



Continual Red-Teaming and Guardrail Distillation for Tool-Using LLM Agents: Prompt-Injection Resistance with Utility Preservation

Wesley Gao
Computer Science, Stony Brook University, Stony Brook, NY, USA
Corresponding Author: Wesley Gao, E-mail: wgao_sb@proton.me

ARTICLE INFORMATION	ABSTRACT
Submitted: March 29, 2026 Accepted: June 21, 2026 Published: July 31, 2026 Volume: 2 Issue: 1 KEYWORDS Tool-using agents; indirect prompt injection; AgentDojo; action gating; guardrail distillation; continual red teaming; security-utility trade-off.	Tool-using language-model agents can convert indirect prompt injection into consequential actions, making guardrail quality a joint security, utility, and efficiency problem. This study evaluates a ReAct-style control, native tool filtering, deterministic self-verification, a calibrated action gate, and a distilled guardrail on AgentDojo v0.1.22. The evaluation covers 97 benign tasks, 629 canonical attack cases, ten attack formulations, and 32,456 recorded actions, of which 20,953 were assigned labels under the stated rules and 1,767 were unsafe. Archived end-to-end runs and fail-closed state replay were analyzed separately. The ReAct control reached 47.69% attack success, 50.08% attacked utility, and 69.07% benign utility. Native tool filtering reduced attack success to 6.84% while raising attacked and benign utility to 56.28% and 72.16%. The learned action gate produced a 0.00–0.95% attack-success bound, but attacked utility fell to 18.92–22.58% and false refusal rose to 46.27–47.76%. Distillation reduced serialized size from 146.40 to 3.52 KiB and probability-scoring time from 358.12 to 1.31 microseconds per action, yet lowered AUROC from 0.9971 to 0.9009 and produced a 71.64–73.13% false-refusal bound. Continual rehearsal also failed to retain protection: final seen-family unsafe recall was 21.71% with a 70.54-point peak-to-current seen-recall loss. The results show that compactness and low attack success do not establish a useful guardrail unless authorization fidelity and retention are measured concurrently.

1. Introduction

Large language models now retrieve information, select tools, and execute multi-step plans. ReAct established the influential pattern of interleaving reasoning, actions, and tool observations (Yao et al., 2023). This capability changes the security boundary because untrusted content can influence calls that send messages, modify records, disclose data, or spend resources. ToolEmu evaluates consequential failures in emulated environments (Ruan et al., 2024), while AgentHarm shows why autonomous systems require safety assessment over complete action trajectories rather than isolated responses (Andriushchenko et al., 2025). A tool agent must therefore be judged by both its text and its external effects.

Indirect prompt injection embeds adversarial instructions in an email, document, webpage, or tool response that an agent must consume. If the model treats this content as trusted control, it can replace the user’s objective with an attacker-selected action. Direct prompt injection first exposed the brittleness of instruction following (F. Perez & Ribeiro, 2022); indirect attacks then compromised retrieval- and tool-connected applications (Greshake et al., 2023). HouYi further established that practical black-box injection requires neither model weights nor privileged access (Y. Liu et al., 2023). The core failure is confusion between authorized instructions and untrusted data, not hostile vocabulary alone. Related work has consequently shifted from surface toxicity to behavior: evidence-based self-verification has been proposed for adversarial prompts, and multi-stage refusal has been framed as a way to preserve legitimate utility while resisting jailbreak behavior (Zheng et al., 2024; Zheng & Li, 2024). Adjacent research on deceptive interface microcopy adds a warning that apparently ordinary text can encode manipulative intent (Xu et al., 2025). Prompt-injection risk-card design likewise places privacy and data-integrity consequences within the human-oversight boundary of an agent system (Su, Rao, et al., 2026).

AgentDojo tests this problem through 97 realistic user tasks and 629 security cases in interactive environments (DeBenedetti et al., 2024). Each case links a legitimate goal, an attacker objective, an injection location, and consequential tools, enabling joint measurement of task completion and attacker success. This pairing exposes defenses that suppress attacks by blocking useful work. False rejection and tool cost must therefore accompany security and utility: PIGuard treats over-defense as a first-class failure mode (H. Li et al., 2025), and WildGuard separates harmfulness from refusal (Han et al., 2024). The same evaluation logic appears outside agent security in calibrated risk tiering, selective recruiter screening, and abstention-aware question answering (Y. Chen et al., 2023; Jin, 2025a; Zhong, Chen, et al., 2025). Across these domains, a score becomes operationally meaningful only when its threshold, rejection behavior, and cost of error are evaluated together.

The relevance of this problem extends beyond a single benchmark. Tool-connected AIOps systems retrieve operational documents, attribute incidents across sources, and summarize evidence that can precede consequential remediation (G. Liu et al., 2023; B. Zhang et al., 2024; G. Liu et al., 2024). More recent DevOps copilots explicitly join runbook retrieval with command-safety evaluation, while generalist-agent studies model reliability over complete tool-use trajectories (B. Zhang et al., 2026; Zhong et al., 2026). Although evaluated on different datasets, this work shows why retrieval quality, action authorization, and trajectory outcomes must be treated as one system rather than as independent text-processing components.

A fixed suite measures known attacks, whereas deployment changes the attacker’s target. Language models can generate adversarial tests at scale (E. Perez et al., 2022), HarmBench standardizes comparisons across attacks and defenses (Mazeika et al., 2024), and adaptive attacks can sharply degrade prompt-injection defenses (Zhan et al., 2025). Continual red teaming instead discovers failures, retains policy-relevant cases, retrains the guardrail, and attacks it again. Continual guardrail updating and distillation have been proposed directly for changing jailbreaks (Zheng et al., 2023). Time-aware conversational memory research separately formalizes confidence-gated writing, updating, and forgetting (Sun et al., 2023), while cross-cloud forecasting and shared-feature transfer examine generalization under distribution shift (He et al., 2024; Zhong, Pan, et al., 2025). The latter studies are methodological analogues rather than security evaluations, but together they motivate measuring both adaptation to a new family and retention of earlier protection.

Action-level control complements this loop. Task Shield checks whether a proposed action remains aligned with the user’s request (Jia et al., 2025); MELON uses masked re-execution to detect illegitimate influence from tool output (Zhu et al., 2025); and CaMeL separates control flow from untrusted data through capabilities (DeBenedetti et al., 2025). These methods place the decisive check before action, where intent, tool arguments, provenance, and side effects can be evaluated together. Evidence-chain RAG adds word-level attribution and provenance explanation, while deterministic provenance work makes the supporting chain inspectable (C. Li et al., 2024; Jin, 2025b). Review-grounded recommendation similarly evaluates whether generated explanations remain faithful to retrieved source text (Chang et al., 2026). Numerical-reasoning guardrails and evidence-ranked claim verification further show how a proposed output or operation can be checked against its evidence (Z. Li et al., 2026; Su, Chen, et al., 2026).

Runtime feasibility creates a second constraint. A contextual teacher may recognize unsafe actions accurately but be too expensive for every tool call, whereas a compact student may lose distinctions required for legitimate completion. Edge-oriented model distillation and hybrid-cloud LLM deployment provide transferable evidence that model size, latency, and infrastructure placement must be evaluated jointly (B. Zhou, Wang, et al., 2025; Xin, 2025b). TinyLLM-assisted intrusion detection similarly illustrates the attraction of compact decision models in real-time monitoring (Lu & Zhou, 2024). Guardrail compression must therefore be evaluated as a security–utility–efficiency trade-off rather than as a size-reduction exercise.

This study evaluates continual red teaming and lightweight guardrail distillation on AgentDojo version 0.1.22. It compares a ReAct-style control, tool filtering, self-verification, action gating, and a distilled student under common tasks and attacks. The continual experiment presents ten archived attack formulations in a fixed stream, retrains the student on each family plus clean actions, and rehearses up to 512 earlier actions. Attack success, attacked and benign utility, false rejection, and tool calls define the security–utility analysis. Three questions guide the study: which intervention best preserves utility while reducing injection success; whether continual exposure improves adaptation without forgetting earlier attacks; and whether teacher decisions can be compressed into a lower-cost gate without losing useful discrimination.

2. Literature Review

2.1 Prompt Injection, Behavioral Safety, and Tool-Agent Evaluation

Prompt-injection research has moved from isolated strings to application data flows. PromptInject systematized direct overrides and instruction leakage (F. Perez & Ribeiro, 2022), whereas indirect injection places adversarial control inside content processed during legitimate work (Greshake et al., 2023). BIPIA formalized indirect attacks across tasks and attack patterns (Y. Liu et al., 2024). InjecAgent constructed adversarial interactions around user tasks, external data, and tools (Zhan et al., 2024), while

complementary benchmarking compared indirect attacks and defenses across language models (Yi et al., 2025). AgentDojo advanced trajectory-level evaluation by scoring legitimate and attacker objectives in stateful environments (Debenedetti et al., 2024).

Representation-based defenses distinguish authority from data. Spotlighting marks or transforms external content (Hines et al., 2024), while instruction-hierarchy training prioritizes privileged messages during conflicts (Wallace et al., 2024). Jatmo narrows exposure by removing general instruction-following behavior from task-specific models (Piet et al., 2024). StruQ makes the control–data boundary explicit through structured queries and secure training (S. Chen, Piet, et al., 2025). These methods address the semantic ambiguity that enables injection, but depend on authority cues or specialization surviving distribution shift.

Learned defenses instead alter the decision policy. SecAlign uses preference optimization to favor trusted instructions over adversarial data (S. Chen, Zharmagambetov, et al., 2025). Its coverage remains tied to represented attacks, because visible formatting rules, refusal patterns, and checks become targets for adaptation (Zhan et al., 2025). Continual training responds by registering new families, updating the defense, and retesting both past and unseen attacks; rehearsal is needed because learning the newest family can erase prior protection. Continual jailbreak red teaming, evidence-based self-verification, and behavior-level refusal therefore represent complementary intervention points: update the policy, inspect the evidence for a response, or block behavior that conflicts with the user’s objective (Zheng et al., 2023; Zheng et al., 2024; Zheng & Li, 2024).

Agent-oriented defenses focus more specifically on authorization. Task Shield interprets injection as drift from the user’s objective (Jia et al., 2025), MELON detects influence through masked re-execution (Zhu et al., 2025), and CaMeL enforces control–data separation and capability limits (Debenedetti et al., 2025). All treat provenance and action trajectory as central. Architectural isolation provides strong control but requires tool-specific policies; a pre-action gate is easier to deploy but must classify contextual actions accurately under strict latency constraints.

Several adjacent security domains reinforce the need to evaluate sequences and consequences. CloudTrail attack-chain modeling localizes stages across event sequences, host-based intrusion detection localizes suspicious windows before generating forensic narratives, and IoT/DDoS studies join detection to mitigation decisions (Bai et al., 2026; Xin, 2026c; Xin et al., 2024; B. Wang et al., 2024). Language-guided feature selection for intrusion detection and TinyLLM-assisted monitoring further connect semantic representations with compact runtime decisions (Y. Li & Lu, 2025; Lu & Zhou, 2024). Their transferable contribution is an evaluation pattern for action-window localization, rare-event detection, and the cost of acting on a false alarm. Security also includes confidentiality and policy adherence, not only attack success. Federated preference learning for knowledge-grounded chat illustrates how personalization can be separated from raw-data centralization through differential privacy (Kuo et al., 2025). Strategy-aware response control in emotional-support dialogue shows a different form of constrained generation in which the response must follow an intended support strategy (Y. Zhang & Zhou, 2026). These are cross-domain examples, but they clarify why an agent guardrail may need to protect data and preserve the user-authorized mode of interaction simultaneously.

2.2 Evidence Grounding, Provenance, and Constrained Retrieval

Retrieval can reduce unsupported generation only if the retrieved material is relevant, attributable, and kept distinct from control instructions. Evidence-grounded DevOps RAG links operations answers to private documents, and bilingual OpsLLM work combines retrieval with incident attribution and alert summarization (B. Zhang et al., 2024; G. Liu et al., 2024). Evidence-chain approaches add word-level hallucination detection, source attribution, and deterministic provenance (C. Li et al., 2024; Jin, 2025b). Retrieval–summary integration for code intelligence and budgeted multi-hop retrieval further show that finding evidence and allocating retrieval effort are separate design problems (Y. Li, 2024; Su et al., 2025). RAG-quality prediction and calibrated abstention on unanswerable questions make the same distinction explicit: retrieval coverage and answer acceptance require separate evaluation (R. Zhang et al., 2025; Zhong, Chen, et al., 2025).

