENIX Symposium on Networked Systems Design and Implementation. 2011.
- [42] Du M, Li F, Zheng G, Srikumar V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In: ACM Conference on Computer and Communications Security. 2017. p. 1285-1298. doi:10.1145/3133956.3134015.
- [43] He P, Zhu J, Zheng Z, Lyu MR. Drain: An Online Log Parsing Approach with Fixed Depth Tree. In: IEEE International Conference on Web Services. 2017. p. 33-40. doi:10.1109/ICWS.2017.13.
- [44] Meng W, Liu Y, Zhu Y, Zhang S, Pei D, Liu Y, et al. LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs. In: IJCAI. 2019. p. 4739-4745. doi:10.24963/ijcai.2019/658.
- [45] Zhang X, et al. Robust Log-Based Anomaly Detection on Unstable Log Data. In: ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2019. p. 807-817. doi:10.1145/3338906.3338931.
- [46] He S, Zhu J, He P, Lyu MR. Experience Report: System Log Analysis for Anomaly Detection. In: IEEE International Symposium on Software Reliability Engineering. 2016. p. 207-218. doi:10.1109/ISSRE.2016.21.
- [47] Xu W, Huang L, Fox A, Patterson DA, Jordan MI. Detecting Large-Scale System Problems by Mining Console Logs. In: Proceedings of the 22nd ACM Symposium on Operating Systems Principles. 2009. p. 117-132. doi:10.1145/1629575.1629587.
- [48] Guo H, Yuan S, Wu X. LogBERT: Log Anomaly Detection via BERT. In: International Joint Conference on Neural Networks. 2021. p. 1-8. doi:10.1109/IJCNN52387.2021.9534113.
- [49] Meliou A, Gatterbauer W, Suciu D. Bringing Provenance to Its Full Potential Using Causal Reasoning. In: 3rd USENIX Workshop on the Theory and Practice of Provenance. 2011.
- [50] Buneman P, Khanna S, Tan WC. Why and Where: A Characterization of Data Provenance. In: International Conference on Database Theory. LNCS 1973. Berlin: Springer; 2001. p. 316-330. doi:10.1007/3-540-44503-X_20.
- [51] Cheney J, Chiticariu L, Tan WC. Provenance in Databases: Why, How, and Where. Found Trends Databases. 2009;1(4):379-474. doi:10.1561/1900000006.
- [52] Pearl J. Causality: Models, Reasoning, and Inference. 2nd ed. Cambridge: Cambridge University Press; 2009.
- [53] Peters J, Janzing D, Schölkopf B. Elements of Causal Inference: Foundations and Learning Algorithms. Cambridge, MA: MIT Press; 2017.
- [54] Granger CWJ. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. Econometrica. 1969;37(3):424-438. doi:10.2307/1912791.
- [55] Pearl J. Causal Diagrams for Empirical Research. Biometrika. 1995;82(4):669-688. doi:10.1093/biomet/82.4.669.
- [56] Shannon CE. A Mathematical Theory of Communication. Bell Syst Tech J. 1948;27:379-423,623-656. doi:10.1002/j.1538-7305.1948.tb01338.x.
- [57] Cover TM, Thomas JA. Elements of Information Theory. 2nd ed. Hoboken: Wiley; 2006. doi:10.1002/047174882X.
- [58] Lindley DV. On a Measure of the Information Provided by an Experiment. Ann Math Stat. 1956;27(4):986-1005. doi:10.1214/aoms/1177728069.
- [59] Chaloner K, Verdinelli I. Bayesian Experimental Design: A Review. Stat Sci. 1995;10(3):273-304. doi:10.1214/ss/1177009939.
- [60] Kaelbling LP, Littman ML, Cassandra AR. Planning and Acting in Partially Observable Stochastic Domains. Artif Intell. 1998;101(1-2):99-134. doi:10.1016/S0004-3702(98)00023-X.
- [61] Altman E. Constrained Markov Decision Processes. Boca Raton: Chapman & Hall/CRC; 1999.

- [62] Li L, Chu W, Langford J, Schapire RE. A Contextual-Bandit Approach to Personalized News Article Recommendation. In: Proceedings of the 19th International Conference on World Wide Web. 2010. p. 661-670. doi:10.1145/1772690.1772758.
- [63] Auer P, Cesa-Bianchi N, Fischer P. Finite-Time Analysis of the Multiarmed Bandit Problem. *Mach Learn.* 2002;47:235-256. doi:10.1023/A:1013689704352.
- [64] Settles B. Active Learning Literature Survey. Computer Sciences Technical Report 1648. Madison: University of Wisconsin-Madison; 2009.
- [65] Dwork C, McSherry F, Nissim K, Smith A. Calibrating Noise to Sensitivity in Private Data Analysis. In: Theory of Cryptography Conference. LNCS 3876. Berlin: Springer; 2006. p. 265-284. doi:10.1007/11681878_14.
- [66] Dwork C, Roth A. The Algorithmic Foundations of Differential Privacy. *Found Trends Theor Comput Sci.* 2014;9(3-4):211-407. doi:10.1561/04000000042.
- [67] Kent K, Souppaya M. Guide to Computer Security Log Management. NIST Special Publication 800-92. Gaithersburg: National Institute of Standards and Technology; 2006. doi:10.6028/NIST.SP.800-92.
- [68] National Institute of Standards and Technology. NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management. Version 1.0. Gaithersburg: NIST; 2020.
- [69] Tabassi E. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. Gaithersburg: National Institute of Standards and Technology; 2023. doi:10.6028/NIST.AI.100-1.
- [70] Autio C, Schwartz R, Dunietz J, Jain S, Stanley M, Tabassi E, et al. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. Gaithersburg: National Institute of Standards and Technology; 2024. doi:10.6028/NIST.AI.600-1.