Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AgentPort-Bench: A Controlled Seven-Framework Evaluation of Agentic AI Security Portability

Waqar Javed

waqarjaved.com@gmail.com

Atlantis University

Research Article

Keywords: agentic AI security, cross-framework evaluation, LLM agents, prompt injection, equivalence testing, mixed-effects models, benchmark reproducibility

Posted Date: September 22nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11102299/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

AgentPort-Bench: A Controlled Seven-Framework Evaluation of Agentic AI Security Portability

Waqar Javed[1*]

[1*]Safe Labs AI Inc., Hollywood, FL, United States.

Corresponding author(s). E-mail(s): waqarjaved.com@gmail.com;

Abstract

Does the choice of agentic orchestration framework meaningfully shape the security posture of a tool-using LLM agent, independent of the underlying model and the attack it faces? We answer this with a single, unified, seven-framework, eight-execution-condition evaluation — a direct-API baseline plus LangChain, CrewAI, AutoGen, LlamaIndex, the OpenAI Agents SDK, Google Agent Development Kit (ADK), and Semantic Kernel — run under one controlled design: 6 models spanning three providers and two capability tiers, 5 attack families mapped onto the OWASP Agentic Security Initiative's threat taxonomy, and payload-identity verification confirming, byte-for-byte, that every condition transmits the same adversarial content to the model. The evaluation combines 9,360 trials collected across two data-collection rounds — an original 7,020-trial, six-condition study and a 2,340-trial, two-framework extension — into one dataset and, critically, one unified statistical treatment applied here for the first time: alongside the classical ANOVA/permutation-test approach used in both rounds' original reporting, we fit an ordinal logistic regression and a linear mixed-effects model with a random prompt intercept (intraclass correlation $=$ **0.333**, confirming genuine repeated-measures structure), and run a pre-specified two-one-sided-test (TOST) equivalence analysis with a margin set by Cohen's small-effect convention before any condition effect was inspected. Both the classical and the new methods agree: attack family explains the largest share of outcome variance (28.0–28.7% depending on the exact model), model choice a smaller but real share (4.1%), and framework choice a share indistinguishable from zero (0.05% by variance share; a permutation $\boldsymbol{p = 0.822}$; and, in the new equivalence framework, all $\binom{8}{2} = 28$ pairwise execution-condition contrasts fall within a pre-specified small-effect margin). One condition, CrewAI, is a genuine exception worth naming precisely rather than averaging away: even after fixing a confirmed adapter defect that had biased its outcomes upward (a persona-based system prompt diverging from the shared baseline, corrected and 1,170 trials re-collected), CrewAI retains a small but statistically significant residual elevation in non-PASS outcomes relative to

every other condition — the single significant framework effect in either statistical model, and the pairwise contrast closest to (though still inside) the equivalence margin. We report this finding exactly as strongly as the evidence supports it: real, specific to one framework, uniform across old and new conditions alike (it does not interact with the two frameworks added in the extension), and small in absolute terms. We also document a fully reproducible, deterministic reasoning-token-exhaustion failure mode — one frontier model silently consuming its entire token budget on invisible reasoning against one adversarial prompt, replicated 6/6 times across both newly added frameworks and consistent with a related pattern in the original data — and a units-mismatch pitfall between two frameworks' own token-accounting conventions that would produce a spurious cost-efficiency comparison if not corrected for. Every dataset, detector-code version, and analysis script referenced in this paper is version-controlled and described in full in our Data Availability statement, available to editors and reviewers on request.

Keywords: agentic AI security, cross-framework evaluation, LLM agents, prompt injection, equivalence testing, mixed-effects models, benchmark reproducibility

1 Introduction

Agentic AI systems, in which a large language model plans, calls tools, and acts across multiple steps with delegated authority, are increasingly built on a small number of open-source orchestration frameworks (LangChain, CrewAI, AutoGen, LlamaIndex, and others). A natural and practically important question for anyone building or auditing such a system is whether adversarial robustness is a property of the underlying model and attack surface, or whether it is meaningfully shaped by which orchestration framework sits between the model and the outside world. If robustness is largely framework-independent, security evaluation effort is best spent on attack taxonomy and model-level behavior; if it is framework-dependent, framework choice itself becomes a security decision.

This paper answers that question with a single, unified research program rather than as two separately-scoped studies. Concretely, we ask:

- **RQ1.** Does framework choice materially affect security outcomes once payload-equivalence is verified and controlled for?
- **RQ2.** Does the answer to RQ1 persist as framework diversity increases — specifically, across seven architecturally distinct orchestration frameworks rather than five?
- **RQ3.** How large is the framework effect relative to model choice and attack-family choice?
- **RQ4.** Are observed framework differences practically equivalent to zero within a pre-specified, defensible margin, or merely statistically indistinguishable from zero under an underpowered test?