The provenance requirement generalizes across applications. Accounting-aware retrieval grounds institutional review in disclosures, while multi-regulation RAG structures legal evidence for product counsel (Y. Chen et al., 2025; J. Li & Zhou, 2026). Financial-search reranking combines query rewriting, hybrid retrieval, and listwise relevance ranking; multilingual humanitarian RAG tests whether evidence access and calibration survive language variation (Meng et al., 2026; Y. Chen & Xu, 2026). A therapist-facing copilot similarly retrieves and ranks support strategies from multi-turn dialogue, and a climate-claim agent ranks evidence against narrative claims (Y. Zhang & H. Zhang, 2025a; Su, Chen, et al., 2026). Collectively, they demonstrate a transferable design requirement: an assistant should expose which evidence influenced a recommendation before that recommendation becomes a consequential action.

AIOps provides the closest adjacent setting because generated guidance can become a command, ticket, or remediation step. Multi-source root-cause attribution combines signals for CPU and network faults, and retrieved normal prototypes provide explicit comparators for failure-injection anomalies (G. Liu et al., 2023; Xin, 2025a). Self-supervised LogBERT-style detection and retrieval-grounded HDFS analysis couple anomaly recognition with interpretable failure narratives (Xin, 2026g; Sun et al., 2026). Conformal alert control extends that chain to evidence-grounded incident tickets, while cost-aware LogEval routing chooses among parsing, detection, diagnosis, and summarization operations (Xin, 2026e; C. Li et al., 2026). Incident visualization cards and a DevOps copilot with command-safety evaluation add human-facing provenance at the final handoff (Nie et al., 2026; B. Zhang et al., 2026). These studies support the need for traceable operational decisions, but their retrieval or anomaly metrics must not be read as evidence about AgentDojo attack success.

Rubrics offer another form of explicit evidence constraint. Rubric-grounded automated essay scoring makes assessment criteria auditable rather than treating a model score as self-justifying (Xin, 2026a). Accounting-aware retrieval likewise binds an assessment to disclosures and supporting financial records (Y. Chen et al., 2025). The connection to self-verification is methodological: an output is accepted only when it can be related to an external criterion.

2.3 Calibration, Selective Action, and Logged-Policy Evaluation

Specialized guardrails provide a basis for a pre-action gate. Llama Guard demonstrated compact, policy-conditioned classification of unsafe prompts and responses (Inan et al., 2023). WildGuard broadened moderation to harmfulness and refusal using adversarial data (Han et al., 2024), while PIGuard explicitly measures benign over-defense in injection detection (H. Li et al., 2025). A tool-agent guard must go further by deciding whether a concrete action is authorized, including ordinary operations whose arguments originated in untrusted content.

Rare unsafe actions create an operating-point problem rather than a ranking problem alone. Cost-sensitive fraud detection explicitly addresses extreme imbalance (Jin et al., 2024a), while a model-risk-oriented default workflow pairs probability calibration with distribution-free uncertainty quantification and explanation (Jin et al., 2024b). Going-concern prediction adds resampling, focal loss, and cost sensitivity under a different imbalanced outcome (Y. Chen et al., 2024). Methodologically, these studies ask whether apparently strong discrimination remains useful after class imbalance, threshold selection, and asymmetric costs are considered.

Selective prediction makes the cost of abstention explicit. Calibrated credit-risk tiering connects uncertainty to rejection, and calibrated resume-job matching evaluates selective refusal in an LLM-assisted screening workflow (Y. Chen et al., 2023; Jin, 2025a). Evidence-calibrated RAG uses retrieval coverage and abstention calibration for unanswerable questions (Zhong, Chen, et al., 2025), while fair credit scoring links model decisions to counterfactual explanations (Xin, 2026b). Together, these applications support reporting acceptance, rejection, calibration, and subgroup cost as separate quantities.

Uncertainty aggregation provides a second methodological analogy. Late fusion learns how to combine calibrated perceptual signals, while LiDAR-camera tracking integrates measurements under explicit state uncertainty (Xin, 2025c; Xin, 2026d). The underlying tasks differ fundamentally from action gating, but both studies caution against treating a raw model score as a calibrated decision probability. In the present setting, the analogous question is whether action-level probabilities remain meaningful once decisions are composed across a trajectory.

Operational systems make the action rule especially visible. Conformal demand envelopes translate uncertainty into bounded GPU oversubscription, workload-semantic forecasting connects multiple horizons to operational risk curves, and profit-aware admission control couples prediction with a cost-sensitive accept/reject policy (S. Chen et al., 2024; Zhao et al., 2025; Zhao et al., 2026). Power-aware inventory planning and GPU-memory prediction likewise connect forecasts to resource commitments (He et al., 2026; C. Wang et al., 2025). Capacity forecasting under heterogeneous clusters and noisy-neighbor VM risk modeling emphasize uncertainty and interference rather than average error alone (He et al., 2023; Nie & Zheng, 2024). The shared implication is that a decision system must report both predictive quality and the operational cost of acting.

Logged-policy evaluation is relevant to the evidence mode used in this study but must be described carefully. Offline counterfactual evaluation for advertising slots and conservative off-policy selection for dynamic bidding both evaluate candidate policies without deploying every candidate online (Mu et al., 2025; Ye et al., 2026). Digital-twin dispatching adds a sequential example in which offline reinforcement learning is assessed under simulated operating dynamics (L. Zhang et al., 2025). These estimands, assumptions, and action spaces differ from AgentDojo's fail-closed replay. The studies therefore serve only as methodological analogues for support awareness, conservative policy comparison, and the limits of conclusions drawn from logged behavior; they are not baselines for the present experiment.

2.4 Continual Adaptation and Distribution Shift

Continual protection requires more than high performance on the newest attack. Online guardrail updating makes changing jailbreak families part of the training loop, while confidence-gated conversational memory treats writing, retrieval, updating, and forgetting as distinct operations (Zheng et al., 2023; Sun et al., 2023). The memory setting is not adversarial tool use, but its separation of update and forgetting motivates an analogous guardrail question: did the latest update add protection without erasing earlier distinctions? Automated red teaming supplies new failures (E. Perez et al., 2022), whereas adaptive attacks test whether visible rules remain exploitable (Zhan et al., 2025). A continual evaluation should consequently report current-family acquisition, earlier-family retention, next-family transfer, and benign false positives.

Distribution-shift research offers additional evaluation patterns. Cross-cloud transfer separates workload and topology changes, and few-shot cold-start forecasting tests adaptation for new inference tenants (He et al., 2024; He et al., 2025). Shared-feature theory studies transfer when domains overlap only approximately (Zhong, Pan, et al., 2025). Macro-market fusion and bike-demand forecasting examine uncertainty under changing external or seasonal regimes (H. Zhou & Zhang, 2025; Xin, 2026f). Across these domains, the useful evaluation principle is to stratify performance by regime rather than rely on one pooled number.

2.5 Compact Guardrails and Deployment Constraints

Knowledge distillation can transfer a costly contextual teacher into a compact runtime model. Performance-guided distillation combines teacher scores, hard-negative mining, and validation to sharpen a student boundary (Di Palo et al., 2024). Successful adaptive attacks supply hard unsafe cases, while benign actions near the threshold reveal excessive rejection. Action AUROC alone is insufficient because one false refusal can terminate an entire trajectory.

Adjacent efficiency studies show why compression must be audited for what it discards. VMAF distillation targets an edge-deployable proxy, while a lightweight medical foundation model uses parameter-efficient transfer and a subject-independent BCI model reduces calibration burden through self-supervised pretraining (B. Zhou, Wang, et al., 2025; Mi et al., 2025; Wu et al., 2025). TinyLLM-assisted intrusion detection prioritizes real-time inference, and hybrid-cloud LLM architecture makes cost and placement explicit (Lu & Zhou, 2024; Xin, 2025b). Across these objectives, smaller size or lower latency is incomplete evidence unless task fidelity is measured after compression.

Resource planning further constrains deployment. Workload-semantic demand forecasting, power-aware inventory planning, and GPU-memory prediction make compute requirements part of the decision pipeline rather than an afterthought (Zhao et al., 2025; He et al., 2026; C. Wang et al., 2025). A compact action gate should be assessed in the same operational spirit: model size and scoring time matter, but only beside security, false refusal, and task completion.

2.6 Human Oversight and Risk Communication

A gate that blocks a consequential action should provide an inspectable reason. Scientific-search evidence cards organize retrieval transparency and visual evidence hierarchy, while medical explanation and patient-safety cards expose uncertainty and rubric-based risk to end users (Jin, 2025c; Zhong, Wu, et al., 2025; B. Zhou, Li, et al., 2025; C. Li et al., 2025). Breast-ultrasound explanation cards and counseling-support cards apply related visual principles in medical and mental-health contexts (Ye et al., 2025; Y. Zhang & H. Zhang, 2025b). Financial-risk dashboards and AI-infrastructure capacity cards extend visual risk communication to operational decisions (Z. Li et al., 2025; G. Liu et al., 2025). For a future operator view, these patterns suggest showing provenance, uncertainty, the blocked action, and the expected side effect while treating visualization as an oversight aid rather than a defense by itself.

Prompt-injection risk cards and evidence-constrained incident visualizations are more directly aligned with tool-agent oversight (Su, Rao, et al., 2026; Nie et al., 2026). They position a human reviewer between model output and a consequential operation. That position is compatible with selective escalation, but the present study evaluates automated fail-closed decisions rather than human comprehension or intervention quality. Human-interface effectiveness therefore remains a separate empirical question.

2.7 Synthesis and Research Gap

The unresolved problem is integration. Benchmarks supply realistic attacks but not an adaptive learning process; source marking, structured inputs, preference optimization, action alignment, and capability systems impose different security, utility, and deployment costs. Guardrails offer compact classifiers, while red teaming offers scalable failure discovery, but neither establishes stable protection across changing attacks. Evidence-grounded systems add provenance, selective-prediction studies add calibration and rejection costs, and operational forecasting studies add resource-aware decision rules. These cross-domain bodies of work contribute transferable evaluation concepts, not evidence of AgentDojo effectiveness. Efficiency is part of robustness because protection must preserve the information flow and tool access required by legitimate work. Repeated testing

must also distinguish durable adaptation from temporary fit to one attack family. This study unifies the directly relevant strands: successive archived attack families create training pressure, teacher judgments encode action authorization, a distilled student supplies runtime decisions, and AgentDojo measures attack success, utility, false rejection, and tool cost together.

3. Methodology

3.1 Dataset and Experimental Corpus

Experiments used AgentDojo v0.1.22 (Debenedetti et al., 2024), commit 2077ff64cdf38aa97ee87a69914224c796d2860. Its 59,761,853-byte archive had SHA-256 digest d8b12f097b0f910f44a081231a55828fd6974615c3afaba3d747886c1a3892e6.

Version 1.1.1 comprised Banking with 16 user tasks and 9 injection goals, Slack with 21 and 5, Travel with 20 and 7, and Workspace with 40 and 6. Their products yielded 144, 105, 140, and 240 cases: 97 benign tasks, 27 goals, and 629 task-goal pairs across 69 tools.

Table 1 summarizes the four suites and exact primary denominators.

Table 1. AgentDojo v1.1.1 suite composition and primary evaluation denominators.

Suite	User tasks	Injection goals	Security pairs	Declared tools	Benchmark
Banking	16	9	144	11	v1.1.1
Slack	21	5	105	11	v1.1.1
Travel	20	7	140	28	v1.1.1
Workspace	40	6	240	24	v1.1.1
Total	97	27	629	74	AgentDojo 0.1.22

The 21,999 JSON logs supplied identifiers, injections, calls, outputs, errors, scores, and durations; task files supplied prompts, ground-truth actions, and scorers. The primary protocol used 97 clean and 629 important_instructions executions. Continual evaluation added direct, ignore_previous, inecagent, three name-removal variants, two wrong-name variants, and tool_knowledge. Together with 451 clean actions, the ten formulations yielded 32,456 actions. The stated rules assigned 20,953 non-ambiguous labels, including 1,767 unsafe; ambiguous actions entered replay but not fitting. Formulations were correlated observations (Mazeika et al., 2024; Zhan et al., 2025).

Table 2 maps the stored fields, tool taxonomy, and scorer types to their analytical roles.

Table 2. Record schema, action taxonomy, and scorer roles used in the empirical analysis.

