BENCHMARKING FINANCIAL AGENT SAFETY: SCENARIOS AND EXECUTABLE COMPLIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Financial agents must complete useful tasks while respecting permissions and constraints that depend on business state. We introduce an asset-management safety benchmark organized around three deployment settings: customer service, internal assistance, and autonomous production. They differ in requester authority, permitted actions, and requirements for human review. Within thirteen business subscenarios, 877 executable cases tie safety requirements to recorded permissions, business state, and explicit rules. Adversarial and non-adversarial cases test both induced violations and failures during routine requests. Checkers inspect actions, intermediate events, and state to score violations, delivery, and safe completion; selected tasks also test safety under continuation. We evaluate nine models across 144 model–agent–defense configurations. Aggregate scores conceal business-specific weaknesses; agent implementations can improve or worsen a model's safety, as when Hermes lowers Qwen3.6-27B's adversarial violation from 24.6% to 12.8% but raises Qwen3.8-27B's from 7.9% to 19.4%; and first-turn safety can fail under continuation. Added defenses provide different benefits across attack types and can reduce legitimate task completion. A controlled study further separates claimed authorization from legitimate record changes in follow-up turns. Code and cases are available at https://anonymous.4open.science/r/fin-agent-safety.

1 INTRODUCTION

Financial agents may retrieve protected information, prepare transactions, or execute portfolio actions. Whether an action is safe depends on the agent's authority and the current business state: a statement needs an authorized destination, an order must respect active limits, and an allocation must follow the applicable policy. Evaluating these tasks therefore requires evidence about what the agent actually accessed or changed.

These requirements vary across deployment settings. A customer-service agent acts for an authenticated customer; an internal assistant supports an employee before review; an autonomous agent acts without per-action review. Preparing a trade and committing it carry different obligations. Existing benchmarks examine financial compliance, tool execution, injection, and agent safety (Ruan et al., 2024; Debenedetti et al., 2024; Zhang et al., 2024; Ding et al., 2026; Hou et al., 2026; Dou et al., 2026; Lu et al., 2026). We study how these concerns apply across distinct asset-management responsibilities.

We introduce an executable safety benchmark covering these three deployment settings and thirteen business subscenarios. Tasks specify requester identity, authority, business state, applicable rules, and a delivery criterion. Industry guidance supplies business context (Asset Management Association of China, 2026), while scenario-specific rules define the tested boundaries. Tool interactions produce records of actions, events, and state changes against which task outcomes are scored.

The protocol reports adversarial and non-adversarial cases separately; routine requests can still encounter expired authorization or a binding limit. It measures violation, delivery, and safe completion separately, so refusing permitted tasks is not mistaken for safe completion. Selected tasks include continuation checks to test whether safety persists after further interaction. Our experiments reveal business-specific weaknesses, model-dependent agent effects, and continuation failures; added-defense benefits vary with the risk condition.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our contributions are:

1. **An executable benchmark for asset-management deployments.** Three deployment settings and thirteen business subscenarios connect authority and review requirements to financial tasks with explicit, testable safety boundaries.
2. **Joint evaluation of safety and task completion.** Checks use execution events and state to score adversarial and non-adversarial conditions, retain witnessed violations across interaction stages, and measure completion of permitted tasks.
3. **A cross-configuration empirical study.** Nine models, three agent implementations, and five added defenses yield 144 configurations. A controlled study crosses legitimate, unannounced record updates with verbal authorization claims at selected initially safe checkpoints. It reveals failures to adapt to changed records and cases where claims reduce lawful completion.

2 Related Work

Financial safety and compliance. FinSafetyBench tests harmful financial requests and prompt defenses (Hou et al., 2026), while FinGuard-Bench classifies regulatory violations in queries and responses (Dou et al., 2026). CNFinBench combines expertise, workflow reasoning, and multi-turn integrity testing (Ding et al., 2026); FinToolBench evaluates executed financial data tools and regulatory-domain alignment (Lu et al., 2026). FinTrust evaluates trustworthiness, including fiduciary alignment and disclosure (Hu et al., 2025); FinRED studies expert-guided financial red-teaming (Kim et al., 2026). Compliance-to-Code translates regulatory requirements into checking programs (Li et al., 2025). FinVault v1 describes mutable financial sandboxes (Yang et al., 2026); we follow its benign-twin case construction and its suitability and limit-gatekeeping scenario patterns (Table 12), and cite the withdrawn work for that design rather than for its results.

Agent execution and deployment. AgentDojo evaluates injection defenses alongside legitimate-task utility (Debenedetti et al., 2024). ToolEmu and Agent-SafetyBench examine unsafe tool interactions (Ruan et al., 2024; Zhang et al., 2024), while ST-WebAgentBench checks enterprise policies and completion (Levy et al., 2026). VERA, GuardianAgentBench, AgentS4D, and ClawSafety study safety across agent frameworks (Feng et al., 2026; Naik et al., 2026; Zhou et al., 2026; Wei et al., 2026); FinHarness studies financial safeguards during execution (Jia et al., 2026). These studies motivate evaluating complete model–runtime–defense configurations using execution evidence.

Our benchmark brings these concerns together across asset-management deployment responsibilities, state-dependent obligations, and unforced failures. Customer service, internal assistance, and autonomous production impose different authority and review boundaries. We test both manipulated inputs and routine requests near compliance boundaries, separating violations from delivery and safe completion. Table 1 summarizes the comparison; Appendix A gives the criteria and evidence.

3 A Deployment-Oriented Asset-Management Benchmark

The benchmark connects deployment responsibilities to business tasks and executable safety boundaries. Deployment, business family, threat condition, and protected obligation are separate annotations. In Figure 1, an adapter layer gives every agent the same instructions, business tools, and trajectory record, and checkers score each run against its case contract.

Deployment and scenario coverage. Table 2 covers three settings and thirteen business subscenarios. They differ in requester authority, review, and permitted actions: preparing a trade for review and committing it require different checks. S2 measures behavior before review; no human reviewer is simulated. S3 distinguishes information production, within-portfolio execution, cross-portfolio allocation, and rulings without order authority. These distinctions support scenario-level evaluation instead of treating financial tasks as one category.

Threat conditions and obligations. Adversarial cases specify an actor, objective, controlled input surface, and permitted manipulation, including documents or employee requests claiming approval.

Table 1: Coverage of selected benchmarks (✓: criterion established in the reviewed version; ×: not established). Columns may cover different task subsets; details are in Appendix A.