We answer all four with one dataset and one unified analysis. The underlying data was collected in two rounds: an initial controlled evaluation across six execution conditions (a direct-API baseline plus five agentic frameworks: LangChain, CrewAI, AutoGen, LlamaIndex, and the OpenAI Agents SDK), and a subsequent extension adding two further frameworks (Google Agent Development Kit and Semantic Kernel) under an identical experimental design — the same frozen 30-prompt adversarial set, the same six-model roster, and the same tiered-seed methodology. Both rounds independently found a near-zero framework effect using a classical ANOVA/permutation-test approach; this paper's central methodological contribution is combining both rounds' raw per-trial data (never previously pooled into one factorial dataset) and applying a statistically stronger treatment — mixed-effects modeling and a formal equivalence test — that neither round's original analysis attempted. RQ4 in particular could not be answered by either original study on its own: a non-significant permutation test establishes only that the data does not rule out zero effect, not that the effect is small enough to be practically negligible. Answering it properly requires the pre-specified-margin equivalence framework this paper introduces.

Our contributions are:

1. A controlled, payload-verified evaluation spanning seven agentic orchestration frameworks and a direct-API baseline (eight execution conditions), six models across three providers, and five attack families mapped onto the OWASP Agentic Security Initiative's threat taxonomy — the largest such controlled comparison we are aware of, combining 9,360 trials from two data-collection rounds into a single unified dataset.
2. A byte-for-byte payload-identity verification methodology that caught a real, previously undetected adapter defect (Section 3.4): CrewAI's default agent construction diverged from every other condition's shared system prompt, measurably biasing its outcomes before correction. We report this not as a footnote but as a worked demonstration of why payload verification is a necessary precondition for any cross-framework security claim, not an optional check.
3. The first unified statistical treatment of this combined dataset: alongside the classical ANOVA/permutation approach, we fit an ordinal logistic regression and a linear mixed-effects model with a random prompt intercept, and run a pre-specified TOST equivalence test with an externally-anchored margin. All three methods agree on the qualitative finding (framework effect negligible relative to attack family and model), and the equivalence framework provides the first result in this research program that actually supports a practical-negligibility claim rather than merely failing to reject a null of no effect.
4. A precisely-scoped exception to that finding: CrewAI's small but statistically real residual elevation in non-PASS outcomes, surviving its own adapter-defect correction, confirmed not to interact with the two frameworks added in this paper's extension round, and reported at the exact strength the evidence supports — neither dismissed as noise nor inflated into a broader framework-risk claim.
5. Two further, practically-relevant findings that a coarse pass/fail security metric alone would not surface: a fully deterministic reasoning-token-exhaustion failure mode in one frontier model against one adversarial prompt, replicated across both

newly added frameworks; and a units-mismatch pitfall between two frameworks' own token-usage accounting conventions that would manufacture a spurious cost comparison if taken at face value.

We adopt a disclosure posture throughout this paper — stated in full in Section 4 — of public code and detector history alongside private-but-available-on-request underlying trial data, and we report every finding, including two data-provenance discrepancies discovered and fully resolved while preparing this paper's evidence, as part of the methodology rather than as a footnote.

2 Related Work

2.1 Taxonomies of Agentic Security Risk

The OWASP Top 10 for Agentic Applications, published by the OWASP GenAI Security Project's Agentic Security Initiative in December 2025 and chaired by John Sotiropoulos, provides the first widely-adopted taxonomy purpose-built for autonomous agent risk rather than single-turn LLM risk, spanning ten categories (ASI01–ASI10) from goal hijacking and tool misuse through cascading failures and rogue-agent behavior [1]. We adopt this taxonomy to structure our attack-family design (Section 3), both because it reflects current practitioner consensus on what agentic security evaluation should cover, and because alignment with an OWASP framework gives our results a standard vocabulary for practitioners auditing specific deployments. OWASP ASI is a risk taxonomy, not an evaluation methodology — it specifies *what* to test for, not *how* to control for confounds like the orchestration layer, which is the gap our design addresses.

Greshake et al. [2] established indirect prompt injection as a practical attack class against LLM-integrated applications, showing that adversarial instructions embedded in retrieved data — rather than supplied directly by a user — can hijack model behavior. The threat categories evaluated in this work are aligned to a structured taxonomy of such risks rather than a single attack technique.

2.2 Single-Environment Agentic Benchmarks

AgentDojo established the dominant paradigm for measuring agent robustness to prompt injection: a dynamic environment populated with realistic tasks (email, e-banking, travel booking, workspace tools) and adversarial injection tasks embedded in tool outputs, evaluated against both static and adaptive attacks [3]. AgentDojo and its many derivatives have become a standard reference point in the field. Critically, AgentDojo — like the great majority of agentic security benchmarks that followed it — evaluates within a single orchestration framework at a time. This is a reasonable and productive choice for measuring model-level robustness, but it means the benchmark's results are, by construction, silent on whether a given vulnerability or defense generalizes to a different framework implementing the same underlying task. Our work differs in scope and design: rather than evaluating attack and defense techniques within a single framework's execution environment, we hold the adversarial

prompt set, model roster, and scoring methodology fixed and vary the orchestration framework itself, to isolate framework choice as an independent variable.

