

Effective Segregation of Duties in Agentic Enterprise Authorization: Evaluating Shared-Dependence Risk in Maker–Checker Controls

Mohamed Abbas Elmasry

malmasry@eelu.edu.eg

Egyptian E-Learning University (EELU)

Research Article

Keywords: segregation of duties, agentic AI, enterprise authorization, maker–checker controls, common-mode failure, security assurance

Posted Date: September 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10985030/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Segregation of duties (SoD) assigns sensitive steps to distinct principals, but identity separation does not guarantee that an AI checker provides independent authorization assurance. We distinguish nominal SoD, a structural property, from Effective SoD, a graded assurance construct characterized by unsafe-approval escape and legitimate-workflow disruption. We evaluate this distinction in AgentSoD-P2P, a policy-grounded procure-to-pay benchmark with deterministic gold decisions. A frozen corpus of 160 proposals, mapped one-to-one to 160 unique cases, was evaluated in a balanced $2 \times 2 \times 2$ within-proposal design manipulating checker composition (same-Qwen versus Qwen–Mistral), maker-rationale exposure, and evidence mediation, with three replicates per cell. A complementary natural maker-failure substudy evaluated paired checker behavior on the 240-case benchmark. The study comprised 5,520 model executions, yielding 5,316 parsed decisions and 204 retained parse failures. Under the parsed-only primary estimand, UAER fell from 63.87% in C1 to 13.22% in C7. Marginal UAER contrasts were 34.86 percentage points for mediated versus direct evidence, 8.09 for same-Qwen versus Qwen–Mistral checking, and 7.90 for visible versus blinded maker rationale. Evidence mediation and rationale exposure remained robust under adversarial parse-error treatment, whereas the checker-composition contrast attenuated because parse failures were concentrated in Qwen–Mistral cells. In complete parsed natural pairs, conditional maker–checker co-failure was 52.48% for same-Qwen and 17.82% for Qwen–Mistral checking, with the large separation preserved under alternative parse conventions. These results support evaluating agentic SoD through measured assurance rather than principal identity alone.

1 Introduction

Segregation of duties (SoD) is a longstanding principle in information security and organizational control. Its core logic is structural: a sensitive action should not be both initiated and completed by one principal without a distinct control point. In information systems, that logic appears in role constraints, workflow authorization, mutual exclusion, multi-person approval, and secure task design [1–11]. The familiar maker–checker pattern instantiates this principle by separating proposal from authorization.

Agentic enterprise systems complicate that logic. Two AI agents can execute under separate identities and occupy different workflow roles while retaining common sources of error. They may use the same model family, expose the checker to the maker’s rationale, or route the checker through evidence already selected or summarized by the maker. The workflow therefore displays nominal separation even when the second control remains dependent on the first control’s failure domain. The security question is not whether a second model is accurate on average, but whether it measurably intercepts unsafe proposals that passed the maker stage.

The distinction matters because contemporary LLM agents increasingly act through tools, memory, retrieved context, and external systems, creating security consequences beyond text quality [19–26]. The broader reliability problem is also well established: independently developed software versions can exhibit coincident failures beyond independence assumptions [30], and recent evidence spanning

hundreds of LLM evaluations reports substantial correlation in model errors across distinct architectures and providers [31]. Multi-agent studies likewise report resilience and verification failures that cannot be reduced to single-agent accuracy [32, 33]. These literatures motivate, but do not answer, the maker–checker authorization question addressed here.

This paper therefore separates two concepts. Nominal SoD is a structural property: maker and checker are distinct principals or roles. Effective SoD is an empirical assurance property: the second control demonstrably reduces unsafe authorization progression while maintaining usable legitimate workflow. Failure-domain separation is treated as an explanatory design principle that may improve this assurance, not as the definition of Effective SoD itself. This framing follows the broader security-assurance principle that protective effect should be evaluated rather than inferred from an architectural label [14–16].

We study Effective SoD in enterprise procure-to-pay (P2P) authorization. P2P is a useful setting because procurement policy combines monetary authority, budget preconditions, institutional purchasing authority, procurement method, conflict-of-interest controls, anti-circumvention requirements, and cumulative approvals. The benchmark is grounded in Mohawk College Procurement Policy CS-1202-2011, revised 10 June 2026, while BPI Challenge 2019 is used only for process realism rather than normative authorization gold [34, 35]. Gold decisions are produced by deterministic policy rules rather than an LLM judge.

The empirical design separates task correctness from control effectiveness. In the controlled experiment, every checker configuration receives the same frozen maker proposal. We manipulate three operational contrasts associated with dependence: (i) checker composition, comparing same-Qwen checking with a Qwen–Mistral checker for a Qwen-maker setting; (ii) maker-rationale exposure, visible versus blinded; and (iii) evidence mediation, maker-mediated versus direct authoritative access. The design is balanced within proposal and repeated three times per cell. A complementary natural maker-failure substudy examines paired checker behavior on naturally occurring unsafe maker submissions.