Item	Count	Role
Execution JSON records	21,999	Archived outcomes and trajectories
Top-level fields	12	Suite, task, attack, messages, outcomes, duration
Clean user tasks	97	Benign utility and safe-action labels
Canonical attacked pairs	629	Primary security evaluation
Attack formulations	10	Continual red-team stream
Pure-read endpoints	45	Low-side-effect actions
State-changing endpoints	29	Guarded consequential actions
Post-state injection checkers	25	Environment-state attack scoring
Trace-only injection checkers	1	Ordered Slack transition scoring
Output-only injection checkers	1	Travel response-content scoring
Inference-excluded fields	6 classes	Outcomes, IDs, attacks, injections, ground truth

3.2 Evidence Modes and Agent Control

Archived end-to-end evidence supported the ReAct control and native filter; offline replay evaluated guardrails over recorded control actions.

The archived gpt-4o-2024-05-13 control alternated calls and results (Yao et al., 2023). Its native filter restricted tools from the query before 97 clean and 629 attacked cases.

Table 3 fixes the decision point, training source, threshold, and evidence mode for every primary method; Figure 1 shows the shared evaluation flow.

Table 3. Primary methods, decision points, training sources, thresholds, and evidence modes.

Method	Decision point	Training	Threshold	Evidence
ReAct-style control	Native function-calling loop	None	None	Archived end-to-end
Native tool filter	Tools selected before the loop	Archived defense	Native configuration	Archived end-to-end
Self-verification	Before tool call or response	Frozen intent/provenance rules	Fixed deterministic rule	Fail-closed replay bounds
Action gate	Before tool call or response	Grouped HGB + sigmoid calibration	$\leq 5\%$ inner safe-action FPR	Fail-closed replay bounds
Distilled guardrail	Before tool call or response	L2 logistic student, T=2 soft targets	$\leq 5\%$ inner safe-action FPR	Fail-closed replay bounds

Archived end-to-end evidence and replay-bounded action-policy evidence



Figure 1. Experimental workflow from archived executions through grouped action modeling, fail-closed policy evaluation, and official scoring.

3.3 Guardrail Methods

Deterministic self-verification compared each action with the request and latest output using function class, overlap, argument support, imperatives, sensitive entities, prior actions, and errors. It denied mutations when unsupported intent coincided with tool-derived arguments or tool text supported the action more strongly than the request; responses used a parallel rule. The provenance-free ablation retained intent and imperatives (Debenedetti et al., 2025; Jia et al., 2025).

This design was motivated by the broader evidence-chain principle that a consequential decision should remain traceable to authorized source material, while behavior-level defenses should judge the proposed action rather than merely classify the surrounding text (C. Li et al., 2024; Jin, 2025b; Z. Li et al., 2026; Zheng & Li, 2024). Together, these ideas informed the separation of provenance features from the authorization rule tested here.

The probabilistic gate added one-hot function identity. Canonical prefixes were unsafe when an action changed the official injection scorer to success; clean, non-mutating, and legitimate ground-truth actions were safe, while unresolved actions were unlabeled. The other nine formulations matched successful episodes to canonical unsafe function–argument signatures, falling back to injection functions and goal–action similarity. The class-balanced HGB teacher used learning rate 0.05, 150 iterations, depth 3, minimum leaf 10, L2 penalty 1.0, and grouped inner-split sigmoid calibration.

Class balancing and post-hoc calibration were selected because rare-event decision systems can rank examples well yet behave poorly at a fixed operating point; cost-sensitive fraud and bankruptcy models, calibrated probability-of-default workflows, and

selective-rejection systems make the same methodological distinction outside agent security (Y. Chen et al., 2024; Jin et al., 2024a; Jin et al., 2024b; Jin, 2025a). The present thresholds were nevertheless estimated only from AgentDojo actions.

The distilled guardrail was logistic SGD with L2 penalty $\alpha = 5 \times 10^{-5}$, numeric and function features, and 256 hashed character 3-grams from each action and latest output. Temperature $T = 2$ softened calibrated teacher probabilities; weighted duplicates implemented soft-target loss over 20 epochs. Numeric-only and hard-label students supplied ablations (Di Palo et al., 2024; H. Li et al., 2025).

The efficiency audit therefore treated compression as a deployment trade-off rather than evidence that task behavior had been preserved. That distinction also arises in compact intrusion models, parameter-efficient foundation models, and edge-quality distillation, where size or latency gains must be read beside task-specific error (Lu & Zhou, 2024; Mi et al., 2025; B. Zhou, Wang, et al., 2025).

Ten continual rounds followed the listed order. Each mixed the new family and clean actions with up to 256 earlier actions per class in a 512-action reservoir, then ran five SGD epochs; a comparator omitted rehearsal. The higher 95th percentile of safe training scores set the threshold. The held-out fifth fold measured seen- and next-family recall (E. Perez et al., 2022). Peak-to-current loss was $L = \max(R_{\text{seen}} - R_{\text{next}})$.

Both adaptation and retention were measured because confidence-gated update systems require explicit forgetting control, and continual guardrail proposals likewise couple online updates with rehearsal or distillation (Sun et al., 2023; Zheng et al., 2023). These sources motivate the evaluation dimensions, not the fixed attack order or reservoir size used in this experiment.

3.4 Fail-Closed State Replay

Replay rebuilt each initialized environment and processed calls sequentially. Allowed calls ran locally; denial stopped the episode, suppressed later calls, and assigned an empty response. Allow-all retained the archived response. Official task and injection functions scored the resulting state and trace; responses were pseudo-actions so output-only goals remained gateable.

Allow-all re-execution covered 726 records. Matches were 726/726 for call count, 724/726 for errors, 697/726 for utility, 623/629 for attacked security, and 689/726 across all applicable fields. Decisions and costs therefore covered every record. Task and security results used exact full-corpus bounds: stable successes divided by 97 or 629 formed the lower endpoint; adding all scorer-unstable records formed the upper endpoint.

Table 4 reports the validation counts by suite.

Table 4. Allow-all state-replay validation by suite. Security matches exclude clean records, whose security field has no safety meaning.

Suite	Records	Utility matches	Security matches	Security records	Error-count matches	Fully matching records
Banking	160	140	144	144	160	140
Slack	126	121	105	105	126	121
Travel	160	160	140	140	158	158
Workspace	280	276	234	240	280	270
Total	726	697	623	629	724	689

3.5 Validation and Statistical Analysis

Five grouped folds used (suite_name, user_task_id) and seed 20260729. SHA-256 assignment kept each task's goals and variants together and gave every action one out-of-fold decision. A grouped 80:20 split fitted and sigmoid-calibrated the teacher. Student thresholds used that in-training subset, not independent calibration; both models selected the higher 95th safe-score percentile, limiting false positives to 5% apart from ties. Explicit injected-value metadata, attack-family fields, outcomes, identifiers, and test ground truth were excluded; runtime-observable action text and latest tool output remained features.

ASR measured injection success; AU and BU measured attacked and benign task success; FRR measured lost control-successful clean tasks. Costs covered executed, mutating, errored, and saved calls. Secondary measures were safe utility, AUROC, AUPRC, Brier score, equal-mass calibration error, size, and CPU time. Efficiency values were means over five fold models; host-specific timing used predict_proba on precomputed features.

Archived 95% intervals used 10,000 cluster-bootstrap samples, resampling suite task groups with their injection goals; replay used exact bounds. Paired stable-record ASR, AU, and BU differences used exact McNemar tests with Holm adjustment across ten comparisons. Folds, replay, bootstrap, and tables used deterministic seeds and stable ordering.

4. Results and Discussion

4.1 Evidence Scope and Replay Fidelity

The ReAct-style control and native tool filter are archived end-to-end executions with clustered confidence intervals; the other primary defenses use fail-closed state replay. Allow-all replay fully matched 689 of 726 episodes, with utility agreement in 697, security agreement in 623 of 629 attacks, error-count agreement in 724, and exact call sequences throughout. The six security and 29 utility discrepancies produced exact worst-to-best bounds rather than assumed outcomes: widths were 0.95 percentage points for ASR, 3.66 for attacked utility, and 6.19 for benign utility.

Separating archived and reconstructed evidence also preserves an auditable lineage when local state replay diverges from an original run. Evidence-chain RAG, deterministic provenance explanations, retrieval-grounded failure narratives, and incident visualization frameworks address the same general need to connect a conclusion to inspectable evidence, although none validates the replay agreement reported here (C. Li et al., 2024; Jin, 2025b; Nie et al., 2026; Sun et al., 2026).

4.2 Overall Security–Utility Performance

The control reached 47.69% ASR (95% CI [42.61%, 52.62%]), 50.08% attacked utility (AU; [42.93%, 56.76%]), and 69.07% benign utility (BU; [59.79%, 78.35%]), with 4.512 attacked calls. The native tool filter reduced ASR to 6.84% [4.61%, 9.22%] while raising AU to 56.28% [48.17%, 64.07%] and BU to 72.16% [63.92%, 80.41%]. Its 14.93% false-refusal rate (FRR) and 3.491 calls made it the strongest observed Pareto compromise.

Self-verification produced 9.54%–10.49% ASR, 31.00%–34.66% AU, 52.58%–58.76% BU, and 22.39%–23.88% FRR. Removing provenance raised ASR to 19.55%–20.51% but improved AU to 36.57%–40.22%, BU to 63.92%–70.10%, and FRR to 5.97%–7.46%. Provenance improved security but increased legitimate rejection, showing why action alignment requires task-level utility measurement (Jia et al., 2025).

The action gate reached 0.00%–0.95% ASR, but AU fell to 18.92%–22.58%, BU to 37.11%–43.30%, and FRR rose to 46.27%–47.76%. The distilled guardrail reached 5.56%–6.52% ASR, 11.92%–15.58% AU, 18.56%–24.74% BU, and 71.64%–73.13% FRR. Their 2.636 and 2.308 mean calls largely reflected early termination; low ASR alone overstated deployability.

This is a system-level selective-prediction failure: rejecting most risky actions is not useful if the same decision rule suppresses legitimate trajectories. Cross-domain work on uncertainty-driven rejection, calibrated default estimation, selective recruiter screening, and abstention-aware retrieval similarly separates confidence quality from the cost of acting or refusing (Y. Chen et al., 2023; Jin et al., 2024a; Jin, 2025a; Zhong, Chen, et al., 2025). The comparison is methodological; the present security and utility rates come only from AgentDojo.

Table 5 and Figure 2 show the primary operating points. Table 6 adds archived controls: prompt repetition achieved 27.82% ASR and 67.25% AU, the transformer detector 7.95% ASR and 21.14% AU, and spotlighting 41.65% ASR and 55.64% AU.

Table 5. Main security, utility, false-refusal, and tool-cost results. Archived rows show clustered 95% confidence intervals; replay rows show exact full-corpus scorer-stability bounds.

Method	Evidence	ASR % [CI or bounds]	Attacked utility % [CI or bounds]	Clean utility % [CI or bounds]	Safe utility % [point or bounds]	False refusal % [point or bounds]	Mean attacked calls
ReAct-style control	Archived end-to-end	47.69 [42.61, 52.62]	50.08 [42.93, 56.76]	69.07 [59.79, 78.35]	29.73	0.00	4.512
Native tool filter	Archived end-to-end	6.84 [4.61, 9.22]	56.28 [48.17, 64.07]	72.16 [63.92, 80.41]	52.62	14.93	3.491
Self-verification	Fail-closed trace replay	9.54–10.49	31.00–34.66	52.58–58.76	26.71–31.32	22.39–23.88	3.097
Action gate	Fail-closed trace replay	0.00–0.95	18.92–22.58	37.11–43.30	18.92–23.53	46.27–47.76	2.636
Distilled guardrail	Fail-closed trace replay	5.56–6.52	11.92–15.58	18.56–24.74	10.97–15.58	71.64–73.13	2.308