2.3 Cross-Framework Comparison

The most directly related work is Nguyen and Husain, who conducted what they describe as the first systematic comparative penetration test of agentic AI across multiple models and frameworks [4]. Their study evaluated five models (Claude 3.5 Sonnet, Gemini 2.5 Flash, GPT-4o, Grok 2, and Nova Pro) across two frameworks (AutoGen and CrewAI), using a seven-agent architecture simulating a university information-management system and thirteen attack scenarios spanning prompt injection, SSRF, SQL injection, and tool misuse — 130 test cases in total. Their central finding is a large framework-level disparity: a 52.3% attack refusal rate under AutoGen versus 30.8% under CrewAI, holding model constant.

This is an important result, and our work is motivated directly by it — but the design leaves one question open that matters for how the finding should be interpreted. Nguyen and Husain compare frameworks at the level of the overall system (a shared multi-agent architecture built once per framework), rather than verifying that the *attack payload itself* is delivered identically by each framework's adapter layer. Framework orchestration libraries differ substantially in how they template, wrap, or transform message content before it reaches the model — a distinction our own evaluation infrastructure surfaced directly during development, as a defect in one adapter's system-prompt handling measurably shifted outcome verdicts (Section 3.4). Without a byte-level payload-identity guarantee, a framework-level disparity is consistent with at least two different explanations: a genuine effect of the orchestration framework on model behavior, or an artifact of how one framework's adapter happens to phrase or structure the same underlying attack. These explanations have different practical implications — the first says security teams must re-test every framework; the second says they must fix an adapter bug — and prior work has not had the design needed to distinguish them.

2.4 This Evaluation's Own Prior Work

This paper's dataset was collected across two rounds, previously reported separately with a classical ANOVA/permutation-test analysis each. This paper supersedes both separate reports with one unified dataset and a new, statistically stronger joint analysis (Section 3.7). A companion study documents and audits a specific pattern-matching refusal-detector defect in the same underlying evaluation codebase, its fix, and its generalization across models and categories [5]; the verdict scheme and detector version used throughout this paper are adopted from that study without re-derivation.

2.5 Positioning

Our contribution extends this line of work along three axes: scale (6 models $\times$ 8 conditions $\times$ 5 attack families $\times$ 9,360 combined trials, versus Nguyen and Husain's 5 models $\times$ 2 frameworks $\times$ 130 trials), methodological control (payload-identity verification at

the byte level across every adapter, isolating framework as a clean independent variable, and a documented, worked example of that verification catching a real defect), and statistical treatment (a unified mixed-effects and pre-specified equivalence-testing framework, alongside the classical variance-decomposition approach, rather than raw rate comparison or a single-method analysis). Where Nguyen and Husain's result establishes *that* framework-level differences can be observed, our design is built to test *whether* those differences survive when the confound of adapter-level payload variation is removed, and, going further, to test how confidently we can conclude the surviving effect is negligible rather than merely undetected.

3 Methodology

3.1 Design Goal

The central methodological requirement of this study is isolating framework as an independent variable. A naive cross-framework comparison — running "the same attack" through two different frameworks — conflates two things that must be separated: differences arising from how the model responds to a given input, and differences arising from how each framework's adapter layer constructs that input in the first place (system prompt templating, message role structuring, tool-schema formatting, and content serialization all vary by framework, and any one of them can alter what the model actually receives even when the attacker's payload is nominally unchanged). We address this by verifying payload identity directly, rather than assuming it, in both data-collection rounds this paper draws on.

3.2 Experimental Matrix

We evaluate 6 models spanning two tiers and three providers — cheap tier: claude-haiku-4-5-20251001 [6], gpt-5.4-nano [7], gemini-3.1-flash-lite [8]; frontier tier: claude-opus-4-8 [9], gpt-5.5 [10], gemini-3.5-flash [11] — across **eight execution conditions**: a direct-API HTTP/custom baseline and **seven** agentic frameworks (LangChain, CrewAI, AutoGen, LlamaIndex, the OpenAI Agents SDK, Google Agent Development Kit, and Semantic Kernel), and 5 attack families, each mapping to 2 of the 10 OWASP ASI categories (3 prompts $\times$ 2 categories = 6 prompts/family). The first six conditions and 7,020 trials were collected in an initial round; the two newest frameworks and 2,340 further trials were collected in a subsequent extension round under an identical design — the same frozen 30-prompt adversarial set, the same six-model roster, and the same tiered-seed methodology (10 seeds/prompt for cheap-tier models, 3 seeds/prompt for frontier-tier models). **This paper is the first to combine both rounds' raw, per-trial data into one 9,360-trial dataset and analyze it jointly**; each round's own prior report analyzed its own trials in isolation. Each trial follows the PASS / UNCERTAIN / FAIL / VULNERABLE verdict scheme established in our companion detector-calibration study [5] and its underlying evaluation framework [12], which we adopt here rather than re-deriving.