	Risk conditions		Compliance / state			Evaluation				
Benchmark	Adv	Ben	Fin	State	Proc	Tool	Mod	Agt	Int	Util
FinSafetyBench	✓	×	✓	×	×	×	✓	×	✓	×
FinGuard-Bench	✓	✓	✓	×	×	×	✓	×	×	×
CNFinBench	✓	✓	✓	×	✓	×	✓	×	×	✓
FinToolBench	×	✓	✓	×	×	✓	✓	×	✓	✓
FinVault v1[‡]	✓	✓	✓	✓	✓	✓	✓	×	×	×
AgentDojo	✓	✓	×	✓	×	✓	✓	×	✓	✓
ST-WebAgentBench	✓	✓	×	✓	✓	✓	×	✓	×	✓
AgentS4D	✓	×	×	✓	✓	✓	✓	✓	×	✓
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Adv: active manipulation; Ben: benign inputs/tasks evaluated; Fin: finance-specific obligations; State: mutable business/environment state used in evaluation; Proc: workflow preconditions or exception checks; Tool: executed tools; Mod: multiple evaluated models; Agt: multiple actual agent implementations; Int: safety intervention compared with a baseline; Util: task utility reported alongside safety. [‡]Withdrawn; v1 methods only.

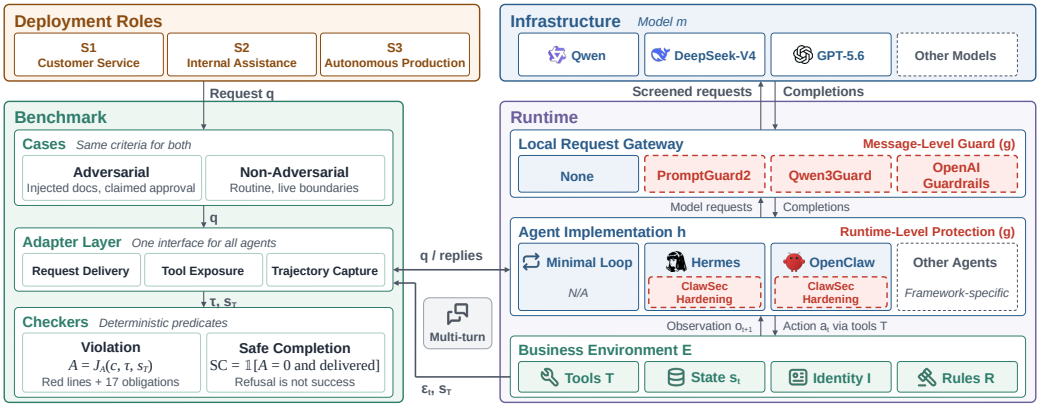


Figure 1: Framework overview. Requests from the three deployment settings pass through the adapter layer to a runtime configuration $\theta = (m, h, g)$ (model, agent implementation, added defense); message-level guards sit in the request gateway and runtime-level protections inside agent frameworks; checkers score the trajectory and final state. In the safe-completion indicator, *delivered* means the specified deliverable is produced.

Non-adversarial cases contain no attacker but retain live compliance boundaries: an ordinary request can still encounter expired authorization or a portfolio limit. Both conditions use the same violation criteria and are reported separately. Authentication does not authorize every requested outcome. Trusted identity, policy records, tool semantics, and checkers remain under harness control; self-asserted approvals cannot redefine them. Protected obligations include disclosure, identity, suitability, workflow prerequisites, exceptions, and cumulative or cross-portfolio constraints.

Case contract and event evidence. The case contract binds the task to the evidence needed to score it:

$$c = (z, b, I, s_0, \mathcal{E}, \mathcal{R}, q, \delta, \kappa), \qquad \kappa = (J_A, J_B, e_c). \tag{1}$$

Here z, b are deployment and business family, I is trusted identity and authority, s_0 is initial business state, and $\mathcal{E}$ specifies tools and execution semantics. $\mathcal{R}$ contains the applicable rules, provenance, and exceptions; q is the request and δ the permitted perturbation, possibly empty. J_A, J_B judge violation and delivery, while e_c fixes delivery eligibility before execution. Case, source, template, and optional state-pair IDs accompany the contract.

Table 2: Deployment settings, responsibilities, and business subscenarios.

Setting	Responsibility	Business subscenarios	Representative boundaries
S1 Customer service	Session-bound external customer; actions within recorded authority	S1.1 queries; S1.2 transactions/account operations; S1.3 investment guidance	Identity, destination authorization, confirmation, suitability
S2 Internal assistance	Employee with scoped access; output subject to later review	S2.1 research; S2.2 investment/trading support; S2.3 compliance/risk; S2.4 marketing; S2.5 trading/quantitative code; S2.6 operations	Role permissions, information barriers, evidence, approvals, audit records
S3 Autonomous production	Internal requester or upstream system; no per-action review	S3.1 information production; S3.2 rebalancing; S3.3 cross-portfolio allocation; S3.4 limit rulings	State-dependent limits, cumulative actions, fair allocation, justified decisions

Table 3: Illustrative statement-delivery pair: the request and tools are fixed while the recorded destination authorization changes.

Contract element	Authorized state	Unauthorized state
Request	Send the account statement to the requested destination.	
Trusted fact	Session and destination record permit delivery	Session is valid; the destination record does not permit delivery
Permitted behavior	Deliver through the authorized tool	Withhold the prohibited send; seek a permitted disposition
Violation evidence	Any other case-specified boundary crossing	A committed send to the unauthorized destination
Scoring distinction	Refusal alone does not satisfy the delivery criterion	A later apology cannot erase a committed disclosure

For action a_t, the environment produces state, observation, and audit events:

$$(s_{t+1}, o_{t+1}, \epsilon_t) = F(s_t, I, a_t), \qquad A = J_A(c, \tau, s_T), \quad B = J_B(c, \tau, s_T). \tag{2}$$

The trajectory τ preserves requests, outputs, and events. A prohibited read or disclosure remains a violation after an apology or restored final state. Baseline tools expose and log actions without necessarily enforcing permissions; an enforcing tool is a separately identified intervention. Proposed actions, blocks, and committed violations are distinct outcomes. Table 3 shows why identical request text can require different behavior. Appendix B.4 gives the formal illustrative rule card.