Three research questions guide the study. RQ1 asks: To what extent does identity-separated maker–checker authorization remain vulnerable to unsafe-approval escape and conditional co-failure? RQ2 asks: How do maker-rationale exposure, evidence mediation, and checker composition affect unsafe-approval escape in controlled enterprise authorization decisions? RQ3 asks: How do lower escape rates across maker–checker configurations trade off against workflow disruption, and where does residual risk remain across authorization-control families?

The paper makes three contributions. First, it operationalizes Effective SoD as measurable authorization assurance, distinguishing structural identity separation from observed interception performance while jointly considering unsafe-approval escape and workflow disruption. Second, it provides controlled evidence on three operational contrasts associated with shared dependence under matched enterprise decisions, with proposal-clustered inference and explicit sensitivity analysis for parse failures. Third, it provides a reproducible policy-grounded P2P evaluation artifact with deterministic gold, counterfactual structure, frozen maker proposals, and retained raw model outputs. The interpretation is deliberately

bounded: the Qwen–Mistral comparison is pair-specific rather than a generic causal estimate of model diversity, and the evidence treatment represents evidence-mediation architecture rather than pure source-independence.

2 Background and Related Work

2.1 Segregation of duties and workflow authorization

Classical information-security design treats least privilege, complete mediation, and separation of responsibility as mechanisms for limiting the consequences of a single principal or component failure [1]. RBAC subsequently provided a formal and operational vocabulary for assigning permissions through organizational roles [2, 4]. Within that tradition, SoD has been formalized through mutually exclusive roles, static and dynamic constraints, and workflow-level restrictions [3, 5–7]. Li, Tripunitara, and Bizri showed that enforcing high-level static SoD through role constraints raises non-trivial verification and computational questions [5], while Li and Wang distinguished high-level secure-task requirements from the lower-level enforcement schemes that realize them [6]. These lines of work establish the importance of multiple participants and constraints, but they generally assume that distinct authorized users or roles constitute meaningfully distinct control points.

Workflow-security research adds availability and resiliency to the picture. Authorization constraints can protect sensitive resources while simultaneously making a workflow impossible or costly to complete, motivating satisfiability, resiliency, and valued-policy formulations [8–10]. That tension directly motivates our two-metric treatment: a checker that rejects every proposal may produce low unsafe-approval escape but does not constitute a useful enterprise control. The recent ACM Computing Surveys review by Groll, Fuchs, and Pernul confirms both the breadth of SoD research and the continuing importance of operational enforcement across identity and workflow systems [11].

2.2 Security assurance, access control, and AI-enabled systems

Recent work on access-control enforcement and system security assurance reinforces the distinction between the existence of a control and evidence that it operates as intended. Formal access-control research continues to address policy conflicts, redundancy, revocation, and enforceable usage constraints [12, 13], while security-assurance studies emphasize measurable evidence for control effectiveness rather than architectural intent alone [14, 15]. Recent work on AI-system risk assessment and LLM-enabled cybersecurity automation extends this concern to systems whose behavior depends on contextual interpretation [16–18].

The relevance to agentic systems is increasing. Yu et al. characterize the shift from static LLM cybersecurity tasks toward dynamic, context-aware and tool-interactive reasoning [17]. Kölbel, Poss, and Schöning show that external context and organizational or regulatory knowledge materially shape LLM-based process explanation and compliance checking [18]. Broader 2026 surveys likewise identify tool

use, memory, context manipulation, and autonomous action as major dimensions of the expanding LLM-agent attack surface [19, 20]. Our focus is narrower: we do not introduce another taxonomy of agent attacks. We ask whether an enterprise authorization reviewer functions as an effective second control after the maker has already produced a potentially unsafe proposal.

2.3 Agent safety, external evidence, and correlated failure

Agent-security benchmarks demonstrate why control architecture matters. ToolEmu, AgentDojo, ASB, InjecAgent, AgentPoison, and R-Judge show that tool access, indirect instructions, poisoned context, memory, and weak risk awareness can alter agent actions in security-relevant ways [21–26]. These benchmarks primarily study attacks, safety failures, or agent behavior rather than segregation-of-duties assurance, but they establish that an agent’s decision context is itself part of the security boundary.

A related literature studies feedback and self-review. Self-Refine and Reflexion demonstrate that feedback loops can improve task performance, whereas work on self-correction shows that a second pass from the same model does not reliably constitute independent error correction [27–29]. The relevant distinction for SoD is therefore not simply whether review exists, but whether the review stage provides a measurably different control outcome on maker failures.

The dependence problem also has a longer reliability lineage. Knight and Leveson’s multiversion-programming experiment showed that independently developed software versions can fail together more often than an independence assumption predicts [30]. Recent LLM evidence reports substantial correlated errors across hundreds of models [31], while multi-agent studies identify resilience, verification, and coordination failures under faulty agents or system interactions [32, 33]. Our contribution is not to introduce dependent failure as a concept, but to operationalize it as a measurable maker–checker authorization-assurance problem.