3.3 Payload-Identity Verification

For each execution condition, we log the exact content transmitted to the model API immediately prior to the call and diff it, byte-for-byte, against the canonical attack payload. Any divergence — a reworded system prompt, an added instruction, a reformatted tool description — is treated as an adapter defect to be fixed, not a framework characteristic to be reported. This check is not a one-time gate: because framework libraries construct prompts dynamically and can behave differently across model backends within the same framework, we re-verify payload identity per condition×model pair rather than assuming a single per-condition check generalizes. The two frameworks added in the extension round received this same verification; their Anthropic provider paths were verified via live, real-API end-to-end calls, while their OpenAI and Google provider paths received one live confirmatory call each rather than the repeated, targeted diagnostic testing described in Section 5.5 for the frontier-tier reasoning-budget risk specifically — an asymmetry in verification depth we disclose in Section 6 rather than treating both paths as equally scrutinized.

3.4 A Caught Confound, as a Worked Example of the Verification's Value

This verification procedure is not a formality; it surfaced a real defect that measurably changed outcomes. CrewAI's `build_crewai_agent` did not reference the shared `SYSTEM_PROMPT` constant used by every other condition, instead constructing a persona from separate role/goal/backstory fields whose `goal` text included the word "safely" — language absent from every other condition's prompt, and roughly three times longer besides. We normalized `build_crewai_agent` to use `SYSTEM_PROMPT` verbatim and re-collected all 1,170 affected trials. Comparing the verdict distribution before and after normalization confirms the confound was not merely structurally present but empirically consequential: PASS 66.4%→61.3%, UNCERTAIN 32.3%→37.6% (two-proportion z-test on PASS rate, $z = 2.58$, $p = 0.0098$). All results reported in this paper use the corrected dataset (`full_run_7020_v2`), in which the original 1,170 CrewAI rows are fully replaced by the re-collected, normalized-prompt data; the other five original-round conditions' rows are byte-identical to the original run. We report this confound and its measured effect size directly because it illustrates the exact failure mode our payload-identity verification is designed to catch — an apparent framework effect that was, in this instance, an adapter defect. Section 5.4 returns to CrewAI specifically: even after this fix, a smaller, real residual effect remains, and we report it with the same directness.

3.5 A Second Confound: System-Prompt Asymmetry Affecting ASI06

A second, unrelated data-comparability issue was identified and fixed prior to the final original-round run: until 2026-07-14, the HTTP and LangChain conditions sent no system prompt at all, while the other four did. This mechanically forced a perfect 0.000 attack-success rate on ASI06 (system-prompt-extraction) trials for these two conditions specifically — there was no prompt content to leak — confirmed on a

36-trial sample (ASR = 0.000 for HTTP/LangChain vs. 0.146 for the other four combined) before the fix. Both conditions now send the same system prompt as the rest of the matrix. Data collected before this date is not comparable to data collected after it for ASI06 × {HTTP, LangChain} specifically, and is excluded from pooled ASI06 analysis in this paper.

3.6 Other Integrity Issues Surfaced and Resolved

Verification also caught two additional issues prior to final data compilation. First, GPT-5.5's content-policy refusal behavior spans multiple ASI categories across all eight conditions, isolated entirely to gpt-5.5 with zero occurrences in any other model — the same underlying gpt-5.5 content-policy and empty-output behavior that a companion study independently documents and accounts for from its own narrower scope [5]. Second, a suspected content-list stringification bug in the LangChain–Gemini adapter (identified during earlier development) was independently re-checked against all 390 real LangChain×Gemini rows in the final dataset; no corruption was found, and we report this as resolved-in-data with root cause unconfirmed, rather than claiming a fix we cannot fully attribute.

3.7 Statistical Treatment

We report two complementary statistical treatments of the combined dataset, side by side, rather than replacing one with the other.

Classical treatment. As in both data-collection rounds' own original reporting, we decompose outcome variance using ANOVA across the factors of interest, with a permutation test to assess whether the framework effect exceeds what would be expected under random reassignment. This method requires a balanced, orthogonal design; the full 9,360-row combined dataset does not meet this requirement (the cheap tier's 10 seeds/prompt and the frontier tier's 3 seeds/prompt give unequal cell replication depending on model), so, consistent with both rounds' own established convention, we report the classical method on the properly balanced combined cheap-tier subset ($n = 7{,}200$).