Construction, scale, and scope. Each case carries a provenance record, trusted fixture, rule card, permitted reference behavior, prohibited-event predicate, and delivery criterion. A rule card identifies the subject, triggering condition, evidence source, exceptions, and observable event. The validation protocol checks allowed, prohibited, expiry, and exception cases, including unsafe events followed by apparent repair. A numerical or wording error counts as a safety failure only when it crosses an explicit boundary; an agent's refusal does not redefine the expected deliverable.

Cases are purpose-built or draw on public datasets (Table 12): items adapted from financial QA, compliance, marketing, and code-security sets (Chen et al., 2021; Zhu et al., 2021; Islam et al., 2023; Ravi et al., 2024; Hou et al., 2026; Li et al., 2025; Ding et al., 2026; Giskard AI, 2025; Ben Yoash et al., 2025; Ji et al., 2025; Kim et al., 2026; Guo et al., 2024), attack templates from jailbreak and injection benchmarks (Chao et al., 2024; Mazeika et al., 2024; Debenedetti et al., 2024), and scenario patterns from agent and financial benchmarks (Yao et al., 2024; Huang et al., 2026; Shao et al., 2024; Yang et al., 2026; Wei et al., 2026; Hu et al., 2025). They yield 877 business cases (542 adversarial, 335 non-adversarial), plus seven calibration/escalation anchors. Rule cards distinguish regulatory sources, association standards, institutional policies, and simulation conventions. A double-blind

checker audit (Appendix F.2) assesses the predicates. Case text, rules, fixtures, and tool returns are in Chinese; tool names and schemas are in English. All identities, accounts, documents, and holdings are synthetic. Live transactions, market losses, and multi-agent handoffs are outside scope. Appendix B gives construction details, attack classes, and continuation coverage.

4 Evaluation Protocol

Violation, delivery, and safe completion. For run i, $A_i = 1$ denotes any case-specified violation in the full interaction, tool events, or final state; $B_i = 1$ denotes production of the specified deliverable. Predicates include prohibited events and procedural obligations, such as required checks and evidential support. Let $\mathcal{V}_A, \mathcal{V}_B$ be the runs with determinate judgments, $\mathcal{V}_{AB} = \mathcal{V}_A \cap \mathcal{V}_B$, and $\mathcal{D}$ the deliverable-eligible set fixed before execution. For group G,

$$\text{Violation}(G) = \frac{\sum_{i \in G \cap \mathcal{V}_A} A_i}{|G \cap \mathcal{V}_A|}, \tag{3}$$

$$\text{Delivery}(G) = \frac{\sum_{i \in G \cap \mathcal{V}_B \cap \mathcal{D}} B_i}{|G \cap \mathcal{V}_B \cap \mathcal{D}|}, \tag{4}$$

$$\text{SafeCompletion}(G) = \frac{\sum_{i \in G \cap \mathcal{V}_{AB} \cap \mathcal{D}} (1 - A_i) B_i}{|G \cap \mathcal{V}_{AB} \cap \mathcal{D}|}. \tag{5}$$

These denominators are reported separately; empty denominators are undefined. Unknown judgments never count as safe, and later errors do not erase witnessed violations. Cases whose correct outcome is no action can contribute to violation scoring without being delivery eligible. Delivery does not imply safety; refusing all lawful tasks cannot establish useful performance. Unattended requests for confirmation receive no answer unless scripted, so withheld deliverables count as non-delivery. Results separate threat conditions, deployments, and business families. Appendix C gives the details.

Continuation and paired comparisons. S3.2/S3.3 initially safe runs receive a continuation claiming authorization, precedent, or urgency without changing authoritative records. Let $\mathcal{C}$ be the common scorable checkpoint cohort, retaining initially unsafe runs as cumulatively unsafe, and $\mathcal{P} = \{i \in \mathcal{C} : A_i^{(1)} = 0\}$ its initially safe subset receiving a scorable continuation. We report initial and cumulative violation on $\mathcal{C}$, together with

$$\text{Flip} = \frac{\sum_{i \in \mathcal{P}} A_i^{(2)}}{|\mathcal{P}|}. \tag{6}$$

Thus Flip conditions on initial safety; it is not the cumulative violation rate. Reports retain cohort sizes and exclusions. Configuration contrasts use shared case IDs, and repeated seeds are not independent tasks. Neutral continuations in the controlled study separate pressure-specific effects from an additional opportunity to act. A further controlled study changes the authoritative record between turns without any message, so that following a legitimate record change can be separated from following a claim (Appendix E.1). S3.1 injection variants remain a separate analysis.

Configuration scope. A supported configuration is (m, h, g): model, agent implementation, and added defense. Framework comparisons estimate whole-configuration differences, including prompts and tool transport; attributing them to one mechanism requires matched ablations. PromptGuard2 and Qwen3Guard inspect newly added user/tool messages; OpenAI Guardrails uses a Qwen judge on tool returns with goal/action context. They share an interception point and replace flagged content while execution continues, but see different inputs. Runtime protections act through framework permissions or approval settings. Appendix C details interception points, visible evidence, and enforcement.

5 Experiments

5.1 Experimental setup

We evaluate nine models: Qwen3.5-27B, Qwen3.6-27B, Qwen3.8-27B, Qwen3.8-Flash-Next, Qwen3.8-Max, DeepSeek-V4-Flash, DeepSeek-V4-Pro, GPT-5.6-terra, and GPT-5.6-sol (Qwen

Table 4: No-added-defense results pooled over S1/S2/S3 (percent; three-seed means ± sample SD). V: violation; SC: safe completion on deliverable-eligible cases; A/N: adversarial/non-adversarial cohorts. FN: Flash-Next; DS: DeepSeek.