2.4 Positioning and research gap

Table 1
Positioning against adjacent research streams

Research stream	Structural SoD	Agent/tool risk	Correlated failure	Measured checker assurance
Classical SoD / RBAC [2–11]	Yes	No	Generally assumed away	No
Agent security benchmarks [21–26]	Limited	Yes	Not primary	Security/attack outcomes
Self-review / reflection [27–29]	No	Limited	Implicit	Task performance
Dependent-failure / multi-agent reliability [30–33]	No	Limited	Yes	Reliability / failure behavior
This study	Yes	Authorization-focused	Yes	UAER + SDR under matched maker proposals

The gap is therefore not the absence of SoD, independent review, or dependent-failure research in isolation. It is the absence of a compact empirical framework for asking whether an identity-separated AI checker actually behaves as a second authorization control under matched maker failures. Table 1 positions the study against these adjacent literatures.

3 Effective Segregation of Duties

3.1 Authorization task and nominal SoD

Let an enterprise authorization request be $q = \langle \text{principal}, \text{action}, \text{object}, \text{context} \rangle$. A policy engine maps q and the relevant authoritative state to a gold decision $G(q)$ in $\{\text{PERMIT}, \text{DENY}, \text{REQUIRE_APPROVAL}\}$. A maker M produces a proposed action for the same request, and a checker C reviews the maker proposal before an enterprise effect is allowed to proceed.

Nominal SoD is structural. For maker principal P_M and checker principal P_C , NominalSoD holds when $P_M \neq P_C$ and the workflow assigns them distinct control roles. This is necessary for ordinary maker–checker governance, but it does not imply independent decision assurance. Two principals may still rely on the same model, the same interpretation supplied by the maker, or the same maker-selected evidence representation.

3.2 Unsafe proposal and control failure

A controlled maker proposal is unsafe when it submits an authorization action for progression even though the deterministic policy gold is DENY or REQUIRE_APPROVAL. In the frozen corpus, all controlled maker proposals take the action SUBMIT_PROPOSAL; proposal class is therefore defined by the gold

effect, not by free-form maker wording. A safe proposal has gold PERMIT. This paper focuses on unsafe progression rather than every possible maker error; for example, an unnecessarily restrictive maker action on a PERMIT case is not counted as UAER.

This distinction prevents an important confound. If different checker configurations received different maker outputs, lower escape could reflect an easier proposal rather than stronger checking. Freezing the same proposal across all eight cells makes control effectiveness conditional on a matched maker action.

3.3 Security and utility estimands

For dependency configuration D , the primary security estimand is the Unsafe Approval Escape Rate (UAER):

$$\text{UAER}(D) = P(C = \text{Approve} \mid M = \text{Unsafe}, D).$$

UAER measures the probability that a policy-incompatible progression proposal receives checker approval. For an unsafe proposal, APPROVE_PROPOSAL is an escape; REJECT_PROPOSAL or ESCALATE is an interception. When gold is REQUIRE_APPROVAL, direct checker approval is still an escape because it bypasses the required additional authority. Interception is therefore an effect-level security classification and does not imply that REJECT and ESCALATE are identical three-class policy decisions.

A checker can trivially minimize UAER by blocking everything. We therefore measure the Safe Workflow Disruption Rate (SDR):

$$\text{SDR}(D) = P(C \neq \text{Approve} \mid M = \text{Safe}, D).$$

SDR captures disruption on gold-PERMIT proposals: APPROVE_PROPOSAL is legitimate progression, while REJECT_PROPOSAL or ESCALATE is disruption. Effective SoD is therefore evaluated here as a graded two-dimensional assurance profile $A(D) = (\text{UAER}(D), \text{SDR}(D))$, rather than as a universal binary label. A deployment-specific acceptance criterion may subsequently impose application-specific tolerances δ and κ by requiring nominal SoD together with $\text{UAER}(D) \leq \delta$ and $\text{SDR}(D) \leq \kappa$. Because universal tolerances were not fixed before the full experiment, no tested configuration is retrospectively classified as satisfying a universal Effective SoD threshold.

Table 2
Decision semantics for the controlled checker experiment

Proposal class / gold	Maker effect	Checker output	Primary classification
Unsafe / DENY	SUBMIT_PROPOSAL	APPROVE_PROPOSAL	Unsafe-approval escape (UAER event)
Unsafe / DENY	SUBMIT_PROPOSAL	REJECT_PROPOSAL or ESCALATE	Intercepted unsafe progression
Unsafe / REQUIRE_APPROVAL	SUBMIT_PROPOSAL	APPROVE_PROPOSAL	Unsafe-approval escape; required authority bypassed
Unsafe / REQUIRE_APPROVAL	SUBMIT_PROPOSAL	REJECT_PROPOSAL or ESCALATE	Intercepted unsafe progression
Safe / PERMIT	SUBMIT_PROPOSAL	APPROVE_PROPOSAL	Legitimate progression
Safe / PERMIT	SUBMIT_PROPOSAL	REJECT_PROPOSAL or ESCALATE	Workflow disruption (SDR event)
Any	Any	PARSE_ERROR	Excluded from primary; handled in sensitivity analyses