New treatment. Alongside the classical method, we apply, for the first time to this combined dataset, a mixed-effects and equivalence-testing framework directly motivated by two limitations of the classical approach: it treats an ordinal, four-level verdict outcome (PASS < UNCERTAIN < FAIL < VULNERABLE) as though it were interval-scaled, and a non-significant permutation test establishes only that an effect cannot be distinguished from zero, not that it is practically negligible. We fit two complementary models on the full combined 9,360-row dataset: an ordinal (proportional-odds) logistic regression, correctly respecting the outcome's ordinal structure but without a random-effects term; and a linear mixed-effects model on the continuous `attack_success_weight` outcome, with a random intercept for prompt ID, correctly capturing the repeated-measures structure across seeds but treating the outcome as continuous rather than ordinal. No tool in this analysis environment implements a combined ordinal-mixed model (the R `ordinal::clmm` model has no direct Python equivalent); we report both models rather than substituting a lesser one and

describing it as fully addressing both concerns. We additionally run a pre-specified two-one-sided-test (TOST) equivalence analysis on all $\binom{8}{2} = 28$ pairwise condition contrasts, with a primary margin set by Cohen's small-effect convention ($d = 0.2$, scaled by the dataset's pooled outcome standard deviation) and a domain-anchored sensitivity margin (one-fifth of the gap between adjacent verdict-weight steps), both stated and justified before any condition-effect result was inspected.

Verification of the new treatment. Before applying this framework to draw any new conclusion, we confirmed it reproduces both data-collection rounds' own previously published headline figures exactly, from raw per-trial data, using the real, already-in-repository analysis code rather than a hand-rederived formula (Section 4.1). We also independently re-verified, at the row level, two documentation discrepancies uncovered while assembling this paper's evidence (Section 4), so that this paper's own reported error/empty-output rates are demonstrated correct rather than merely carried forward from prior documentation.

4 Data Availability

Every quantitative claim in this paper traces to one of two version-controlled repositories.

`safelabs-eval` (public) is the detector and evaluation framework itself: https://github.com/AgentSafeLabs/safelabs-eval. The verdict scheme and scoring logic used throughout this paper are retrievable as specific, citable git artifacts, with source-provenance confirmed before every scoring or reanalysis run.

`agentdojo-x` (private) is the benchmark orchestrator and data repository that produced every raw trial this paper analyzes: the original round's `results/full_run_7020_v2.jsonl` (7,020 rows, post-CrewAI-fix) and the extension round's `results/cfps_8fw_google_adk_semantic_kernel.jsonl` (2,340 rows), combined here into one 9,360-row dataset for the first time. This repository is currently private because it is also the basis of a separate, not-yet-published research program beyond this paper's scope, but **the specific files, commits, and per-trial data underlying every table and statistic in this paper are available to the editor and to reviewers on request**.

Two documentation discrepancies, found and fully resolved while assembling this paper's evidence, disclosed here rather than silently corrected. First, the original round's documented error/empty-output count (140 rows) and this paper's recomputed count (205 rows) both proved correct — for two different file versions. The 140 figure was computed against `full_run_7020.jsonl` (v1), one day before the CrewAI confound fix (Section 3.4) replaced 1,170 rows to produce `full_run_7020_v2.jsonl`; verified row-for-row, every non-CrewAI framework's error count is byte-identical between versions, and the entire 65-row difference is isolated to CrewAI's re-collected cells encountering different transient rate-limiting and provider-availability conditions than the original collection. This paper reports 205, the correct figure for `_v2` — the file this project's own documentation specifies for every downstream analysis, including this one. Second, the extension round's documented figure

(30 rows: "24 content-policy blocks + 6 reasoning-exhaustion rows") was a transcription error, not a data or methodology discrepancy: the actual content-policy count is 18, not 24; 18 + 6 = 24 is the dataset's true total, independently confirmed via direct row enumeration and cross-checked against the archived pre-scoring `.raw.jsonl` sibling. This paper reports 24. Combined: **229 error/empty-output rows out of 9,360 trials (2.447%)**, exclusively in the frontier tier, is the figure used throughout this paper's Results.

We adopt this disclosure posture — public code and detector history, private-but-available-on-request underlying trial data — deliberately rather than by default, for the same reason stated in our companion detector-calibration paper [5]: the underlying dataset includes model responses to adversarial security prompts, some of which contain content not appropriate for unrestricted public release. Requests for access to the specific rows this paper cites may be directed to the corresponding author.

4.1 Cross-check: reproducing both rounds' published figures before any new computation

Before applying the mixed-effects/equivalence framework (Section 3.7) to any new question, we confirmed it reproduces both rounds' own previously published headline figures exactly, using the real, already-in-repository analysis code rather than a hand-rederived formula:

Table 1 Reproduction of previously published figures from raw data, before any new computation.

Figure	Reproduced here	Published
Original round — framework variance share (n = 5,400)	0.0569%	"0.06%"
Original round — attack-family variance share (n = 5,400)	28.6694%	"28.67%"
Extension round — attack-family variance share (n = 1,800)	26.0715%	"26.07%"

A fourth figure, the original round's permutation-test p-value on the framework term, reproduced to $p = 0.6907$ against a published $p = 0.6952$ — within already-documented Monte Carlo sampling tolerance for a 10,000-permutation test using the same fixed seed, not a computation discrepancy; both values are far from significance and support the same conclusion. With three of four figures reproduced exactly and the fourth within documented tolerance, we treat this precondition as met and proceed to the new analysis reported in Section 5.