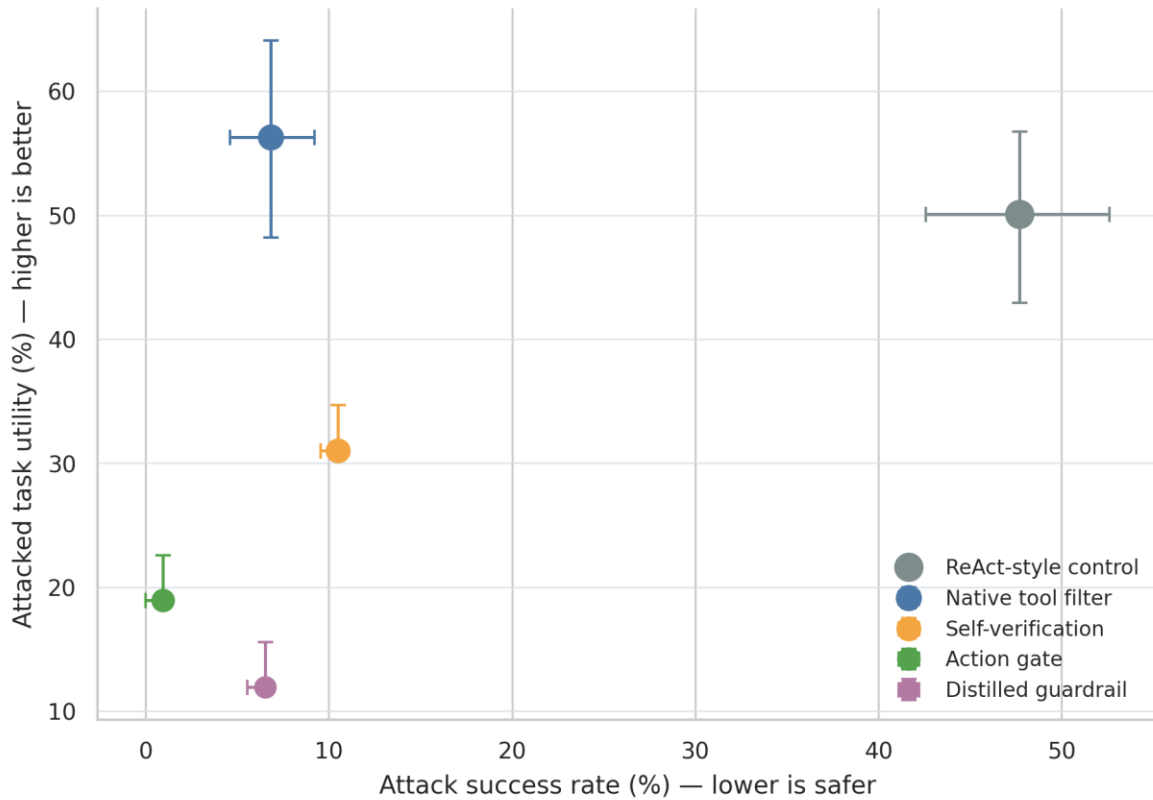


Figure 2. Security-utility frontier for the five primary methods. Archived error bars are clustered 95% confidence intervals; replay error bars are exact scorer-stability bounds; marker size encodes mean calls.

Table 6. Recomputed archived diagnostics for native AgentDojo defenses under the canonical attack.

Method	Evidence	ASR % [CI or bounds]	Attacked utility % [CI or bounds]	Clean utility % [CI or bounds]	Safe utility % [point or bounds]	False refusal % [point or bounds]	Mean attacked calls
ReAct-style control	Archived end-to-end	47.69	50.08	69.07	29.73	0.00	4.512
Native tool filter	Archived end-to-end	6.84	56.28	72.16	52.62	14.93	3.491
Repeat user prompt	Archived end-to-end	27.82	67.25	84.54	52.62	1.49	4.955
Transformer PI detector	Archived end-to-end	7.95	21.14	41.24	19.24	49.25	4.917
Spotlighting with delimiters	Archived end-to-end	41.65	55.64	72.16	37.20	8.96	4.302

4.3 Suite and Attack Variation

Risk differed by application. Table 7 and Figure 3 show control ASR of 92.38% in Slack, 62.50% in Banking, 40.42% in Workspace, and 11.43% in Travel; native filtering reduced these to 8.57%, 11.11%, 4.17%, and 5.71%. The action gate recorded 0.00% replay ASR in every suite, but AU ranged from 5.88% in Slack to 38.89% in Banking. Distilled-guardrail ASR ranged from 0.71% in Travel to 11.81% in Banking, while Slack and Workspace AU fell to 2.94% and 3.78%.

Table 7. Per-suite security, utility, and call counts. Replayed rates use scorer-stable cases, with their exact denominators shown.

Method	Suite	Security N	ASR %	Attacked utility N	Attacked utility %	Clean N	Clean % utility	Mean calls
ReAct-style control	Banking	144	62.50	144	69.44	16	75.00	3.042
ReAct-style control	Slack	105	92.38	105	63.81	21	80.95	7.467
ReAct-style control	Travel	140	11.43	140	64.29	20	65.00	6.457
ReAct-style control	Workspac e	240	40.42	240	24.17	40	62.50	2.967
Native tool filter	Banking	144	11.11	144	56.94	16	81.25	2.090
Native tool filter	Slack	105	8.57	105	47.62	21	80.95	4.305
Native tool filter	Travel	140	5.71	140	70.00	20	90.00	5.864
Native tool filter	Workspac e	240	4.17	240	51.67	40	55.00	2.592
Self-verification	Banking	144	18.75	126	45.24	14	57.14	1.944
Self-verification	Slack	105	16.19	102	14.71	19	57.89	3.105

Method	Suite	Security N	ASR %	Attacked utility N	Attacked utility %	Clean N	Clean %	utility	Mean calls
Self-verification	Travel	140	5.71	140	55.71	20	60.00		6.257
Self-verification	Workspace	234	3.42	238	18.91	38	52.63		1.942
Action gate	Banking	144	0.00	126	38.89	14	57.14		1.458
Action gate	Slack	105	0.00	102	5.88	19	31.58		1.571
Action gate	Travel	140	0.00	140	15.71	20	15.00		6.071
Action gate	Workspace	234	0.00	238	17.65	38	50.00		1.804
Distilled guardrail	Banking	144	11.81	126	25.40	14	28.57		1.556
Distilled guardrail	Slack	105	3.81	102	2.94	19	10.53		1.676
Distilled guardrail	Travel	140	0.71	140	22.14	20	15.00		4.300
Distilled guardrail	Workspace	234	5.56	238	3.78	38	23.68		1.875

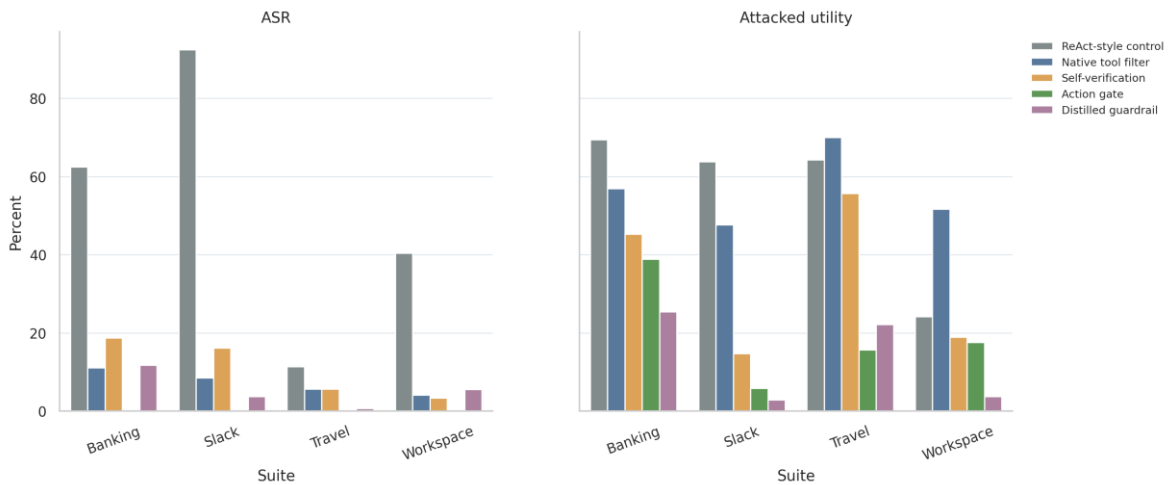


Figure 3. Per-suite attack success and attacked utility for all primary methods. Replayed bars use scorer-stable cases.

Attack wording also mattered (Table 8). Direct, ignore_previous, and InjecAgent formulations produced 3.66%, 5.41%, and 5.72% ASR, whereas important_instructions reached 47.69%. Name-removal variants remained at 44.83%–46.10%; wrong-name variants fell to 23.21%–23.69%; tool knowledge reached 34.50%. One template therefore did not represent performance across the ten formulations, consistent with adaptive-defense concerns (Zhan et al., 2025).

Table 8. Archived ReAct outcomes, exact-payload exposure, and call cost across ten attack formulations.

Attack family	Runs	ASR %	Utility %	Exact-payload exposure %	Mean calls
direct	629	3.66	67.25	52.31	3.571
ignore_previous	629	5.41	66.77	77.11	3.603
injecagent	629	5.72	68.52	52.31	3.599
important_instructions	629	47.69	50.08	51.99	4.512
important_instructions_no_user_name	629	44.83	52.15	52.15	4.428
important_instructions_no_model_name	629	46.10	51.83	51.99	4.455
important_instructions_no_names	629	45.79	49.60	51.67	4.467
important_instructions_wrong_model_name	629	23.69	65.82	52.15	4.037
important_instructions_wrong_user_name	629	23.21	65.02	51.67	3.990
tool_knowledge	629	34.50	57.71	36.57	4.221

The suite and formulation gaps are also a distribution-shift warning. Cross-cloud transfer, cold-start workload forecasting, shared-feature adaptation, and changing-regime demand forecasting all motivate reporting performance after the operating distribution moves rather than extrapolating from one aggregate score (He et al., 2024; He et al., 2025; Zhong, Pan, et al., 2025; Xin, 2026f). Those application domains differ from prompt injection, but the evaluation principle transfers directly.

4.4 Action Discrimination and Distillation

Across 20,953 labeled actions, including 1,767 unsafe actions, Table 9 and Figures 4□□ show teacher AUROC 0.9971, AUPRC 0.9814, Brier 0.0083, and ECE 0.0009. Strong action discrimination did not prevent high task FRR because a rejected action can invalidate a trajectory. The student reached 0.9009 AUROC, 0.7955 AUPRC, 0.0353 Brier, and 0.0379 ECE; its teacher-probability error was 0.0500 with 92.7218% decision agreement.

Table 9. Grouped out-of-fold action-detector discrimination and calibration.

Detector	Labeled actions	Unsafe actions	AUROC	AUPRC	Brier	ECE
Action-gate teacher	20953	1767	0.9971	0.9814	0.0083	0.0009
Distilled guardrail	20953	1767	0.9009	0.7955	0.0353	0.0379
Numeric-only student	20953	1767	0.9749	0.8542	0.0256	0.0140
Hard-label student	20953	1767	0.9926	0.9366	0.0249	0.0234
Deterministic verifier	20953	1767	0.9754	0.8894	0.0246	0.0301

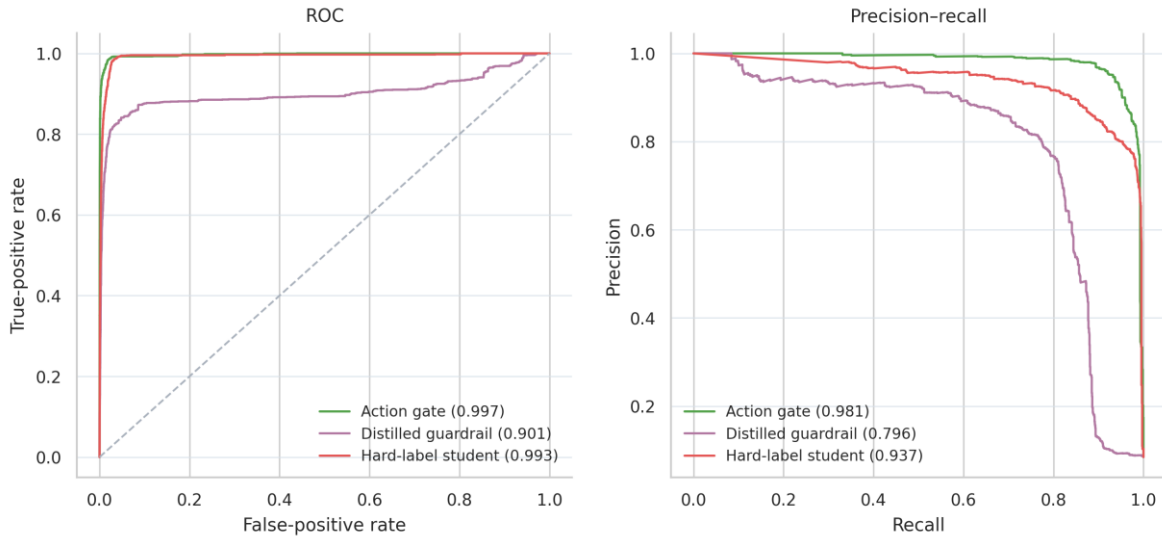


Figure 4. Out-of-fold ROC and precision-recall curves for the action gate and compact students.