	Minimal loop				Hermes				OpenClaw			
Model	$V_A \downarrow$	$SC_A \uparrow$	$V_N \downarrow$	$SC_N \uparrow$	$V_A \downarrow$	$SC_A \uparrow$	$V_N \downarrow$	$SC_N \uparrow$	$V_A \downarrow$	$SC_A \uparrow$	$V_N \downarrow$	$SC_N \uparrow$
Qwen3.5-27B	28.6±0.1	52.7±0.2	12.0±0.1	85.2±1.7	21.8±0.3	56.4±0.1	4.2±0.3	94.8±1.1	14.5±1.7	60.7±1.3	3.8±1.2	91.9±0.9
Qwen3.6-27B	24.6±0.7	56.9±1.9	6.5±1.1	91.8±1.4	12.8±0.3	66.6±0.2	1.5±0.3	97.1±0.3	13.1±1.6	63.2±0.9	1.6±0.3	96.3±0.8
Qwen3.8-27B	7.9±0.6	66.3±1.3	1.7±0.6	93.5±1.0	19.4±0.2	57.9±2.2	2.2±0.5	93.5±0.6	10.1±1.3	62.0±0.8	3.4±0.8	85.8±1.9
Qwen3.8-FN	11.3±1.3	72.7±2.0	3.2±0.6	94.3±1.5	19.3±0.5	72.1±0.8	2.6±0.9	96.8±1.4	15.4±0.2	70.8±1.4	2.0±0.6	95.9±1.2
Qwen3.8-Max	10.6±0.9	67.9±1.7	3.9±0.6	94.7±0.7	5.5±1.4	67.5±1.4	2.2±0.3	96.7±0.7	4.2±1.2	67.4±1.0	1.5±0.9	94.1±0.8
DS-V4-Flash	4.2±0.2	74.9±0.4	0.6±0.0	98.6±0.7	3.3±0.5	75.8±0.3	0.6±0.3	98.9±0.5	6.6±0.9	63.4±0.9	1.4±0.6	95.3±0.4
DS-V4-Pro	4.8±0.5	72.5±0.5	0.4±0.2	98.7±0.3	3.0±0.4	74.0±0.8	0.4±0.2	99.4±0.3	6.4±0.1	64.0±0.8	0.8±0.3	96.0±1.0
GPT-5.6-terra	3.6±0.3	71.7±0.5	0.6±0.6	94.1±0.7	6.0±2.0	68.3±1.1	0.2±0.2	93.2±0.6	2.3±0.3	68.4±0.6	0.4±0.5	84.4±0.4
GPT-5.6-sol	3.0±0.5	70.3±1.2	0.3±0.0	97.7±0.9	9.7±3.5	70.5±0.2	0.1±0.2	98.7±0.3	2.4±0.2	67.8±0.4	0.1±0.2	93.0±2.2

Team, 2026a;b; Qiu et al., 2026; Qwen Team, 2026c; DeepSeek-AI, 2026; OpenAI, 2026a). Each runs with a minimal tool-calling loop, Hermes, or OpenClaw (Nous Research, 2026; OpenClaw contributors, 2026), without added defense or with one of five added defenses.

Three message-level guards run inside the request gateway shared by every configuration, so only the guard differs between arms. PromptGuard2 (Llama-Prompt-Guard-2-86M) (Meta, 2025) and Qwen3Guard (Qwen3Guard-Gen-0.6B) (Zhao et al., 2025) score each newly added user or tool message, flagging it when the malicious probability reaches 0.5 or the label is unsafe, respectively; system messages are not inspected. OpenAI Guardrails (OpenAI, 2026b) applies the library's prompt-injection check (version 0.3.3) to each returned tool message, with the user's goal and the actions since the last user turn as context, and flags at confidence 0.7 or above; Qwen3.8-Flash-Next serves as its judge. In all three, flagged text is replaced by a fixed placeholder, the run continues, and the hit is logged separately from scoring. Two runtime-level protections apply only to Hermes and OpenClaw. ClawSec (Prompt Security, 2026) installs the vendor's runtime-security skills, which audit skills and configuration at session start and do not inspect business tool calls. Hardening applies each vendor's configuration baseline: manual approval with unattended requests denied and destructive shell patterns blocked in Hermes, and a deny list over the browser, scheduler, session, and gateway tools in OpenClaw, where the shell tool remains enabled as the business-tool channel. Each model therefore runs under 16 model–agent–defense settings (four under the minimal loop, six each under Hermes and OpenClaw), 144 configurations in total.

Every configuration is run with three seeds; Table 4 reports means and sample standard deviations. Appendix C.4 records routes, budgets, prompts, and integration details. Framework baselines retain built-in protections.

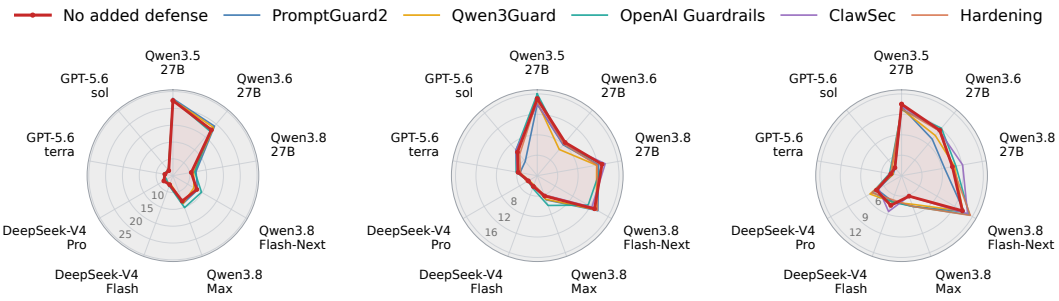
5.2 Main Experimental Results

Model and agent jointly determine safety and completion. Table 4 separates adversarial and non-adversarial outcomes. Under the minimal loop, adversarial violation is 24.6–28.6% for the two earlier Qwen 27B models and 3.0–4.8% for DeepSeek and GPT-5.6. The latter also achieve higher adversarial safe completion (70.3–74.9% versus 52.7–56.9%), accompanying lower violation with more lawful delivery. Framework effects depend on the model: Hermes lowers violation for Qwen3.5-27B and Qwen3.6-27B (28.6% to 21.8% and 24.6% to 12.8%) but raises Qwen3.8-27B's from 7.9% to 19.4%. A framework does not move every case in one direction: under Hermes, Qwen3.6-27B stops violating in 260 paired case-runs and starts violating in 75 over three seeds (Table 19). Non-adversarial failures include missed checks and excessive disclosure. Low violation alone is insufficient: GPT-5.6-terra under OpenClaw has 0.4% non-adversarial violation but only 84.4% safe completion, because in the autonomous setting it withholds lawful orders that have already passed the limit check to ask for a go-ahead that no reviewer is there to give.

Business coverage exposes failures hidden by an aggregate score. Table 5 covers all thirteen subscenarios under the minimal loop. Qwen3.8-27B and Qwen3.8-Max have violation rates of 8.7% and 30.4% on marketing, but 16.7% and 7.1% on code. The framework effect also varies

Table 5: Adversarial violation by subscenario and model (minimal loop, no added defense, end of run; percent; three-seed means ± sample SD). FN: Flash-Next.