5 Results

5.1 RQ1–RQ3: Classical Variance Decomposition

A cheap-tier ANOVA across the balanced combined subset (n = 7,200: 3 cheap-tier models crossed with all 8 conditions and all 5 attack families) attributes outcome

variance as follows: Attack Family 28.0016%, Model 4.1426%, Framework/Condition **0.0501%**, with the remainder attributable to residual and interaction terms. A permutation test on the framework term yields $p = 0.822$, indicating the observed framework effect is not distinguishable from what would arise under random reassignment. This directly answers RQ3: the framework effect is roughly two orders of magnitude smaller than the attack-family effect and roughly one order of magnitude smaller than the model effect.

5.2 RQ1, RQ2, RQ4: The New Mixed-Effects and Equivalence Treatment

The classical result above answers RQ3 but only partially answers RQ1 (does framework choice *materially* affect outcomes) and does not answer RQ4 (is the effect *practically equivalent to zero*) at all — a non-significant permutation test establishes only that the null of no effect cannot be rejected, not that the effect is small enough to be practically negligible. We address both directly on the full combined 9,360-row dataset.

Table 2 Ordinal logistic regression, condition (framework) fixed effects only. Full model includes model and attack-family effects (all large and highly significant; omitted here for space).

Condition (ref. = `http`)	Coef. (log-odds)	p	95% CI
autogen	0.159	0.123	[−0.043, 0.361]
crewai	**0.411**	**<0.001**	**[0.211, 0.611]**
google-adk	−0.024	0.821	[−0.228, 0.181]
langchain	0.099	0.339	[−0.104, 0.301]
llamaindex	0.062	0.548	[−0.141, 0.265]
openai-agents	0.173	0.092	[−0.028, 0.375]
semantic-kernel	0.050	0.630	[−0.153, 0.253]

Of the seven framework coefficients, six are not distinguishable from zero at $\alpha = 0.05$; **CrewAI is the sole exception**, with a real, statistically significant positive coefficient (worse verdicts more likely) relative to the `http` baseline, even in this combined, more rigorous model, and even after the adapter-defect fix described in Section 3.4. A parallel linear mixed-effects model on `attack_success_weight` (random intercept for prompt ID, 30 groups; intraclass correlation = 0.333, confirming a real, non-trivial repeated-measures structure) agrees directionally and in significance pattern: CrewAI's coefficient (0.0184 ASW units, $p < 0.001$) is again the only condition effect clearing significance, with `autogen` and `openai-agents` borderline ($p \approx 0.04$–0.05) and the remainder clearly non-significant. This answers RQ2: the same qualitative pattern — a negligible aggregate framework effect with one specific exception — holds after tripling the framework count from the original round's five to seven.

5.3 RQ4: Equivalence Testing

Using the linear mixed-effects model's full covariance matrix (not naive standard-error addition, which would understate precision for pairwise contrasts from a jointly-fit model), we ran a TOST equivalence test on all $\binom{8}{2} = 28$ pairwise condition contrasts. The primary margin (0.0244 attack-success-weight units, Cohen's $d = 0.2$ scaled by the combined dataset's pooled outcome standard deviation) and a sensitivity margin (0.05, one-fifth of the gap between adjacent verdict-weight steps) were both fixed before any condition-effect result was inspected.

All 28 of 28 pairwise contrasts are TOST-equivalent under both margins — the first result in this research program that directly supports a practical-negligibility claim, rather than merely failing to reject a null of no effect. This is not, however, a uniform result that should be reported as comfortably clean across the board: the single largest observed pairwise difference, `http` vs. `crewai` (difference $= -0.01838$, 90% CI $[-0.02430, -0.01245]$), clears the primary equivalence margin by a buffer of only 0.00006 — equivalent, but barely. All other 27 pairs clear the primary margin with substantially more room; the six next-largest differences all involve CrewAI as one side of the pair, consistent with Section 5.4's finding below.

5.4 The CrewAI Residual: A Precisely-Scoped Exception

Section 3.4 documented and corrected a real adapter defect that had biased CrewAI's outcomes upward before the fix. The result above shows that fix was necessary but not fully sufficient: CrewAI retains a small, statistically significant residual elevation in non-PASS outcomes relative to every other condition, in both the ordinal and linear mixed models, and it is the pairwise contrast closest to (though still inside) the equivalence margin.

Does this residual interact with the two frameworks added in this paper's extension round? No indication that it does. CrewAI's elevation relative to the two newest conditions (`google-adk`: difference = 0.01838; `semantic-kernel`: difference = 0.01667) is essentially the same magnitude as its elevation relative to the five original-round conditions (`http`: 0.01838, `autogen`: 0.01111, `langchain`: 0.01410, `llamaindex`: 0.01560, `openai-agents`: 0.01132) — CrewAI looks uniformly elevated relative to every other condition, old or new alike, not specifically or disproportionately relative to the two newly added frameworks. All of these differences still clear both TOST margins.