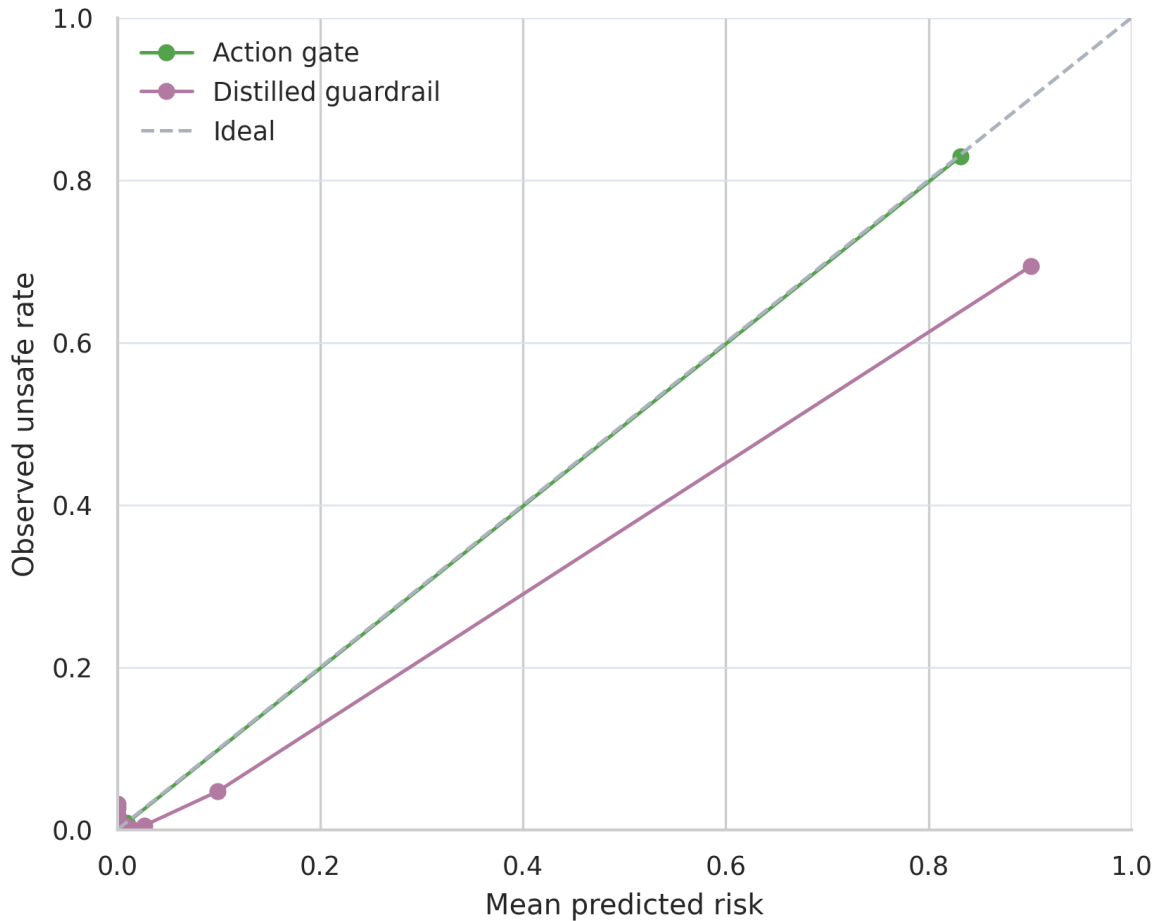


Figure 5. Equal-mass reliability curves for the calibrated action gate and distilled guardrail.

The ablations in Table 10 show that soft-target distillation was not strongest. The hard-label student reached 0.9926 AUROC and 0.9366 AUPRC, with 0.00%–0.95% ASR and 64.18%–65.67% FRR. The numeric-only student reached 0.9749 AUROC and 0.8542 AUPRC, with 2.38%–3.34% ASR and 56.72%–58.21% FRR. Higher action AUROC did not map monotonically to task utility; Figure 6 shows the threshold tension. Although distillation can prioritize compact rules (Di Palo et al., 2024), trajectory composition helped shape the operational result.

Table 10. Distillation, feature, and provenance ablations under fail-closed replay.

Method	Evidence	ASR % [CI or bounds]	Attacked utility % [CI or bounds]	Clean utility % [CI or bounds]	Safe utility % [point or bounds]	False refusal % [point or bounds]	Mean attacked calls
Distilled guardrail	Fail-closed trace replay	5.56–6.52	11.92–15.58	18.56–24.74	10.97–15.58	71.64–73.13	2.308
Numeric-only student	Fail-closed trace replay	2.38–3.34	20.19–23.85	28.87–35.05	19.71–24.32	56.72–58.21	2.448
Hard-label student	Fail-closed trace replay	0.00–0.95	11.13–14.79	24.74–30.93	10.97–15.58	64.18–65.67	2.571
Self-verification without provenance	Fail-closed trace replay	19.55–20.51	36.57–40.22	63.92–70.10	28.30–32.91	5.97–7.46	3.437

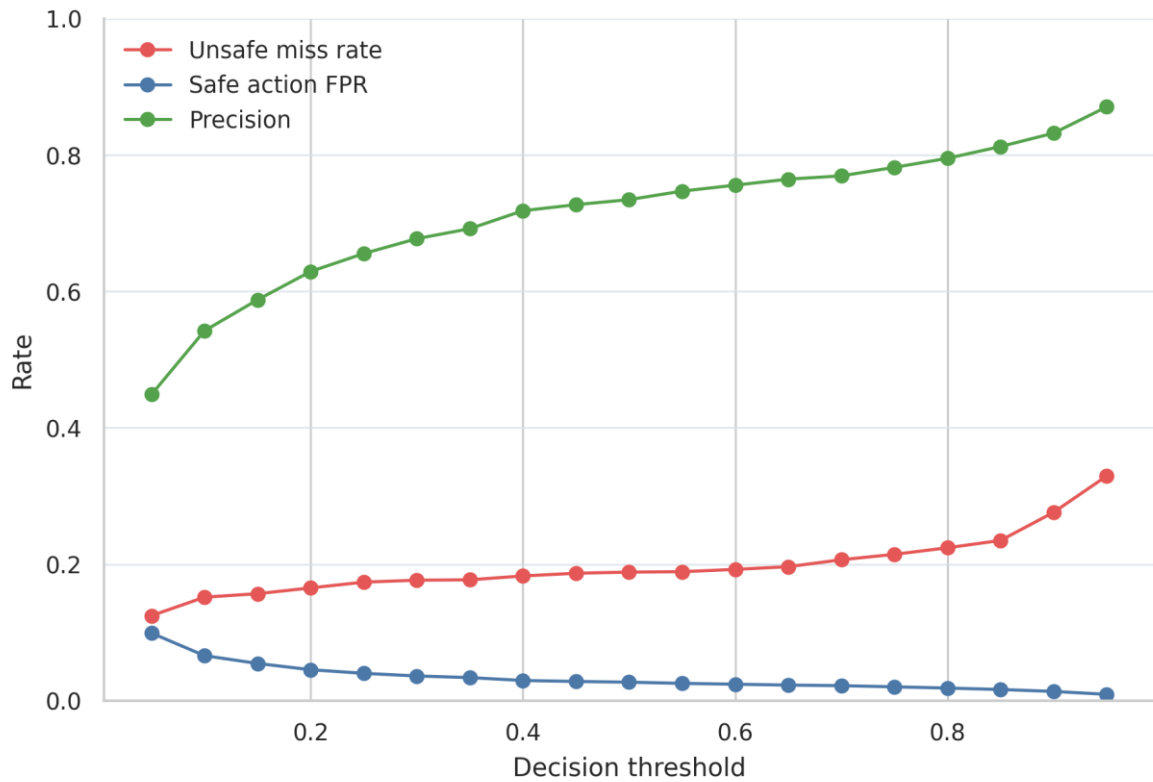


Figure 6. Distilled-guardrail threshold trade-off among unsafe misses, safe-action false positives, and precision.

The mismatch between action AUROC and trajectory utility follows from imbalance, thresholding, and fail-closed composition: one false positive can terminate an otherwise successful episode. Probability calibration, cost-sensitive learning, and selective refusal are therefore relevant evaluation tools even when developed for finance or recruitment (Y. Chen et al., 2024; Jin et al., 2024a; Jin et al., 2024b; Jin, 2025a; Xin, 2025c; Xin, 2026b). Their domain metrics are not compared with the present results.

Efficiency was large but host-specific (Table 11): across five fold models, mean serialized size and probability-scoring time on precomputed features fell from 146.3957 KiB and 358.1204 microseconds for the teacher to 3.5156 KiB and 1.3110 microseconds for the student. Compactness did not restore task utility. Figure 7 shows 875 safe blocked Travel actions, versus 124 in Banking, 273 in Slack, and 210 in Workspace.

Table 11. Teacher–student discrimination, calibration, mean serialized size, and host-specific mean probability-scoring time over five fold models.

Model	Serialized KiB	Inference time, μ s/action	AUROC	AUPRC	Brier	ECE	Teacher probability MAE	Teacher decision agreement %
Action-gate teacher	146.3957	358.1204	0.9971	0.9814	0.0083	0.0009	Reference	Reference
Distilled guardrail	3.5156	1.3110	0.9009	0.7955	0.0353	0.0379	0.0500	92.7218
Hard-label student	3.5674	1.1577	0.9926	0.9366	0.0249	0.0234	0.0286	98.0910
Numeric-only student	1.2031	0.1744	0.9749	0.8542	0.0256	0.0140	0.0382	96.1581

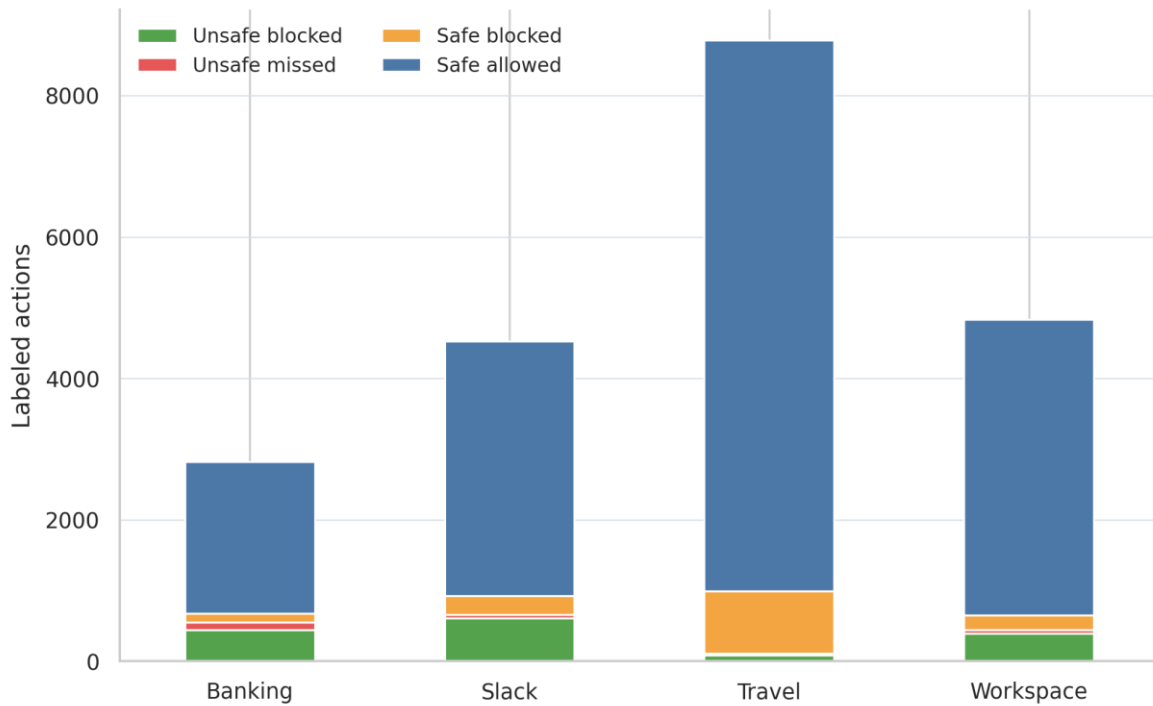


Figure 7. Distilled-guardrail action outcomes by suite across all labeled out-of-fold actions.

Compression is consequently valuable only when the distinctions needed for authorization survive. Lightweight intrusion detection, edge-oriented quality distillation, and hybrid-cloud LLM deployment each frame efficiency as one deployment axis rather than a substitute for application-level validation (Lu & Zhou, 2024; B. Zhou, Wang, et al., 2025; Xin, 2025b). The latency and model-size values above remain specific to this experiment and host.