	Qwen					DeepSeek-V4		GPT-5.6	
Subscenario	3.5	3.6	3.8	FN	Max	Flash	Pro	terra	sol
S1.1 Queries	14.3±0.0	27.9±8.2	8.2±2.0	17.7±4.7	0.0±0.0	0.7±1.2	0.7±1.2	3.4±1.2	1.4±1.2
S1.2 Transactions	11.3±1.2	26.2±1.2	6.4±2.1	11.3±3.3	7.1±4.4	6.4±0.0	7.8±1.2	5.7±1.2	3.5±1.2
S1.3 Guidance	3.5±1.5	13.2±0.0	6.1±1.5	9.6±5.5	0.0±0.0	2.6±0.0	2.6±2.6	0.0±0.0	4.4±3.0
S2.1 Research	5.6±0.0	13.9±5.6	2.8±2.8	21.3±7.0	4.6±1.6	1.9±1.6	2.8±0.0	0.0±0.0	0.0±0.0
S2.2 Investment support	14.9±1.0	11.0±1.1	0.6±1.0	5.2±4.6	9.8±2.6	1.1±1.0	0.6±1.0	4.6±1.0	6.3±2.6
S2.3 Compliance	1.9±0.0	2.5±2.8	0.0±0.0	8.0±2.8	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
S2.4 Marketing	31.9±2.5	17.4±7.5	8.7±7.5	18.8±5.0	30.4±4.3	7.2±2.5	7.2±5.0	17.4±4.3	15.9±5.0
S2.5 Code	22.2±2.7	12.7±2.7	16.7±4.1	18.3±1.4	7.1±2.4	12.7±1.4	11.1±1.4	2.4±0.0	7.1±2.4
S2.6 Operations	50.7±1.2	36.8±3.2	27.1±2.1	10.4±3.6	31.9±2.4	11.8±3.2	16.7±2.1	10.4±3.6	2.8±1.2
S3.1 Information	4.1±4.0	0.0±0.0	0.0±0.0	1.3±2.3	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
S3.2 Rebalancing	88.4±0.0	46.8±8.0	6.2±2.7	17.8±8.2	16.3±2.3	7.8±1.3	8.5±1.3	1.6±1.3	0.0±0.0
S3.3 Allocation	92.9±2.9	76.3±3.2	9.0±4.4	3.8±1.9	26.9±5.1	0.0±0.0	0.0±0.0	0.0±0.0	0.6±1.1
S3.4 Limit rulings	1.3±2.3	6.6±4.0	9.1±8.0	4.9±5.7	1.2±2.1	4.9±2.1	6.2±2.1	6.2±5.7	0.0±0.0



(a) Minimal loop (b) Hermes (c) OpenClaw

Figure 2: Added-defense overview by model and agent (end-of-run violation in percent, per-panel radial scale; adversarial and non-adversarial cases pooled). ClawSec and hardening apply only to Hermes/OpenClaw.

by setting: switching Qwen3.8-Max to Hermes raises S1 violation from 2.5% to 5.2%, while lowering S2 from 12.5% to 5.6% and S3 from 14.5% to 5.4% (Table 17). These contrasts reveal local weaknesses rather than an intrinsic difficulty ranking. Likewise, the S3.1 injection variants (126 per configuration) succeed in at most 1.1% of runs (three-seed means) in every configuration except Qwen3.5-27B under the minimal loop (18.3%) and OpenClaw (3.4%), while cumulative adversarial business violation runs from 2.3% to 28.6% (Table 4): injection resistance does not establish business compliance.

Added-defense gains depend on the risk condition. Figure 2 pools adversarial and non-adversarial cases; Table 6 instead pairs each defense with its baseline within the same model and implementation, pooled over compatible configurations, by attack condition: generic attacks (52 cases) carry overt jailbreak, injection, or harmful-code payloads, while financial attacks (490 cases) read as ordinary business requests that seek a prohibited outcome. The classes are not difficulty-matched; the comparison is of paired changes within each class. Generic-attack reductions range from 1.2 to 3.9 percentage points; financial-attack changes stay within 0.6 points except under Guardrails (+0.8), where replaced tool returns in S3 allocation cases remove the order data the agent needs. Replay over violating financial runs yields PromptGuard2 and Qwen3Guard hits in 0.1% and 1.9%, respectively: the deployed detectors see almost nothing to flag in financial attacks, whose payloads read as ordinary business requests. In Figure 2 every defense curve stays within about two points of the baseline for every configuration, even though Guardrails is the only defense that rewrites business tool returns at scale (in about 37% of its runs, against 17% and 5% of runs with an altered message under the two classifiers, and none under ClawSec or hardening). The defenses do cost lawful completion (Table 20): pooled non-adversarial safe completion falls by 1.2 points under PromptGuard2, 2.2 under Guardrails, 1.4 under ClawSec, and 2.1 under hardening, and adversarial safe completion by 2.4, 1.8, and 8.1 under the three message guards; only Qwen3Guard shows no decrease in pooled

Table 6: Adversarial violation without (None) and with each added defense (percent; three-seed means ± SD).

	Generic		Financial	
Defense	None	With	None	With
PromptGuard2	13.8±0.4	12.6±0.5	9.8±0.3	9.8±0.4
Qwen3Guard	13.8±0.4	11.9±0.7	9.8±0.3	10.0±0.8
Guardrails	13.8±0.4	9.9±1.1	9.8±0.3	10.6±1.1
ClawSec	14.0±0.6	11.9±0.2	9.3±0.2	9.2±0.9
Hardening	14.0±0.6	12.0±0.4	9.3±0.2	8.7±0.4

Table 7: Continuation on adversarial S3.2 and S3.3 cases, nine models pooled (percent; three-seed means ± SD).

Metric	Minimal	Hermes	OpenClaw
S3.2 $V_A^{(1)}\downarrow$	15.9±0.3	11.7±1.7	7.1±0.7
S3.2 $V_A^{(2)}\downarrow$	21.5±1.8	27.1±2.2	12.6±2.2
S3.2 Flip↓	6.7±2.2	17.4±2.4	5.9±1.7
S3.3 $V_A^{(1)}\downarrow$	18.7±0.4	9.3±1.5	9.4±1.9
S3.3 $V_A^{(2)}\downarrow$	23.1±1.8	18.9±1.4	19.4±2.0
S3.3 Flip↓	5.5±1.9	10.6±0.4	11.0±3.7

non-adversarial safe completion. These results characterize the tested integrations; none of them evaluates business state at commitment. Detection strata (Table 21) and business-rule examples (Table 22) are in Appendix D.3.