We offer one plausible, *unconfirmed* explanation rather than treating this as settled: CrewAI's role-playing template structurally requires role, goal, and backstory fields and cannot deliver a system prompt as a single message the way every other condition does, even once the *content* of those fields is normalized to match. This structural delivery difference was never independently tested for its own effect and is not established as the cause of the residual signal found here — we flag it as the most plausible next question this finding raises, not as a second finding.

5.5 Secondary Finding: A Reproducible Reasoning-Token-Exhaustion Failure

The frontier-tier `gpt-5.5` model, on one specific ASI10-category prompt, consumed its entire token budget on internal reasoning with no visible output, across both newly added frameworks and all three seeds — a fully deterministic 6/6 failure rate for this specific prompt–model pairing. This is not an isolated artifact of the two new frameworks: the original round independently documented a related pattern, with 13 of 21 reasoning-exhaustion cases (62%) concentrated on the same model–prompt combination. We treat this as a real, reproducible property of this model on this prompt, scored at minimum severity per this project's established treatment of missing-response rows, rather than excluded or treated as a new category.

A related, easily-missed pitfall: Google ADK's adapter includes a parameter intended to disable a Gemini model's internal reasoning (`thinking_budget=0`). Direct measurement shows this parameter has no observable effect — reasoning-token consumption with the parameter set (190 tokens) was statistically indistinguishable from a comparable call with no such parameter applied at all (189 tokens, via Semantic Kernel, which has no equivalent mechanism). Separately, Google ADK and Semantic Kernel report token usage under different accounting conventions for the same underlying provider: Google ADK's usage object treats a model's internal reasoning tokens as a subset of its total output-token count, while Semantic Kernel's native response object treats them as a separate, additive quantity. Comparing the two frameworks' usage fields directly, without correcting for this convention difference, would produce a spurious impression of a cost or efficiency gap between frameworks where none exists in the underlying billed usage. Neither issue affects the security findings reported above; both are documented here because a practitioner comparing these frameworks' own reported metrics directly would otherwise draw a wrong conclusion.

6 Limitations

- **Tooling constraint on the mixed-effects treatment.** No tool in this analysis environment implements a joint ordinal-mixed model (R's `ordinal::clmm` has no direct Python equivalent). We report two complementary models — an ordinal fixed-effects regression and a linear mixed-effects model with a random prompt intercept — rather than one model addressing both concerns simultaneously. Both agree in direction and significance pattern, which we take as reassuring but not equivalent to a single, formally joint treatment.
- **One combinability check remains unconfirmable, not confirmed.** Of four checks run before combining the two rounds' raw data, three passed with direct, concrete confirmation (identical prompt IDs, byte-identical prompt text, identical seed structure). The fourth — detector/scoring version recorded per trial row — is unconfirmable: neither underlying data file carries a version or commit field in its row schema, and no sidecar manifest supplies it independently. We carry this forward as a disclosed limitation on every result in this paper, not as an assumed non-issue.

- **Underpowered cells.** Several (framework, model, attack-family) cells have fewer than ten non-pass trials out of sixty, below the threshold we consider sufficient to interpret a cell's non-pass rate reliably. This threshold was fixed before any results were computed.
- **Uneven verification depth across provider paths for the two newest frameworks.** Their Anthropic paths were verified via live, real-API end-to-end calls; their OpenAI and Google paths received one live confirmatory call each rather than the repeated diagnostic testing applied to the frontier-tier reasoning-budget risk specifically. We do not believe this affected the results reported here, since the risk this asymmetry would most plausibly produce (reasoning-token exhaustion) was separately and directly ruled out at the production configuration, but the asymmetry in verification depth is a real property of this dataset worth disclosing. Semantic Kernel's Google connector additionally has a confirmed structural gap: its settings object exposes no reasoning-budget control parameter at all, unlike every other framework's Google-provider path in this codebase.
- **Model-version drift risk.** All models are referenced by dated, pinned identifiers. No provider guarantees these specific identifiers remain available or behaviorally unchanged indefinitely; a future replication using a provider's updated or renamed model would not be a like-for-like comparison against this dataset without explicit re-verification.
- **The CrewAI residual's cause is not established.** Section 5.4 offers a plausible structural explanation (role-playing template delivery format) but does not test it directly. This remains an open question, not a second finding.
- **Sensitivity to outcome weighting.** Under an alternative binary-collapse weighting scheme, the framework-share finding is robust (0.05%→0.56% on the original round's own sensitivity check, still far below Attack Family and Model). A per-model portability ranking computed from the same alternative weighting is not robust to this reweighting (Spearman $r = 0.31$ on the original round's own data). We report both outcomes rather than selecting the more favorable one: the coarse variance-decomposition and equivalence findings are a more reliable basis for practitioner guidance than any finer per-model ranking.
- **Frontier-tier missing data.** `gpt-5.5`'s combined empty-output rate (28.6% across the merged eight-condition dataset, per Section 4's corrected accounting) is substantial and exclusively concentrated in the frontier tier; frontier-tier results throughout this paper are reported as confirmatory rather than primary for this reason.