4.6 Continual Learning and Paired Evidence

Continual performance was volatile (Table 12 and Figure 8). Seen-family unsafe recall rose from 31.34% in round 4 to 92.25% in round 8, then fell to 23.34% and 21.71% in rounds 9–10. Peak-to-current seen-recall loss reached 70.54 points despite the 512-action reservoir. Next-family recall peaked at 84.85% in round 8 before falling to 21.21%; safe-action FPR remained 0.64%–1.37% in rounds 4–10. The collapse therefore reflected lost unsafe recall, not threshold relaxation.

Table 12. Ten-round continual red-team stream with a 512-action rehearsal reservoir.

Round	New attack family	Training actions	Reservoir actions	Seen-family unsafe recall %	Next-family unsafe recall %	Clean/safe action FPR %	No-rehearsal recall %	Peak-to-current seen-recall loss, pp
1	direct	1786	0	0 unsafe test actions	0.00	1.37	0 unsafe test actions	0.00
2	ignore_previous	1802	279	0.00	0 unsafe test actions	7.68	0.00	0.00
3	injecagent	1797	311	0.00	12.31	3.08	50.00	0.00
4	important_instructions	2778	347	31.34	34.38	0.64	31.34	0.00
5	important_instructions_no_user_name	2133	512	77.86	75.00	1.01	84.73	0.00
6	important_instructions_no_model_name	2173	512	60.00	61.90	0.82	94.87	17.86
7	important_instructions_no_names	2153	512	76.74	69.23	1.25	88.76	1.12
8	important_instructions_wrong_model_name	1979	512	92.25	84.85	1.37	90.14	0.00
9	important_instructions_wrong_user_name	1969	512	23.34	21.21	1.13	25.87	68.91
10	tool_knowledge	2074	512	21.71	End of stream	0.95	74.00	70.54

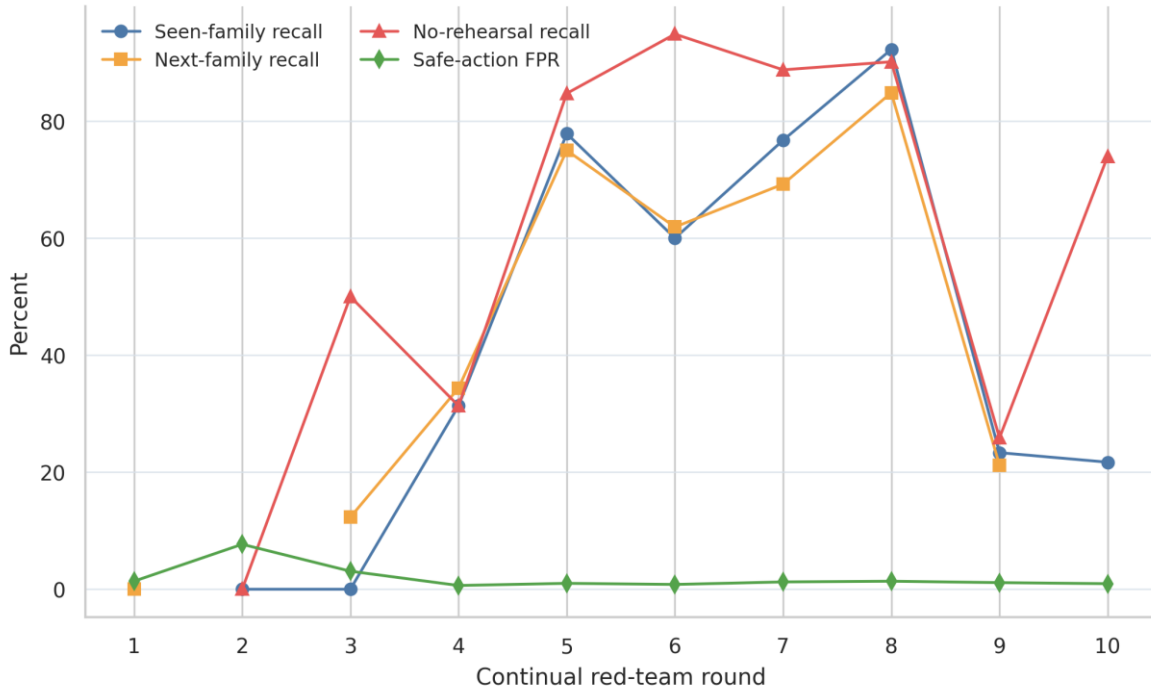


Figure 8. Seen-family retention, next-family transfer, safe-action false positives, and the no-rehearsal comparator.

The late-round collapse makes forgetting a first-class security outcome rather than a training diagnostic. Time-aware memory update and forgetting, continual guardrail updating, cross-cloud transfer, and cold-start forecasting all reinforce the need to evaluate retained behavior after each distributional change (Sun et al., 2023; Zheng et al., 2023; He et al., 2024; He et al., 2025). They do not establish a common failure mechanism; they support the measurement requirement.

Paired tests in Table 13 confirmed the trade-off. Against control, native filtering changed ASR by -40.859 points (Holm-adjusted $p = 8.577 \times 10^{-63}$) and AU by $+6.200$ ($p = .007499$), while its $+3.093$ BU change was nonsignificant ($p = .6776$). Self-verification, action gating, and distillation reduced ASR by 38.523, 48.154, and 42.536 points on exact-match subsets, but reduced AU by 18.317, 30.858, and 38.119 and BU by 16.484, 32.967, and 52.747; all replay contrasts remained significant after Holm adjustment. The evaluated intervention points produced distinct security-utility profiles: native filtering was balanced, action gating was over-conservative, and continual distillation was efficient but unstable.

Table 13. Exact paired outcome tests against the ReAct-style control with Holm correction. Replayed comparisons use scorer-stable pairs.

Metric	Method A	Method B	Paired N	B minus A, pp	A0/B1	A1/B0	Matched odds ratio	Raw p	Holm p
ASR	ReAct-style control	Native tool filter	629	-40.859	13	270	0.050	1.225e-63	8.577e-63
ASR	ReAct-style control	Self-verification	623	-38.523	0	240	0.002	1.132e-72	9.056e-72
ASR	ReAct-style control	Action gate	623	-48.154	0	300	0.002	9.818e-91	9.818e-90
ASR	ReAct-style control	Distilled guardrail	623	-42.536	1	266	0.006	2.260e-78	2.034e-77
Attacked utility	ReAct-style control	Native tool filter	629	6.200	121	82	1.473	7.499e-03	7.499e-03
Attacked utility	ReAct-style control	Self-verification	606	-18.317	5	116	0.047	1.562e-28	9.370e-28
Attacked utility	ReAct-style control	Action gate	606	-30.858	15	202	0.077	5.322e-43	3.725e-42
Attacked utility	ReAct-style control	Distilled guardrail	606	-38.119	2	233	0.011	1.004e-66	1.004e-65
Clean utility	ReAct-style control	Native tool filter	97	3.093	13	10	1.286	6.776e-01	6.776e-01
Clean utility	ReAct-style control	Self-verification	91	-16.484	0	15	0.032	6.104e-05	3.052e-04
Clean utility	ReAct-style control	Action gate	91	-32.967	1	31	0.048	1.537e-08	1.076e-07
Clean utility	ReAct-style control	Distilled guardrail	91	-52.747	0	48	0.010	7.105e-15	7.105e-14

A deployment-facing guardrail should therefore expose why an action was blocked, which evidence influenced it, its calibrated uncertainty, and the side effect being prevented. Evidence cards, capacity-risk briefs, prompt-injection risk cards, and incident cards offer concrete interface patterns for such oversight (Jin, 2025c; G. Liu et al., 2025; Su, Rao, et al., 2026; Nie et al., 2026). The current experiment did not test interface comprehension, so this is a design implication for subsequent human evaluation rather than an empirical claim about the eight figures reported here.

5. Conclusions

The evaluated intervention points produced distinct security–utility profiles. The ReAct-style control completed 50.08% of attacked tasks but allowed 47.69% of injected goals to succeed. Native tool filtering gave the strongest end-to-end balance: attack success fell to 6.84%, attacked utility rose to 56.28%, benign utility reached 72.16%, and mean attacked calls decreased from 4.512 to 3.491. Deterministic self-verification reduced attack success to 9.54% $\square$ 0.49%, but its 31.00% $\square$ 34.66% attacked-utility and 22.39% $\square$ 23.88% false-refusal bounds exposed the cost of conservative action checks.

The calibrated action gate supplied the lowest attack-success bound, 0.00% $\square$ 0.95%, and achieved 0.9971 AUROC and 0.9814 AUPRC at action level. That discrimination did not translate into balanced task behavior: attacked utility was 18.92% $\square$ 22.58%, benign utility was 37.11% $\square$ 43.30%, and false refusal was 46.27% $\square$ 47.76%. The distilled guardrail was 41.6 times smaller and approximately 273 times faster than the teacher, but its AUROC fell to 0.9009, AUPRC to 0.7955, and false refusal rose to 71.64% $\square$ 73.13%. The hard-label student's 0.9926 AUROC and 98.09% teacher-decision agreement further show that temperature-2 soft targets did not improve this compact model.

Continual rehearsal did not solve the weakness. After ten attack families, unsafe-action recall was 21.71%, peak-to-current seen-recall loss reached 70.54 points, and the no-rehearsal comparator retained 74.00% recall. Thus, the 512-action reservoir did not preserve earlier-family unsafe recall under the fixed update order. Overall, native tool restriction delivered the best measured operating point, while learned gating traded excessive utility for security. Effective deployment requires action authorization, task completion, false refusal, calibration, cost, and continual retention to be optimized together.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers

Artificial Intelligence (AI) Use Disclosure: The authors declare that no artificial intelligence tools were used in the preparation of this manuscript.

References

- [1] Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, Z., Fredrikson, M., Gal, Y., & Davies, X. (2025). AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=AC5n7xHuR1>
- [2] Bai, J., Chen, S., Zheng, D., & Kuo, M.-J. (2026). Interpretable attack-chain stage detection from AWS CloudTrail event sequences via linear models and HMM smoothing. *Informatics, Electrical and Electronics Engineering (Infotron)*, 6(1), 28–43. <https://doi.org/10.33474/infotron.v6i1.24923>
- [3] Chang, X., Lu, Y., & Zhong, Z. S. (2026). Review-grounded explainable recommendation with faithfulness evaluation on Amazon Reviews. *JECS (Journal of Electrical Engineering and Computer Sciences)*, 11(1), 9–22. <https://doi.org/10.54732/jeecs.v11i1.2>
- [4] Chen, S., He, S., & Sun, E. (2024). Risk-bounded GPU resource oversubscription via conformal demand envelopes in production AI clusters. *Journal of Advanced Computing Systems*, 4(5), 119–134. <https://doi.org/10.69987/JACS.2024.40509>
- [5] Chen, S., Piet, J., Sitawarin, C., & Wagner, D. (2025). StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)* (pp. 2383–2400). USENIX Association. <https://www.usenix.org/conference/usenixsecurity25/presentation/chen-sizhe>
- [6] Chen, S., Zharmagambetov, A., Mahloujifar, S., Chaudhuri, K., Wagner, D., & Guo, C. (2025). SecAlign: Defending against prompt injection with preference optimization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2833–2847). Association for Computing Machinery. <https://doi.org/10.1145/3719027.3744836>
- [7] Chen, Y., & Xu, H. (2026). Trust-calibrated multilingual RAG for humanitarian information platforms: Empirical evaluation on OMoS-QA for migration information access. *International Journal of Graphic Design*, 4(1), 141–164. <https://doi.org/10.51903/ijgd.v4i1.3552>
- [8] Chen, Y., Zhang, Y., & Sherman, M. (2024). Going concern and bankruptcy prediction under extreme class imbalance: Cost-sensitive learning, resampling, and focal loss with explainable financial-ratio portraits. *Journal of Advanced Computing Systems*, 4(4), 80–96. <https://doi.org/10.69987/JACS.2024.40407>
- [9] Chen, Y., Zhang, Y., Chau, D., & Sherman, M. (2023). Credit card default risk tiering with probability calibration and uncertainty-driven rejection: A reproducible study on the UCI Credit Card Clients dataset. *Journal of Advanced Computing Systems*, 3(4), 31–47. <https://doi.org/10.69987/JACS.2023.30403>
- [10] Chen, Y., Zhou, S., & Lin, E. (2025). Accounting-aware evidence retrieval for institutional due diligence of tokenized trade receivable RWA. *Journal of Technology Informatics and Engineering*, 4(3). <https://doi.org/10.51903/jtie.v4i3.542>
- [11] Debenedetti, E., Shumailov, I., Fan, T., Hayes, J., Carlini, N., Fabian, D., Kern, C., Shi, C., Terzis, A., & Tramèr, F. (2025). *Defeating prompt injections by design*. arXiv. <https://doi.org/10.48550/arXiv.2503.18813>
- [12] Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., & Tramèr, F. (2024). AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *The Thirty-Eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://openreview.net/forum?id=m1YYAQjO3w>
- [13] Di Palo, F., Singhi, P., & Fadlallah, B. H. (2024). Performance-guided LLM knowledge distillation for efficient text classification at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 3675–3687). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.emnlp-main.215>
- [14] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security* (pp. 79–90). Association for Computing Machinery. <https://doi.org/10.1145/3605764.3623985>
- [15] Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., & Dziri, N. (2024). WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems* (Vol. 37, pp. 8093–8131). <https://doi.org/10.52202/079017-0261>
- [16] He, S., Chang, X., & Sun, E. (2024). Cross-cloud transfer learning for AI training capacity forecasting under workload and topology distribution shift. *Journal of Advanced Computing Systems*, 4(1), 100–120. <https://doi.org/10.69987/JACS.2024.40108>
- [17] He, S., Li, C., & Rao, H. (2025). Few-shot cold-start workload forecasting for new AI inference tenants with time-series foundation models. *Journal of Technology Informatics and Engineering*, 4(1), 306–324. <https://doi.org/10.51903/jtie.v4i1.546>
- [18] He, S., Nie, J., & Li, C. (2026). Power-aware inventory planning for AI infrastructure using job-level forecasting and LLM workload explanations. *Journal of Technology Informatics and Engineering*, 5(1). <https://doi.org/10.51903/jtie.v5i1.548>
- [19] He, S., Tu, H., & Liu, I. (2023). Safe PD capacity forecasting with time-series foundation models and calibrated uncertainty for heterogeneous GPU clusters. *Journal of Advanced Computing Systems*, 3(4), 48–66. <https://doi.org/10.69987/JACS.2023.30404>