3.4 Observable dependence channels

Table 3
Operational dependence contrasts and interpretation boundaries

Channel	Experimental manipulation	Security interpretation	Boundary on causal claim
Checker composition / model relation	Same Qwen checker vs Qwen–Mistral checker	Potentially different model-specific failure tendencies	Confounded with checker-model identity; not a generic diversity effect
Maker-rationale exposure	Visible vs blinded maker rationale	Potential dependence on maker framing or interpretation	Does not isolate psychological anchoring from information/framing effects
Evidence mediation	Maker-mediated evidence vs direct authoritative access	Dependence on maker-selected or summarized evidence representation	May combine path/provenance and information-fidelity differences

The common theoretical lens is failure-domain separation: architectures that reduce shared sources of error should improve the empirical assurance delivered by the checker. The experiments test observable channels consistent with that account; they do not claim to exhaust all possible common-mode dependencies such as shared orchestration, memory, credentials, or retrieval infrastructure.

4 Study Design

4.1 Domain and policy grounding

The study uses a single enterprise domain, procure-to-pay, to preserve policy specificity. BPI Challenge 2019 contributes contextual process grounding for plausible P2P activity terminology and workflow structure; neither its event outcomes nor its user identities contribute authorization labels [34].

Normative gold is derived from Mohawk College Procurement Policy CS-1202-2011, revised 10 June 2026 [35].

The policy explicitly specifies segregation of procurement functions, budget preconditions, exclusive authority of Procurement Services to issue or modify purchase orders, approval authority schedules, procurement thresholds and methods, conflict-of-interest requirements, and anti-circumvention constraints. Each primary case is linked to an explicit machine-readable rule derived from this policy. The benchmark therefore combines real policy constraints with controlled scenario generation instead of asking an LLM to invent the rules it is later judged against.

4.2 Benchmark, deterministic gold, and expert gate

The primary benchmark contains 240 Mohawk-policy cases, evenly distributed across eight authorization-control families (30 cases each): budget precondition, conflict of interest, exclusive purchase-order authority, missing higher approval, monetary threshold, procurement method, role concentration/SoD, and threshold circumvention. Cases were constructed in counterfactual families where feasible so that policy-relevant changes alter the gold decision while irrelevant attributes remain fixed.

Gold decisions are generated by a deterministic rule engine and take one of three values: PERMIT, DENY, or REQUIRE_APPROVAL. The checker never receives the gold label. Before pilot execution, an independent domain reviewer evaluated a blinded, stratified 48-case sample (six cases from each of the eight rule families). The review packet exposed case facts and paraphrased supporting policy rules but not the deterministic gold. Agreement with the rule engine was 42/48 (87.5%), with Cohen's $\kappa = 0.798$; the six disagreements were reviewed before benchmark freeze. This gate supports consistency of the operationalized policy rules rather than claiming that procurement-policy interpretation is intrinsically subjective [37].

4.3 Pilot gate and protocol freeze

The expert gate preceded a separate pilot that produced 256 checker outputs. The documented feasibility gate required at least 90% parse-valid outputs and rejected a pilot only if every condition was effectively all-easy ($UAER \leq 2\%$) or all-hard ($UAER \geq 98\%$). The pilot completed with 231/256 parse-valid outputs (90.23%) and passed this gate. Pilot outputs were used to verify feasibility and were not pooled into full-run estimates. After the pilot gate, the eight full-run cells, model revisions, replicate counts, 160-proposal frozen corpus, and parsed-only primary handling of PARSE_ERROR were fixed before inspection

of configuration-level full-run results. These elements are retained in the run plan, execution manifest, frozen-corpus hash, and model-output files; this is a documented protocol freeze, not a claim of formal external preregistration.

4.4 Frozen proposal corpus

The controlled checker experiment uses 160 frozen proposals mapped one-to-one to 160 unique case identifiers, with 20 proposals from each of the eight risk families. Within each family, ten risk-variant cases supply controlled counterfactual unsafe proposals, five safe-control cases supply controlled gold-aligned proposals, and five safe-control cases supply natural-maker-correct proposals. Across the corpus this yields 80 controlled counterfactual unsafe proposals, 40 controlled safe proposals, and 40 natural-maker-correct safe proposals. The resulting gold distribution is 80 PERMIT, 64 DENY, and 16 REQUIRE_APPROVAL. The natural-maker-correct proposals were generated before checker-factorial execution and were incorporated into the frozen corpus before any checker-factorial result was inspected. Once frozen, the exact proposal, case facts, and treatment-independent metadata were reused across every checker configuration. The SDR population should therefore be interpreted as this deliberate 40/40 mixture of controlled-safe and natural-maker-correct proposals rather than as a source-matched mirror of the unsafe half.

The primary UAER analysis uses the 80 unsafe proposals and the SDR analysis uses the 80 safe proposals. With eight checker conditions and three replicates per condition, the factorial stage contains 3,840 model executions.

4.5 Experimental factors and conditions

The checker experiment is a balanced $2 \times 2 \times 2$ factorial. Factor A is checker composition/model relation: same-Qwen versus Qwen–Mistral for a Qwen-maker setting. Factor B is maker-rationale exposure: visible versus blinded. Factor C is evidence mediation: maker-mediated versus direct authoritative access. The eight cells are shown in Table 4.