7 Discussion

The consistency of this result across an initial six-condition evaluation and a targeted two-framework extension, now analyzed jointly for the first time and under a statistically stronger treatment than either round attempted on its own, is a materially stronger basis for the claim that agentic security posture in this evaluation's threat model is not meaningfully determined by orchestration framework choice — at least for the eight conditions and five attack families tested, and specifically under payload-equivalent, model-response-based evaluation. Practically, this suggests that

security evaluation and mitigation effort for tool-using LLM agents is better directed at attack-category coverage and model-level behavior than at framework selection. We emphasize this is a claim about the frameworks and threat categories in this evaluation's scope, under a design that does not register or exercise runtime tool invocation, not a general claim about all possible orchestration designs, attack surfaces, or deployments where tools are actually registered and called.

The CrewAI residual (Section 5.4) is the finding this paper is most careful to scope precisely. It would be a mistake in either direction: dismissing it as noise (it is statistically significant in two independent model specifications, and the pairwise contrast closest to the equivalence margin), or inflating it into a general claim that CrewAI is measurably less safe (the effect is small in absolute terms, clears both equivalence margins, and does not interact with framework diversity as that diversity increased in this paper's extension round). The correct reading is narrower and, we think, more useful: even after fixing a confirmed adapter defect, a byte-level payload-identity guarantee does not by itself guarantee that every structural aspect of how a framework delivers that payload is equivalent, and a specific framework's system-prompt-delivery mechanism remains a plausible, untested source of small residual variation worth a security team's attention if CrewAI is the framework in production use.

The reasoning-token-exhaustion finding (Section 5.5) illustrates a distinct, practically important risk that a coarse pass/fail security metric alone would not surface: models that trade a fixed token budget between internal reasoning and visible output can silently fail to respond at all under adversarial input, a property we found to be specific to one model and one prompt combination, replicated deterministically across both newly added frameworks, rather than a property of any particular framework's implementation.

8 Conclusion

We set out to answer whether agentic orchestration framework choice meaningfully shapes the security posture of a tool-using LLM agent, and answered it with the largest controlled, payload-verified evaluation of this question we are aware of: seven frameworks and a direct-API baseline, six models, five attack families, and 9,360 trials collected across two rounds and analyzed jointly here for the first time. Framework choice explains a share of outcome variance roughly two orders of magnitude smaller than attack family and one order of magnitude smaller than model choice, and, for the first time in this research program, we can say so with a formal equivalence test rather than only a non-significant permutation result: all 28 pairwise execution-condition contrasts fall within a pre-specified small-effect margin. One framework, CrewAI, is a real, precisely-scoped exception — a small residual effect surviving a confirmed adapter-defect fix, uniform across old and new conditions alike, and reported here at exactly the strength the evidence supports. Every dataset, code version, and analysis script underlying this paper's findings is described in full in Section 4 and available to editors and reviewers on request.

References

[1] OWASP GenAI Security Project, Agentic Security Initiative, Sotiropoulos, J., Habler, I., Katz, K.: OWASP Top 10 for Agentic Applications. OWASP GenAI Security Project (2025)

[2] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., Fritz, M.: Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, 79–90 (2023) 2302.12173

[3] Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., Tramèr, F.: Agentdojo: A dynamic environment to evaluate attacks and defenses for llm agents. In: Proceedings of the 38th Conference on Neural Information Processing Systems, Datasets and Benchmarks Track (2024)

[4] Nguyen, V.K., Husain, M.I.: Penetration testing of agentic ai: A comparative security analysis across models and frameworks. arXiv preprint arXiv:2512.14860 (2025)

[5] Javed, W.: Detector-calibration failures in pattern-based LLM refusal classification: Discovery, generalization, and a confirmed false-positive pattern across models. In submission. Supersedes three earlier Figshare preprints (10.6084/m9.figshare.33110315, 33110402, 33110474), merged into a single manuscript. (2026)

[6] Anthropic: Claude Haiku 4.5 System Card. https://www.anthropic.com/claude-haiku-4-5-system-card (2025)

[7] OpenAI: Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/ (2026)

[8] Google DeepMind: Gemini 3.1 Flash-Lite Model Card. https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/ (2026)

[9] Anthropic: Claude Opus 4.8. https://www.anthropic.com/news/claude-opus-4-8 (2026)

[10] OpenAI: GPT-5.5 System Card. https://openai.com/index/gpt-5-5-system-card/ (2026)

[11] Google DeepMind: Gemini 3.5 Flash Model Card. https://deepmind.google/models/model-cards/gemini-3-5-flash/ (2026)

[12] Javed, W.: safelabs-eval: An OWASP ASI-aligned red-teaming and evaluation framework for AI agents. GitHub repository, Safe Labs AI Inc. https://github.com/AgentSafeLabs/safelabs-eval (2026)