- [20] Hines, K., Lopez, G., Hall, M., Zarfati, F., Zunger, Y., & Kiciman, E. (2024). *Defending against indirect prompt injection attacks with spotlighting*. arXiv. <https://doi.org/10.48550/arXiv.2403.14720>
- [21] Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., & Khabsa, M. (2023). *Llama Guard: LLM-based input-output safeguard for human-AI conversations*. arXiv. <https://doi.org/10.48550/arXiv.2312.06674>
- [22] Jia, F., Wu, T., Qin, X., & Squicciarini, A. (2025). The Task Shield: Enforcing task alignment to defend against indirect prompt injection in LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 29680–29697). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2025.acl-long.1435>
- [23] Jin, J. (2025a). Calibrated resume-job matching for trustworthy LLM-assisted recruiter screening: Pairwise matching, probability calibration, and selective refusal on two public recruitment datasets. *Journal of Technology Informatics and Engineering*, 4(3), 625–648. <https://doi.org/10.51903/jtie.v4i3.529>
- [24] Jin, J. (2025b). Evidence-chain reliable RAG: Hallucination detection, source attribution, and deterministic provenance explanations. *Journal of Technology Informatics and Engineering*, 4(2). <https://doi.org/10.51903/jtie.v4i2.535>
- [25] Jin, J. (2025c). LLM-style evidence cards for scientific search interfaces: A UI/UX design framework for retrieval transparency, ranking trust, and visual evidence hierarchy. *International Journal of Graphic Design*, 3(2), 397–414. <https://doi.org/10.51903/ijgd.v3i2.3698>
- [26] Jin, J., Huang, T., & Lu, S. (2024a). Cost-sensitive learning, simulated PU learning, and one-class autoencoding for extreme-imbalance credit card fraud detection. *Journal of Advanced Computing Systems*, 4(6), 64–73. <https://doi.org/10.69987/JACS.2024.40605>
- [27] Jin, J., Huang, T., & Lu, S. (2024b). A model-risk-friendly probability of default workflow: Calibration, distribution-free uncertainty quantification, and SHAP explanations on the UCI Credit Card Default dataset. *Journal of Advanced Computing Systems*, 4(6), 74–85. <https://doi.org/10.69987/JACS.2024.40606>
- [28] Kuo, M.-J., Zheng, D., & Hires, J. (2025). Federated topic-preference learning for knowledge-grounded chat with differential privacy. *Journal of Technology Informatics and Engineering*, 4(2), 385–401. <https://doi.org/10.51903/jtie.v4i2.502>
- [29] Li, C., Bai, J., & Wang, S. (2024). Evidence-chain reliable RAG: Word-level hallucination detection, source attribution, and provenance explanation for LLM applications. *Journal of Advanced Computing Systems*, 4(2), 76–92. <https://doi.org/10.69987/JACS.2024.40207>
- [30] Li, C., Liu, G., & Zhao, Z. (2026). Cost-aware LLM-style routing for AIOps log analysis: Log parsing, anomaly detection, fault diagnosis, and incident summarization on LogEval task files. *Journal of Technology Informatics and Engineering*, 5(2), 91–103. <https://doi.org/10.51903/jtie.v5i2.538>
- [31] Li, C., Zhou, B., & Gao, K. (2025). Risk-calibrated patient-facing AI safety cards: A UI/UX benchmark for explainable medical AI response interfaces. *International Journal of Graphic Design*, 3(2). <https://doi.org/10.51903/ijgd.v3i2.3709>
- [32] Li, H., Liu, X., Zhang, N., & Xiao, C. (2025). PiGuard: Prompt injection guardrail via mitigating overdefense for free. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 30420–30437). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2025.acl-long.1468>
- [33] Li, J., & Zhou, A. (2026). Multi-regulation RAG for AI product counsel: A legal governance framework for cross-border digital commerces. *Rule of Law Studies Journal*, 2(2), 105–123. <https://doi.org/10.64780/rolsj.v2i2.225>
- [34] Li, Y. (2024). Findable then explainable: Retrieval-summary integration for code intelligence on a lightweight CodeSearchNet subset. *Journal of Advanced Computing Systems*, 4(7), 65–82. <https://doi.org/10.69987/JACS.2024.40706>
- [35] Li, Y., & Lu, S. (2025). Language-guided feature selection for DDoS and intrusion detection on CICIDS2017. *Journal of Technology Informatics and Engineering*, 4(1), 284–305. <https://doi.org/10.51903/jtie.v4i1.531>
- [36] Li, Z., Zhang, K., & Wong, A. (2026). Numerical-reasoning guardrails for a quant research assistant: A compact reproducible benchmark using SEC and FRED data. *Journal of Technology Informatics and Engineering*, 5(2), 75–90. <https://doi.org/10.51903/jtie.v5i2.541>
- [37] Li, Z., Zhou, S., & Zhou, Z. (2025). Financial risk dashboard design for institutional RWA investors: Visual hierarchy, chart comprehension, and explainability in FinChart-Bench. *International Journal of Graphic Design*, 3(1). <https://doi.org/10.51903/ijgd.v3i1.3715>
- [38] Liu, G., He, S., & Liu, I. (2023). LLM-augmented multi-source root cause attribution for CPU and network faults in microservices. *Journal of Advanced Computing Systems*, 3(6), 39–57. <https://doi.org/10.69987/JACS.2023.30604>
- [39] Liu, G., He, S., & Wong, H. (2025). LLM-compatible visual brief cards for AI infrastructure capacity dashboards: A UI/UX framework for turning forecast risk into graphic design decisions. *International Journal of Graphic Design*, 3(1). <https://doi.org/10.51903/ijgd.v3i1.3723>
- [40] Liu, G., Li, C., & Zhang, E. (2024). OpsLLM for cloud incident triage: Bilingual RAG-based root cause analysis and alert summarization for AI infrastructure operations. *Journal of Advanced Computing Systems*, 4(4), 97–111. <https://doi.org/10.69987/JACS.2024.40408>
- [41] Liu, Y., Deng, G., Li, Y., Wang, K., Wang, Z., Wang, X., Zhang, T., Liu, Y., Wang, H., Zheng, Y., & Liu, Y. (2023). *Prompt injection attack against LLM-integrated applications*. arXiv. <https://doi.org/10.48550/arXiv.2306.05499>
- [42] Liu, Y., Jia, Y., Geng, R., Jia, J., & Gong, N. Z. (2024). Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium (USENIX Security 24)* (pp. 1831–1847). USENIX Association. <https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupe>
- [43] Lu, S., & Zhou, D. (2024). TinyLLM-assisted intrusion detection for real-time IoT networks. *Journal of Advanced Computing Systems*, 4(8), 72–87. <https://doi.org/10.69987/JACS.2024.40809>
- [44] Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakhaee, E., Li, N., Basart, S., Li, B., Forsyth, D., & Hendrycks, D. (2024). HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. *Proceedings of Machine Learning Research*, 235, 35181–35224. <https://proceedings.mlr.press/v235/mazeika24a.html>
- [45] Meng, S., Chen, J., & Zheng, I. (2026). LLM-inspired offline reranking for financial search: Query rewriting, hybrid retrieval, and listwise relevance ranking on FiQA. *Journal of Technology Informatics and Engineering*, 5(1), 361–378. <https://doi.org/10.51903/jtie.v5i1.537>
- [46] Mi, G., Ye, T., & Wood, D. (2025). A lightweight medical foundation model for cross-modal multi-task pretraining and parameter-efficient few-shot transfer on MedMNIST. *Journal of Technology Informatics and Engineering*, 4(3), 572–589. <https://doi.org/10.51903/jtie.v4i3.492>
- [47] Mu, J., Ye, T., & Patel, P. (2025). Offline counterfactual evaluation for advertising and recommendation slot policies: A reproducible study on the Open Bandit Dataset (Small). *Journal of Technology Informatics and Engineering*, 4(3), 521–543. <https://doi.org/10.51903/jtie.v4i3.500>