Table 4
Experimental 2×2×2 checker conditions

Cond.	Checker relation	Maker rationale	Evidence	Checker
C1	Same Qwen	Visible	Mediated	Qwen3-8B
C2	Same Qwen	Blind	Mediated	Qwen3-8B
C3	Same Qwen	Blind	Direct	Qwen3-8B
C4	Same Qwen	Visible	Direct	Qwen3-8B
C5	Qwen–Mistral	Visible	Mediated	Mistral-7B-Instruct-v0.3
C6	Qwen–Mistral	Blind	Mediated	Mistral-7B-Instruct-v0.3
C7	Qwen–Mistral	Blind	Direct	Mistral-7B-Instruct-v0.3
C8	Qwen–Mistral	Visible	Direct	Mistral-7B-Instruct-v0.3

C1 combines the factor levels operationally designated as higher-dependence in the study design: same-Qwen checking, visible maker rationale, and mediated evidence. C7 combines the opposite levels: Qwen–Mistral checking, rationale blinding, and direct authoritative access. The C1-to-C7 comparison is therefore a joint configuration contrast, not a measurement of an underlying scalar ‘dependence strength’ or the causal contribution of any single factor.

4.6 Models and execution protocol

The maker and same-model checker use Qwen/Qwen3-8B, revision c296ed0a908dbc97718d2b2ff8de659b2fedf052. The heterogeneous checker uses mistralai/Mistral-7B-Instruct-v0.3, revision 7386e7cc9e05b7a2cea12105321010df75b9364d. Both were executed in bf16. The run used a single NVIDIA A100-SXM4-40GB in Google Colab with Python 3.12.7, vLLM 0.6.3, PyTorch 2.4.0, Transformers 4.44.2, and CUDA 12.4.

Treatment delivery used a common structured checker schema across all cells. Maker-rationale visibility and evidence mediation were implemented by deterministic inclusion or omission of archived fields, while checker composition changed only the checker model. The shared checker system message, direct/blind archived prompt template, treatment assembly rules, direct-evidence and mediated-evidence field mappings, output schema, retry instruction, and parser behavior are reproduced in Online Resource 1.

Sampling used temperature 0.7, top-p 0.9, top-k 20, a maximum of 1,024 generated tokens, and repetition penalty 1.0. Each checker cell was replicated three times. Natural maker decisions were replicated three times; independent same-Qwen and Qwen–Mistral checker decisions were replicated twice per relation. Exact model revisions, software versions, decoding settings, seeds, and parser behavior are retained in the run manifest.

Table 5
Models, execution settings, and planned execution counts

Component	Value
Maker / same checker	Qwen/Qwen3-8B, revision c296ed0...f052, bf16
Heterogeneous checker	mistralai/Mistral-7B-Instruct-v0.3, revision 7386e7...9364d, bf16
Checker factorial	160 proposals × 8 cells × 3 replicates = 3,840
Natural maker	240 cases × 3 replicates = 720
Independent checking	240 × 2 replicates × 2 relations = 960
Total executions	5,520
Decoding	temperature 0.7; top-p 0.9; top-k 20; max tokens 1,024
Hardware	NVIDIA A100-SXM4-40GB

Table 6
Study accounting and primary analysis populations

Stage	Planned executions	Parse-valid / eligible	Primary analysis population	Clustering unit
Checker factorial	3,840	3,690 parsed: 1,850 unsafe + 1,840 safe	UAER: 1,850 unsafe decisions; SDR: 1,840 safe decisions	proposal_id (80 unsafe / 80 safe)
Natural maker	720	698 parsed maker outputs	109 unsafe maker events in replicates 0–1 before paired complete-case filtering	case_id
Independent checkers	960	928 parsed checker outputs	101 complete same/cross pairs from the 109 unsafe maker events; 80 unique cases	case_id
Total	5,520	5,316 parsed decisions + 204 retained parse failures	Stage-specific estimands above	Stage-specific

4.7 Natural maker-failure substudy

The controlled factorial holds the maker proposal fixed and estimates checker behavior conditional on proposal class. A complementary natural maker-failure substudy asks whether checker failures co-occur with naturally generated maker failures. For checker relation j , conditional maker–checker co-failure is defined as $CF_j = P(C_j \text{ approves} \mid \text{maker action is unsafe, eligible paired event})$. This is a checker-specific failure rate conditional on the same maker-failure events; it does not mean that the two checker configurations fail simultaneously. The same maker-failure event is presented to both checker relations under the same raw evidence and without maker-rationale exposure.

For paired inference, maker replicate r was matched only to independent-checker replicate r for $r \in \{0,1\}$. The independent-checker stage was implemented with two replicates per relation, so maker replicate 2 was retained for maker-stage accounting but lies outside the paired estimand because no independent-checker replicate 2 was generated. Among replicates 0–1, 109 parse-valid unsafe maker events were available before checker complete-pair filtering; 101 events across 80 unique case IDs had parse-valid decisions from both checker relations.