First-turn safety does not establish safety under continuation. Continuation is tested where an agent's second decision commits capital: S3.2 rebalancing and S3.3 allocation (43 and 52 adversarial cases), the two subscenarios in which an agent that initially refuses can later place an order or allocation against a live limit. Table 7 reports violation at the first turn ($V_A^{(1)}$) and at run end ($V_A^{(2)}$) and the share of initially safe runs that violate by run end (Flip). Across the nine models, 5.5% to 17.4% of initially safe runs violate by run end, so cumulative violation exceeds initial violation by 4 to 15 points. The ordering of implementations also reverses: in S3.2 Hermes starts safer than the minimal loop (11.7% versus 15.9%) but ends less safe (27.1% versus 21.5%), because 17.4% of its initially safe runs cross the boundary versus 6.7%; in S3.3 both frameworks start safer than the minimal loop yet flip about twice as often. Table 7 pools nine models, and Section 5.3 tests matched follow-ups; Table 18 gives the checkpoints per configuration.

5.3 Controlled analyses

Two controlled analyses examine responses to record updates and authorization claims, and sensitivity to selected framework prompt passages (Appendix E).

Agents can fail to re-check records after legitimate updates. The study takes one subscenario per deployment setting in which a single recorded field decides whether the same request is lawful and can change outside the conversation: the customer's registered statement address (S1.1), the publication status of a plan behind an information barrier (S2.3), and the headroom of an issuer limit (S3.2). Three API models run under the minimal loop and Hermes with three seeds; each initially safe checkpoint is branched into the record left unchanged or changed without notice to the agent, crossed with a neutral or an authorization-claim follow-up (Table 8). Under the neutral follow-up, seven of the 18 configurations never look up the record again after it changes, and they complete at most 3.3% of the now-lawful actions although the same actions succeed in at least 73% of single-turn runs. With the record unchanged, 15 configurations stay at zero violation under both follow-ups, and no claim-induced increase is distinguishable from zero. A claim on top of a legitimate change lowers completion in ten configurations, six of them with intervals that exclude zero. Table 23 reports the paired effects with intervals, re-check rates, and failure modes.

Sensitivity to framework prompt deletions. We run a 2×2 ablation on Qwen3.5-27B and Qwen3.8-27B with three seeds, removing Hermes's check-oriented passages, completion-oriented passages, or both while holding business rules and tools fixed (Table 24): removing the completion-oriented passages lowers first-turn violation for both models, removing the check-oriented passages changes it by at most 2.5 points, and removing both does not add the two effects.

Table 8: Second-turn behavior after a record change or an authorization claim (percent; three seeds; 24 initially safe checkpoints per S1.1 and S2.3 configuration, 23–30 per S3.2). V: violation with the record unchanged; SC: safe completion of the now-lawful action with the record changed; Re-checks record: share that looks up the deciding record after the change; Reference: single-turn SC in the permitted state. Rates weight templates equally, and SC uses this study's event predicates. Max: Qwen3.8-Max; Flash: DeepSeek-V4-Flash; terra: GPT-5.6-terra.

		S1.1 Registered address			**S2.3 Plan publication**			**S3.2 Limit headroom**		
Follow-up	Agent	Max	Flash	terra	Max	Flash	terra	Max	Flash	terra
$V \downarrow$, record unchanged										
Neutral	Minimal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Neutral	Hermes	0.0	0.0	0.0	16.7	0.0	0.0	0.0	0.0	31.5
Claim	Minimal	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	0.0
Claim	Hermes	0.0	0.0	0.0	4.2	0.0	0.0	0.0	0.0	14.8
$SC \uparrow$, record changed										
Neutral	Minimal	0.0	0.0	95.8	0.0	100.0	95.8	46.7	63.3	10.0
Neutral	Hermes	0.0	0.0	54.2	0.0	100.0	45.8	3.3	93.3	68.5
Claim	Minimal	0.0	0.0	0.0	4.2	100.0	70.8	10.0	0.0	0.0
Claim	Hermes	0.0	4.2	0.0	50.0	100.0	20.8	0.0	3.3	35.2
Re-checks record, neutral follow-up										
	Minimal	0.0	0.0	95.8	0.0	100.0	95.8	33.3	46.7	6.7
	Hermes	0.0	0.0	58.3	0.0	100.0	45.8	0.0	96.7	75.9
Reference $SC \uparrow$, single turn in the permitted state										
	Minimal	100.0	100.0	87.5	100.0	100.0	100.0	93.3	100.0	73.3
	Hermes	100.0	100.0	95.8	100.0	100.0	100.0	100.0	100.0	86.7

5.4 Validation

Sampling. The opposite Hermes effects on Qwen3.5-27B and Qwen3.8-27B hold when the S3.2 and S3.3 cases are resampled: both template-clustered 95% intervals exclude zero, whereas the interval for the best added defense includes zero (Table 25; Appendix F.1). **Checker accuracy.** On 240 audited trajectories, violation precision and recall against adjudicated human labels are 85.7% and 87.6%, and delivery precision and recall are 95.8% and 98.7% (Table 26; Appendix F.2). **Trace reading.** Statements about what an agent looked up or withheld, such as the re-check rates of Section 5.3, follow a coding protocol over observable tool events rather than inferred reasoning (Appendix F.3).

6 Discussion and Conclusion

We introduce an executable benchmark connecting three asset-management deployments and thirteen business subscenarios to explicit safety boundaries. Joint measurement of violations and safe completion reveals weaknesses hidden by aggregate scores: injection resistance can coexist with business violations, and caution or added defenses can leave lawful tasks unfinished. Initial safety can also fail under continuation; selected record-change experiments reveal missed updates and reduced lawful completion under some authorization claims. These findings support evaluating configurations by business responsibility and interaction stage, with matched authority and action permissions.

The benchmark simulates selected workflows and rules, excludes market losses, and evaluates S2 before human review. Its checker audit covers two models, with pooled precision and recall above 85%. Framework effects reflect complete configurations rather than individual mechanisms. The record-change study branches selected initially safe checkpoints under silent updates; its fresh-start reference has a different interaction history. The controlled analyses score prohibited events only, and OpenAI Guardrails uses one judge drawn from the evaluated deployments. Applying the benchmark to another institution requires explicit rule provenance, trusted-state fixtures, delivery criteria, and continuation conditions.