- [48] Nie, J., & Zheng, D. (2024). Noisy-neighbor-aware VM degradation risk modeling with unsupervised residual fusion. *Journal of Advanced Computing Systems*, 4(4), 112–123. <https://doi.org/10.69987/JACS.2024.40409>
- [49] Nie, J., Liu, G., Li, C., & Zou, T. (2026). Evidence-constrained incident visualization cards for distributed cloud logs: A UI/UX framework for turning Hadoop, OpenStack, and ZooKeeper logs into actionable SRE design interfaces. *International Journal of Graphic Design*, 4(1), 179–185. <https://doi.org/10.51903/ijgd.v4i1.3703>
- [50] Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., & Irving, G. (2022). Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 3419–3448). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.emnlp-main.225>
- [51] Perez, F., & Ribeiro, I. (2022). *Ignore previous prompt: Attack techniques for language models*. arXiv. <https://doi.org/10.48550/arXiv.2211.09527>
- [52] Piet, J., Alrashed, M., Sitawarin, C., Chen, S., Wei, Z., Sun, E., Alomair, B., & Wagner, D. A. (2024). Jatmo: Prompt injection defense by task-specific finetuning. In J. García-Alfaro, R. Kozik, M. Choraś, S. K. Katsikas, & G. Rios (Eds.), *Computer security—ESORICS 2024* (Lecture Notes in Computer Science, Vol. 14982, pp. 105–124). Springer. https://doi.org/10.1007/978-3-031-70879-4_6
- [53] Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C. J., & Hashimoto, T. B. (2024). Identifying the risks of LM agents with an LM-emulated sandbox. In *The Twelfth International Conference on Learning Representations*. <https://openreview.net/forum?id=GEcwtMk1uA>
- [54] Su, W., Chen, S., & Qian, E. (2026). Narrative-aware scientific claim verification agent with evidence ranking for ClimateCheck. *Journal of Technology Informatics and Engineering*, 5(1), 327–340. <https://doi.org/10.51903/jtie.v5i1.549>
- [55] Su, W., Chen, S., & Zhao, C. (2025). Budgeted multi-hop retrieval agent for compositional question answering: A retrieval-policy evaluation on the official MultiHop-RAG benchmark. *Journal of Technology Informatics and Engineering*, 4(3). <https://doi.org/10.51903/jtie.v4i3.543>
- [56] Su, W., Rao, H., & Ma, E. (2026). Privacy and data-integrity risk cards for LLM agents: A UI/UX design framework for secure human oversight under prompt-injection attacks. *International Journal of Graphic Design*, 4(1), 186–191. <https://doi.org/10.51903/ijgd.v4i1.3699>
- [57] Sun, X., Lu, Y., & Chen, J. (2023). Controllable long-term user memory for multi-session dialogue: Confidence-gated writing, time-aware retrieval-augmented generation, and update/forgetting. *Journal of Advanced Computing Systems*, 3(8), 9–24. <https://doi.org/10.69987/JACS.2023.30802>
- [58] Sun, X., Zhong, Z. S., & Wu, Q. (2026). Retrieval-grounded HDFS log anomaly detection and deterministic failure narrative generation. *Journal of Computational Systems and Applications*, 3(1), 15–30. <https://doi.org/10.64229/j6d7fr94>
- [59] Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., & Beutel, A. (2024). *The instruction hierarchy: Training LLMs to prioritize privileged instructions*. arXiv. <https://doi.org/10.48550/arXiv.2404.13208>
- [60] Wang, B., He, Y., Shui, Z., Xin, Q., & Lei, H. (2024). Predictive optimization of DDoS attack mitigation in distributed systems using machine learning. *Applied and Computational Engineering*, 64(1), 89–94. <https://doi.org/10.54254/2755-2721/64/20241350>
- [61] Wang, C., Wen, Z., Zhang, R., Xu, P., & Jiang, Y. (2025). GPU memory requirement prediction for deep learning task based on bidirectional gated recurrent unit optimization Transformer. In *2025 5th International Conference on Artificial Intelligence, Virtual Reality and Visualization (AIVRV)*. IEEE. <https://doi.org/10.1109/AIVRV67401.2025.11350369>
- [62] Wu, Q., Mi, G., & Wood, D. (2025). Calibration-light subject-independent motor imagery BCI via self-supervised pretraining and Conformer. *Journal of Technology Informatics and Engineering*, 4(1), 239–262. <https://doi.org/10.51903/jtie.v5i1.493>
- [63] Xin, Q. (2025a). Explaining OpenStack failure-injection log anomalies with retrieved normal prototypes. *International Journal of Emerging Information Science and Technology*, 6(2), 125–146. <https://doi.org/10.18196/eist.v6i2.31232>
- [64] Xin, Q. (2025b). Hybrid cloud architecture for efficient and cost-effective large language model deployment. *Journal of Information Systems and Informatics*, 7(3), 2182–2195. <https://doi.org/10.51519/journalisi.v7i3.1170>
- [65] Xin, Q. (2025c). Uncertainty-aware late fusion for 3D perception (confidence calibration + fusion rule learning). *Journal of Technology Informatics and Engineering*, 4(1), 215–238. <https://doi.org/10.51903/jtie.v4i1.485>
- [66] Xin, Q. (2026a). Auditable automated essay scoring and formative feedback: A rubric-grounded pipeline for secondary and higher education. *Journal of Applied Artificial Intelligence in Education*, 2(1), 1–9. <https://doi.org/10.66053/jaaie.v2i1.348>
- [67] Xin, Q. (2026b). Explainable and fair credit risk scoring with counterfactual explanations: A reproducible evaluation on the German Credit dataset (HELOC-motivated). *Journal of Information and Technology*, 14(2), 215–231. <https://doi.org/10.32664/j-intech.v14i02.2228>
- [68] Xin, Q. (2026c). Host-based intrusion detection with system call sequences: Window localization and forensic narratives. *Aviation Electronics, Information Technology, Telecommunications, Electricals, and Controls (AVITEC)*, 8(2), 325–334. <https://doi.org/10.28989/avitec.v8i2.3973>
- [69] Xin, Q. (2026d). LiDAR-camera object-level fusion for multi-target tracking using JPDA and EKF: A reproducible empirical study on a PandaSet-parameterised five-sequence dataset. *Journal of Technology Informatics and Engineering*, 5(1), 54–76. <https://doi.org/10.51903/jtie.v5i1.486>
- [70] Xin, Q. (2026e). Log anomaly detection with conformal alert control and evidence-grounded incident ticket generation. *Aviation Electronics, Information Technology, Telecommunications, Electricals, and Controls (AVITEC)*, 8(2), 247–264. <https://doi.org/10.28989/avitec.v8i2.3974>
- [71] Xin, Q. (2026f). Probabilistic bike-sharing demand forecasting under changing weather and seasonal regimes with transformer-based models. *Transport Findings*. <https://doi.org/10.32866/001c.157499>
- [72] Xin, Q. (2026g). Self-supervised log anomaly detection with LogBERT-style transformers: Full empirical evaluation on a reproducible SynHDFS benchmark. *JEECS (Journal of Electrical Engineering and Computer Sciences)*, 11(1), 23–35. <https://doi.org/10.54732/jeeecs.v11i1.3>
- [73] Xin, Q., Xu, Z., Guo, L., Zhao, F., & Wu, B. (2024). IoT traffic classification and anomaly detection method based on deep autoencoders. *Applied and Computational Engineering*, 69(1), 64–70. <https://doi.org/10.54254/2755-2721/69/20241511>
- [74] Xu, H., Chen, Y., & Med, A. (2025). Automatic detection and explanation of dark patterns from interface microcopy: Empirical comparison of BERT-style encoders, RoBERTa-style encoders, and LLM-style decoders on the ec-darkpattern dataset. *Journal of Technology Informatics and Engineering*, 4(3), 590–612. <https://doi.org/10.51903/jtie.v4i3.491>
- [75] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=WE_vluYUL-X

- [76] Ye, T., Chang, X., & Zhong, E. (2025). Uncertainty-aware breast ultrasound explanation cards: A visual communication framework for image-based AI diagnostic support using BreastMNIST_224. *International Journal of Graphic Design*, 3(2). <https://doi.org/10.51903/ijgd.v3i2.3701>
- [77] Ye, T., Mu, J., & Hunter, J. (2026). Off-policy evaluation and conservative policy selection for slot-level dynamic bidding and ranking on the Open Bandit Dataset (Small). *Journal of Technology Informatics and Engineering*, 5(1), 178–199. <https://doi.org/10.51903/jtie.v5i1.503>
- [78] Yi, J., Xie, Y., Zhu, B., Kiciman, E., Sun, G., Xie, X., & Wu, F. (2025). Benchmarking and defending against indirect prompt injection attacks on large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1* (pp. 1809–1820). Association for Computing Machinery. <https://doi.org/10.1145/3690624.3709179>
- [79] Zhan, Q., Fang, R., Panchal, H. S., & Kang, D. (2025). Adaptive attacks break defenses against indirect prompt injection attacks on LLM agents. In *Findings of the Association for Computational Linguistics: NAACL 2025* (pp. 7116–7132). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2025.findings-naacl.395>
- [80] Zhan, Q., Liang, Z., Ying, Z., & Kang, D. (2024). InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024* (pp. 10471–10506). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.findings-acl.624>
- [81] Zhang, B., Rao, H., & Zhao, D. (2024). Evidence-grounded RAG for cloud-native DevOps: Hallucination-resistant AIOps question answering over private operations documents. *Journal of Advanced Computing Systems*, 4(3), 109–125. <https://doi.org/10.69987/JACS.2024.40308>
- [82] Zhang, B., Sun, X., Liu, G., & Zhou, B. (2026). LLM-style DevOps copilot for cloud-native troubleshooting: Retrieval-augmented runbook generation and command-safety evaluation. *Journal of Technology Informatics and Engineering*, 5(2), 104–118. <https://doi.org/10.51903/jtie.v5i2.534>
- [83] Zhang, L., Ma, R., & Greg, P. (2025). Digital-twin dispatching for urban mobility via spatio-temporal transformers and offline reinforcement learning. *Journal of Technology Informatics and Engineering*, 4(2), 337–363. <https://doi.org/10.51903/jtie.v4i2.501>
- [84] Zhang, R., Wen, Z., Wang, C., Tang, C., Xu, P., & Jiang, Y. (2025). Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms. *arXiv*. <https://doi.org/10.48550/arXiv.2511.19481>
- [85] Zhang, Y., & Zhang, H. (2025a). A therapist-facing session copilot for live counseling support: Reasoning-guided retrieval and ranking from multi-turn counseling dialogues. *Journal of Technology Informatics and Engineering*, 4(2), 464–486. <https://doi.org/10.51903/jtie.v4i2.547>
- [86] Zhang, Y., & Zhang, H. (2025b). Visualizing the right counseling support: Evidence-linked recommendation cards for explainable mental health intake interfaces. *International Journal of Graphic Design*, 3(1). <https://doi.org/10.51903/ijgd.v3i1.3722>
- [87] Zhang, Y., & Zhou, Z. (2026). Strategy-aware therapist imitation for emotional support dialogues: A reproducible ESConv study for LLM response control. *Advances in Educational Technology and Psychology*, 10(2), 92–97. <https://doi.org/10.23977/aetp.2026.100213>
- [88] Zhao, S., Bai, J., & Roberson, D. (2025). Multi-horizon GPU demand forecasting with workload semantics and operational risk curves: An empirical study on Alibaba Clusterdata GPU Trace. *Journal of Technology Informatics and Engineering*, 4(3), 544–571. <https://doi.org/10.51903/jtie.v4i3.498>
- [89] Zhao, S., Ren, Y., & Chang, X. (2026). Profit-aware spot GPU admission control with cost-sensitive loss and evidence-grounded policy memos for AI workload supply-demand matching. *Journal of Technology Informatics and Engineering*, 5(2), 45–59. <https://doi.org/10.51903/jtie.v5i2.545>
- [90] Zheng, D., & Li, C. (2024). Behavior-level jailbreak resistance via multi-stage refusal + utility preservation. *Journal of Advanced Computing Systems*, 4(1), 83–99. <https://doi.org/10.69987/JACS.2024.40107>
- [91] Zheng, D., Li, C., & Davidson, H. (2023). Continual red-teaming for in-the-wild jailbreaks via online guardrail updates and guardrail distillation. *Journal of Advanced Computing Systems*, 3(2), 35–49. <https://doi.org/10.69987/JACS.2023.30203>
- [92] Zheng, D., Zhang, B., & Geibel, J. (2024). VerifySafe: Toxicity-safe agent responses under adversarial prompts with evidence-based self-verification. *Journal of Advanced Computing Systems*, 4(1), 67–82. <https://doi.org/10.69987/JACS.2024.40106>
- [93] Zhong, Z. S., Chen, J., Zhong, E., & Sun, X. (2025). Evidence-calibrated RAG for unanswerable question answering: Retrieval coverage, abstention calibration, and hallucination-proxy analysis on SQuAD 2.0. *Journal of Technology Informatics and Engineering*, 4(2), 502–520. <https://doi.org/10.51903/jtie.v4i2.536>
- [94] Zhong, Z. S., Li, C., & Rao, H. (2026). Trajectory reliability prediction for generalist AI agents: Tool-use failure analysis and success forecasting on ZClawBench. *Journal of Technology Informatics and Engineering*, 5(1). <https://doi.org/10.51903/jtie.v5i1.539>
- [95] Zhong, Z. S., Pan, X., & Lei, Q. (2025). Bridging domains with approximately shared features. *Proceedings of Machine Learning Research*, 258, 559–567. <https://proceedings.mlr.press/v258/zhong25a.html>
- [96] Zhong, Z. S., Wu, Q., & Mi, G. (2025). Uncertainty-aware medical image explanation cards: LLM-generated visual explanations for AI-assisted radiology interfaces. *International Journal of Graphic Design*, 3(2), 415–436. <https://doi.org/10.51903/ijgd.v3i2.3616>
- [97] Zhou, B., Li, C., & Liu, L. (2025). Risk-calibrated patient-facing AI safety cards: A UI/UX design framework for rubric-based medical risk communication. *International Journal of Graphic Design*, 3(2). <https://doi.org/10.51903/ijgd.v3i2.3696>
- [98] Zhou, B., Wang, H., & Chang, X. (2025). Distilling VMAF into an edge-deployable quality predictor: A pilot shot-level proxy with LLM-ready quality tokens. *Journal of Technology Informatics and Engineering*, 4(2), 447–463. <https://doi.org/10.51903/jtie.v4i2.522>
- [99] Zhou, H., & Zhang, K. (2025). News-based uncertainty and macro-market fusion for VIX direction forecasting: Evidence from 2015–2024 FRED panel. *Journal of Technology Informatics and Engineering*, 4(2), 487–501. <https://doi.org/10.51903/jtie.v4i2.540>
- [100] Zhu, K., Yang, X., Wang, J., Guo, W., & Wang, W. Y. (2025). MELON: Provable defense against indirect prompt injection attacks in AI agents. *Proceedings of Machine Learning Research*, 267, 8031–80329. <https://proceedings.mlr.press/v267/zhu25z.html>