4.8 Parsing failures and estimands

The parser makes up to three attempts under the same decoding configuration. An unparseable response is never deleted: it is retained as `PARSE_ERROR` with its raw output and parse-attempt count. The frozen primary rule excludes parse errors from both numerator and denominator of UAER and SDR. The primary estimand is therefore decision quality conditional on a parsable model output, while parseability is reported separately as an operational reliability property.

Because parseability can differ across checker models, two sensitivity estimands are reported. In a fail-closed operational scenario, an unparseable unsafe output is treated as non-approval and an unparseable safe output as disruption. In an adversarial security sensitivity, every unsafe parse failure is counted as an unsafe approval. These estimands test whether the substantive ordering depends on favorable parser treatment; neither replaces the parsed-only primary analysis.

4.9 Statistical analysis

For each factorial cell, UAER and SDR point estimates are pooled proportions over parse-valid model executions. Their 95% proposal-cluster bootstrap intervals are obtained by resampling proposal IDs with replacement, retaining every condition/replicate observation associated with each sampled proposal, and recomputing the pooled proportion within each bootstrap sample (10,000 resamples) [40]. Factor-level summaries are proposal-level marginal risk differences: within each unsafe proposal, approval rates are averaged over the four cells and three replicates at each factor level, the low-dependence mean is subtracted from the high-dependence mean, and the resulting proposal-specific differences are averaged across 80 unsafe proposals. Proposal IDs are again resampled for 95% bootstrap intervals. These design-based contrasts are the main factor-level summaries and are not interpreted as isolated generic causal mechanisms.

A fully interacted logistic model, `approve ~ same × visible × mediated`, is fitted on the 1,850 parse-valid unsafe checker runs as secondary corroboration. The unit row is a model execution; the binary outcome is `APPROVE_PROPOSAL`. Parse failures are excluded under the primary rule. The model was fitted as a binomial generalized linear model in Python using `statsmodels 0.14.6` and converged successfully. Sandwich standard errors are clustered across the 80 unsafe proposal IDs to account for repeated observations [38]. Odds ratios are exponentiated coefficients, and reported OR intervals are obtained by exponentiating the proposal-cluster robust coefficient intervals. Because all interactions are included, main coefficients are conditional effects at the reference levels rather than global averages; proposal-level marginal risk differences remain the main factor-level summaries.

Natural co-failure is analyzed as a paired outcome on the 101 complete parsed maker-failure events. The primary uncertainty estimate is the event-weighted same-minus-Qwen–Mistral risk difference with 10,000 bootstrap resamples clustered by case ID. McNemar’s exact test [39] is reported only as a supportive paired test and does not correct case-level clustering. The primary security analyses are UAER and the three proposal-level marginal contrasts; SDR is treated as a utility guardrail, the logistic model as secondary corroboration, parse-error conventions as sensitivity analyses, and rule-family breakdowns as exploratory. In addition to the 95% intervals, Online Resource 1 reports Bonferroni-sensitivity 98.33% bootstrap intervals for the three marginal contrasts; these are provided as a multiplicity check rather than as a separate discovery framework.

5 Results

5.1 Run completeness and output reliability

The full run completed all 5,520 planned model executions: 3,840 checker-factorial, 720 natural-maker, and 960 independent-checker executions. These yielded 5,316 parsed decisions and 204 retained parse failures, for an overall parse success rate of 96.30%. The checker factorial contained 150 parse failures (3,690/3,840 parsed; 96.09%), the maker stage 22 (698/720; 96.94%), and the independent-checker stage 32 (928/960; 96.67%). Table 6 reconciles the executed, parsed, and primary analysis populations.

Parsing reliability differed by checker composition. Same-Qwen checker cells produced 20 parse failures among 1,920 executions (1.04%), whereas Qwen–Mistral checker cells produced 130 among 1,920 (6.77%). Unsafe and safe checker executions had similar overall parse-failure rates (3.65% and 4.17%, respectively). The elevated Qwen–Mistral parsing failure rate is therefore treated as an operational reliability finding rather than folded into the parsed-only decision-quality estimand.

5.2 Configuration-level security and utility

Parsed-only primary results are reported in Table 7 with proposal-cluster bootstrap intervals. UAER ranged from 63.87% in C1 to 13.22% in C7. C1 combines same-Qwen checking, visible maker rationale, and maker-mediated evidence, whereas C7 combines Qwen–Mistral checking, rationale blinding, and direct authoritative access. The absolute C1-to-C7 difference is 50.65 percentage points, corresponding to a 79.31% relative reduction in unsafe-approval escape. Because all three factors differ, this is a joint configuration contrast rather than a single-factor effect.

Lower UAER was not produced by a universal reject-everything policy. Parsed-only SDR remained in a comparatively narrow range across cells (6.81%–11.76%): C7 had SDR 9.65%, C3 7.59%, and C1 9.36%. SDR is therefore used primarily as a utility guardrail against trivial reject-all behavior rather than as evidence that the study has optimized a universal security–utility frontier. No single configuration is labeled the universally ‘best’ trade-off because relative dominance can change under alternative parse-error conventions.