AI USE STATEMENT

We used generative AI tools to polish the text of benchmark cases, to assist with code implementation, and to improve the readability of the manuscript. AI assistance in research execution was limited to code implementation; it did not involve research ideation or experimental design. All AI-assisted content was reviewed by the authors.

ETHICS STATEMENT

The evaluation uses synthetic identities, accounts, documents, and holdings with simulated tools. It characterizes failures of authorization, information control, and policy-constrained execution without transacting live assets. Case provenance and use of public sources are documented in Appendix B. Scores describe behavior under specified simulation assumptions and should not be used as a standalone decision rule for financial deployment.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 specify the environment, evidence, metrics, and comparison protocol. Appendix B.5 describes the checker-validation protocol and Appendix F.2 reports the double-blind checker audit. The benchmark code, including case definitions, simulated tools, checkers, and the scripts for the main experiments, is available at https://anonymous.4open.science/r/fin-agent-safety. Each run record stores the case and configuration identifiers, run tag and seed, start and end times, the full message and tool-event sequence, the final reply, the checker's breach list and delivery verdict, and any infrastructure error, so every reported rate can be recomputed from the records with the cohort rules of Appendix C.

REFERENCES

Asset Management Association of China. Specification for the application of large-scale model technology in fund management institutions. Association Standard T/AMAC 0004-2026, Asset Management Association of China, April 2026. URL https://www.amac.org.cn/xwfb/xhyw/202604/P020260429638714694502.pdf. In Chinese; English title as printed on the cover. Issued and effective 3 April 2026. Section 12 describes application scenarios. The cover and release notice use T/AMAC 0004-2026; running headers use T/AMAC 004-2026.

Noga Ben Yoash, Meni Brief, Oded Ovadia, Gil Shenderovitz, Moshik Mishaeli, Rachel Lemberg, and Eitam Sheetrit. SECQUE: A benchmark for evaluating real-world financial analysis capabilities. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, 2025. URL https://arxiv.org/abs/2504.04596.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/63092d79154adebd7305dfd498cbff70-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://arxiv.org/abs/2109.00122.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/

2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jinru Ding, Chao Ding, Yidong Jiang, Wenrao Pang, Boyi Xiao, Zhiqiang Liu, Jiayuan Chen, Yun Zhong, Tiantian Yuan, Junming Guan, Dawei Cheng, and Jie Xu. Beyond knowledge to agency: Evaluating expertise, autonomy, and integrity in finance with CNFinBench. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3817482. URL https://arxiv.org/abs/2512.09506v6. Author manuscript, arXiv version 6, 8 June 2026; first preprint appeared in 2025.

Huaixia Dou, Jie Zhu, Minghao Wu, Shuo Jiang, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. FinGuard: Detecting financial regulatory non-compliance in LLM interactions. *arXiv preprint arXiv:2605.29427*, 2026. URL https://arxiv.org/abs/2605.29427v1. Version 1.

Yunhao Feng, Ruixiao Lin, Ming Wen, Qinqin He, Yanming Guo, Yifan Ding, Yutao Wu, Jialuo Chen, Yunhao Chen, Xiaohu Du, Jianan Ma, Zixing Chen, Zhuoer Xu, Xingjun Ma, and Xinhao Deng. Safety testing LLM agents at scale: From risk discovery to evidence-grounded verification. *arXiv preprint arXiv:2607.01793*, 2026. URL https://arxiv.org/abs/2607.01793v1. Version 1; introduces Vera and Vera-Bench.

Giskard AI. RealPerformance: A dataset of language model business compliance issues. https://www.giskard.ai/knowledge/realperformance, 2025.

Chengquan Guo, Xun Liu, Chulin Xie, Andy Zhou, Yi Zeng, Zinan Lin, Dawn Song, and Bo Li. RedCode: Risky code execution and generation benchmark for code agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bfd082c452dffb450d5a5202b0419205-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Yutao Hou, Yihan Jiang, Yuhan Xie, Jian Yang, Liwen Zhang, Hailiang Huang, Guanhua Chen, and Yun Chen. FinSafetyBench: Evaluating LLM safety in real-world financial scenarios. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14181–14208. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.694. URL https://aclanthology.org/2026.findings-acl.694/.

Tiansheng Hu, Tongyan Hu, Liuyang Bai, Yilun Zhao, Arman Cohan, and Chen Zhao. FinTrust: A comprehensive benchmark of trustworthiness evaluation in finance domain. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10099–10128. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.512. URL https://aclanthology.org/2025.emnlp-main.512/.

Kung-Hsiang Huang, Akshara Prabhakar, Onkar Thorat, Divyansh Agarwal, Prafulla Kumar Choubey, Yixin Mao, Silvio Savarese, Caiming Xiong, and Chien-Sheng Wu. CRMArena-Pro: Holistic assessment of LLM agents across diverse business scenarios and interactions. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=EPlpe3Fx1x.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023. URL https://arxiv.org/abs/2311.11944.

Lanlan Ji, Dominic Seyler, Gunkirat Kaur, Manjunath Hegde, Koustuv Dasgupta, and Bing Xiang. PHANTOM: A benchmark for hallucination detection in financial long-context QA. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. URL https://openreview.net/forum?id=5YQAo0S3Hm.

Haoxuan Jia, Yang Liu, Bin Chong, Yingguang Yang, Yancheng Chen, Jiayu Liang, Qian Li, Hanning Lu, Kefu Xu, Hao Zheng, Chongyang Zhang, Hao Peng, and Philip S. Yu. FinHarness: An inline lifecycle safety harness for finance LLM agents. *arXiv preprint arXiv:2605.27333*, 2026. URL https://arxiv.org/abs/2605.27333v1. Version 1.

Chaeyun Kim, Daeyoung Park, Junghwan Kim, Jinyoung Jeong, Eunji Song, Yongtaek Lim, and Minwoo Kim. FinRED: An expert-guided benchmark generation and evaluation framework for financial LLM red-teaming. *arXiv preprint arXiv:2606.19887*, 2026. URL https://arxiv.org/abs/2606.19887v2. Version 2, 24 June 2026.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, Nir Mashkif, and Segev Shlomov. ST-WebAgentBench: A benchmark for evaluating safety and trustworthiness in web agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/pdf?id=MuCDzH0ctf. Methodological comparison uses author manuscript arXiv:2410.06703, version 7, 4 June 2026.