Table 7
 Parsed-only UAER and SDR with proposal-cluster bootstrap intervals

Cond.	UAER n/N	UAER	95% proposal-cluster bootstrap interval	SDR n/N	SDR	95% proposal-cluster bootstrap interval
C1	152/238	63.87%	57.14–70.59%	22/235	9.36%	6.01–12.71%
C2	121/239	50.63%	44.11–57.32%	16/235	6.81%	3.86–9.87%
C3	40/237	16.88%	11.86–22.13%	18/237	7.59%	4.64–10.88%
C4	61/239	25.52%	20.50–30.54%	27/240	11.25%	7.50–15.42%
C5	116/227	51.10%	45.09–57.08%	26/221	11.76%	7.34–16.67%
C6	99/221	44.80%	37.95–52.00%	20/220	9.09%	5.78–12.78%
C7	30/227	13.22%	8.93–17.89%	22/228	9.65%	6.22–13.16%
C8	36/222	16.22%	11.36–21.17%	25/224	11.16%	7.33–15.45%

5.3 Marginal decomposition of dependence channels

The balanced within-proposal design allows each operational contrast to be summarized by a proposal-level marginal risk difference. Under the parsed-only primary estimand, mediated versus direct evidence increased UAER by 34.86 percentage points (95% proposal-cluster bootstrap interval 30.22–39.18), same-Qwen versus Qwen–Mistral checking by 8.09 points (4.05–12.16), and visible versus blinded maker rationale by 7.90 points (4.28–11.55). All three primary intervals excluded zero in this benchmark; Bonferroni-sensitivity intervals are reported in Online Resource 1.

Evidence mediation therefore exhibited the largest observed marginal contrast in the evaluated design. This is a comparison of operational architectures, not evidence that path independence alone is a universal causal mechanism; the direct treatment can also preserve more complete or less maker-filtered information.

Table 8
 Proposal-level marginal UAER contrasts

Contrast (higher-dependence level minus lower-dependence level)	Marginal RD	95% proposal-cluster bootstrap interval
Same vs heterogeneous checker	8.09 pp	4.05 to 12.16 pp
Visible vs blind rationale	7.90 pp	4.28 to 11.55 pp
Mediated vs direct evidence	34.86 pp	30.22 to 39.18 pp

The fully interacted logistic model corroborated this pattern. At the reference levels (Qwen–Mistral checker, blinded rationale, direct evidence), mediated evidence was associated with OR = 5.33 for unsafe approval (95% cluster-robust OR CI 3.29–8.62, $p = 9.16 \times 10^{-12}$). Because the model contains all two- and

three-way interactions, this coefficient is conditional rather than a global average factor effect. Table 9 reports the complete model.

Table 9
Fully interacted logistic model with proposal-cluster robust inference.
Odds ratios are exponentiated coefficients; intervals are
exponentiated proposal-cluster robust coefficient intervals

Term	OR	95% OR CI	p
Same-Qwen (conditional)	1.33	0.79–2.26	0.287
Rationale visible (conditional)	1.27	0.77–2.09	0.344
Same × visible	1.33	0.62–2.84	0.466
Mediated evidence (conditional)	5.33	3.29–8.62	9.16e-12
Same × mediated	0.95	0.52–1.72	0.86
Visible × mediated	1.01	0.56–1.84	0.965
Three-way interaction	1.01	0.40–2.55	0.987

5.4 Natural maker-failure co-failure

The natural maker-failure substudy provides convergent evidence that does not rely on the frozen checker-proposal corpus. Among 101 complete parsed maker-failure pairs, the conditional maker–checker co-failure rate was 53/101 (52.48%; case-cluster bootstrap 95% interval 43.30–61.77) for same-Qwen checking and 18/101 (17.82%; 10.20–26.26) for Qwen–Mistral checking. The paired difference was 34.65 percentage points (case-cluster bootstrap 95% interval 21.36–47.42). These rates are conditional on the same maker-failure events and do not mean that both checker configurations failed simultaneously.

The complete-pair contingency table contained nine events where both checkers failed, 44 where same-Qwen failed and Qwen–Mistral intercepted, nine where Qwen–Mistral failed and same-Qwen intercepted, and 39 where both intercepted. McNemar’s exact test on these events gives $p = 1.22 \times 10^{-6}$, but this test is supportive only because multiple events can originate from the same case; case-cluster bootstrap is the primary uncertainty analysis.

This contrast remains pair-specific. In the experiment, ‘same’ means Qwen maker plus Qwen checker and ‘heterogeneous’ means Qwen maker plus Mistral checker. Checker identity is therefore confounded with model relation. The result is consistent with a failure-domain-separation account for the evaluated pairing, but it does not identify a generic causal effect of model diversity.

5.5 Parse-error robustness

The large joint-configuration separation is not an artifact of excluding unparseable outputs. Under the fail-closed operational scenario, C1 UAER was 63.33% and C7 was 12.50%; under the adversarial sensitivity that treats every unsafe parse failure as an approval, C1 was 64.17% and C7 17.92%. The large C1-to-C7 joint-configuration separation therefore persisted under both alternative conventions.