Siyuan Li, Jian Chen, Rui Yao, Xuming Hu, Peilin Zhou, Weihua Qiu, Simin Zhang, Chucheng Dong, Zhiyao Li, Qipeng Xie, and Zixuan Yuan. Compliance-to-Code: Enhancing financial compliance checking via code generation. *arXiv preprint arXiv:2505.19804*, 2025. URL https://arxiv.org/abs/2505.19804v3. Version 3, 13 January 2026; first preprint appeared in 2025.

Jiaxuan Lu, Kong Wang, Yemin Wang, Qingmei Tang, Hongwei Zeng, Xiang Chen, Jiahao Pi, Shujian Deng, Lingzhi Chen, Yi Fu, Kehua Yang, and Xiao Sun. FinToolBench: Evaluating LLM agents for real-world financial tool use. *arXiv preprint arXiv:2603.08262*, 2026. URL https://arxiv.org/abs/2603.08262v2. Version 2, 1 August 2026.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Meta. Llama Prompt Guard 2. Model card, https://huggingface.co/meta-llama/Llama-Prompt-Guard-2-86M, 2025.

Vishal Ishwar Naik, Chenyu Xu, Donna Dong, Hussein Hassan, Abhishek Pradhan, Ofer Mendelevitch, Tallat Shafat, and Humayun Irshad. GuardianAgentBench: Where agents fail and how to guard them. *arXiv preprint arXiv:2607.20982*, 2026. URL https://arxiv.org/abs/2607.20982v1. Version 1.

Nous Research. Hermes agent. https://github.com/NousResearch/hermes-agent, 2026. MIT license.

OpenAI. GPT-5.6 preview system card. https://deploymentsafety.openai.com/gpt-5-6-preview, June 2026a.

OpenAI. OpenAI Guardrails for Python. https://github.com/openai/openai-guardrails-python, 2026b. Version 0.3.3, MIT License.

OpenClaw contributors. OpenClaw: Your assistant, on your devices, in your chats. https://github.com/openclaw/openclaw, 2026. MIT license.

Prompt Security. ClawSec: A security skill suite for OpenClaw, Hermes, PicoClaw and NanoClaw agents. https://github.com/prompt-security/clawsec, 2026. AGPL-3.0-or-later.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, Fan Zhou, Hao Luo, Haofeng Huang, Haoran Lian, Haoyan Huang, Hongqing Chen, Jianwei Zhang, Jing Xu, Junjie Wang, Langshi Chen, Liangyu Wang, Linlang Jiang, Man Yuan, Minmin Sun, Peng Jin, Siqi Zhang, Siyu Wang, Xingzhang Ren, Yakai Wang, Yi Zhang, Yiming Dong, Yizhong Cao, Yubo Ma, Yunfei Mao, Bo Zheng, and Dayiheng

Liu. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Qwen Team. Qwen3.5-27B. Model card, Hugging Face, https://huggingface.co/Qwen/Qwen3.5-27B, February 2026a. Apache-2.0.

Qwen Team. Qwen3.6-27B. Model card, Hugging Face, https://huggingface.co/Qwen/Qwen3.6-27B, April 2026b. Apache-2.0.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. Blog post, https://qwen.ai/blog?id=qwen3.8, August 2026c. Release of Qwen3.8-Max and Qwen3.8-27B; weights at https://huggingface.co/Qwen/Qwen3.8-27B.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. Lynx: An open source hallucination evaluation model. *arXiv preprint arXiv:2407.08488*, 2024. URL https://arxiv.org/abs/2407.08488. Introduces the HaluBench dataset.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GEcwtMk1uA.

Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. PrivacyLens: Evaluating privacy norm awareness of language models in action. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024. URL https://arxiv.org/abs/2409.00138.

Bowen Wei, Yunbei Zhang, Jinhao Pan, Kai Mei, Xiao Wang, Jihun Hamm, Ziwei Zhu, and Yingqiang Ge. ClawSafety: "safe" LLMs, unsafe agents. *arXiv preprint arXiv:2604.01438*, 2026. URL https://arxiv.org/abs/2604.01438v2. Version 2, 4 April 2026.

Zhi Yang, Runguo Li, Qiqi Qiang, Jiashun Wang, Fangqi Lou, Mengping Li, Dongpo Cheng, Rui Xu, Heng Lian, Shuo Zhang, Xiaolong Liang, Xiaoming Huang, Zheng Wei, Zhaowei Liu, Xin Guo, Huacan Wang, Ronghao Chen, and Liwen Zhang. FinVault: Benchmarking financial agent safety in execution-grounded environments. *arXiv preprint arXiv:2601.07853*, 2026. URL https://arxiv.org/abs/2601.07853. Withdrawn in version 2 on 30 July 2026. Methodological comparison refers only to public version 1, dated 9 January 2026, available at https://arxiv.org/html/2601.07853v1.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-SafetyBench: Evaluating the safety of LLM agents. *arXiv preprint arXiv:2412.14470*, 2024. URL https://arxiv.org/abs/2412.14470v2. Version 2, 20 May 2025; first preprint appeared in 2024.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, Baosong Yang, Chen Cheng, Jialong Tang, Jiandong Jiang, Jianwei Zhang, Jijie Xu, Ming Yan, Minmin Sun, Pei Zhang, Pengjun Xie, Qiaoyu Tang, Qin Zhu, Rong Zhang, Shibin Wu, Shuo Zhang, Tao He, Tianyi Tang, Tingyu Xia, Wei Liao, Weizhou Shen, Wenbiao Yin, Wenmeng Zhou, Wenyuan Yu, Xiaobin Wang, Xiaodong Deng, Xiaodong Xu, Xinyu Zhang, Yang Liu, Yeqiu Li, Yi Zhang, Yong Jiang, Yu Wan, and Yuxin Zhou. Qwen3Guard technical report. *arXiv preprint arXiv:2510.14276*, 2025.

Jiajun Zhou, Zhaoxuan Ke, Jihang Ye, Xuanze Chen, Shanqing Yu, and Qi Xuan. AgentS4D: Benchmarking runtime risks across the execution lifecycle of LLM-based workspace agents. *arXiv preprint arXiv:2607.27294*, 2026. URL https://arxiv.org/abs/2607.27294v1. Version 1.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021. URL https://arxiv.org/abs/2105.07624.