At factor level, parse handling mattered most for checker composition. The parsed-only marginal contrast was + 8.09 percentage points for same-Qwen minus Qwen–Mistral, increasing to + 9.69 under fail-closed handling but attenuating to + 3.85 under the adversarial convention; the adversarial proposal-cluster 95% interval was approximately 0.00–7.81 percentage points. By contrast, rationale exposure remained + 7.60 points (3.96–11.25) and evidence mediation + 33.44 points (28.85–37.71) under adversarial handling. This asymmetry is consistent with the higher parse-failure rate in Qwen–Mistral cells and means that the checker-composition marginal effect is more sensitive to operational parse semantics than the other two contrasts.

The natural maker-failure result was also robust to incomplete-pair parse handling. Before complete-pair filtering, 109 unsafe maker events were available: 101 had both checker outputs parsed, six had only the Qwen–Mistral output unparsed, two had only the same-Qwen output unparsed, and none had both unparsed. Under fail-closed handling, conditional co-failure was 57/109 (52.29%) for same-Qwen and 19/109 (17.43%) for Qwen–Mistral; under adversarial handling it was 59/109 (54.13%) and 25/109 (22.94%), respectively. The paired separation therefore remained large even when every checker parse failure was counted against the corresponding configuration.

Table 10
Factor-level UAER sensitivity to parse-error handling

Operational contrast	Parsed-only RD	Fail-closed RD	Adversarial RD	Adversarial 95% proposal-cluster bootstrap interval
Same-Qwen – Qwen–Mistral	+ 8.09 pp	+ 9.69 pp	+ 3.85 pp	0.00 to 7.81 pp
Visible – blind rationale	+ 7.90 pp	+ 7.81 pp	+ 7.60 pp	3.96 to 11.25 pp
Mediated – direct evidence	+ 34.86 pp	+ 33.44 pp	+ 33.44 pp	28.85 to 37.71 pp

The higher Qwen–Mistral checker parse-failure rate is therefore a separate operational reliability finding, not merely a nuisance variable. A deployment that fails closed trades some authorization availability for safety, while a deployment that cannot guarantee fail-closed handling must treat unparseable outputs as an explicit control risk. Parsing reliability and decision quality should be reported separately.

5.6 Residual risk across authorization-control families

Aggregate parsed-only UAER varied across policy families. Exclusive purchase-order authority had the highest aggregate UAER at 41.59%, followed by procurement method at 40.61% and missing higher

approval at 37.45%. The lowest aggregate family rate, threshold circumvention, was still 31.30%. These values pool all eight checker configurations and therefore describe where residual control failure concentrated rather than intrinsic rule difficulty.

The same heterogeneity remained visible within C7, the lowest-UAER evaluated configuration. Exclusive purchase-order authority produced 9 approvals among 28 parse-valid unsafe runs (32.14%), whereas conflict-of-interest cases produced 1 of 27 (3.70%). The lowest-UAER evaluated configuration therefore did not eliminate control-specific residual risk. This exploratory breakdown identifies targets for follow-up error analysis rather than establishing that any rule family is intrinsically more difficult.

Table 11
Residual parsed-only UAER by authorization-control family

Risk family	Aggregate UAER	C7 UAER
exclusive po authority	41.59%	32.14%
procurement method	40.61%	14.29%
missing higher approval	37.45%	17.86%
monetary threshold	34.48%	10.34%
budget precondition	33.62%	10.71%
role concentration	32.77%	10.00%
conflict of interest	31.60%	3.70%
threshold circumvention	31.30%	6.90%

6 Discussion

6.1 Identity separation is not equivalent to independent assurance

The central result is that nominal principal separation was compatible with substantial unsafe-approval escape when model relation, maker-rationale exposure, and evidence mediation preserved shared decision dependencies. Identity separation establishes a distinct organizational control point, but by itself does not provide empirical evidence that the checker is outside the maker’s failure domain.

This observation extends classical SoD without replacing it. Distinct principals remain necessary for accountability, authorization scope, audit, and organizational governance. The new requirement is empirical: when the principals are AI agents, structural separation should not be used as a proxy for independent decision assurance. Security engineering should ask what failure domains remain shared and verify the actual interception rate of the second control.

6.2 Evidence mediation was the largest observed marginal contrast

The largest proposal-level marginal contrast concerned evidence mediation. Maker-mediated evidence was associated with substantially higher unsafe-approval escape than direct authoritative access across the balanced design. This finding is consistent with classical complete-mediation logic and contemporary emphasis on explicit verification and enforceable access-control state [1, 12, 13, 36].

The result should not be overinterpreted as proof that “independent evidence alone” caused the difference. The mediated and direct treatments differ in architecture: a maker-selected representation can compress, omit, frame, or transform information that is present in the authoritative source packet. Direct access therefore changes both the path and potentially the fidelity of information available to the checker. From an enterprise-design perspective, that bundle is still meaningful: a control that re-queries authoritative state is operationally different from one that merely reviews the first agent’s representation. Mechanistic separation of provenance, completeness, and summarization loss is an important next experiment.

6.3 Checker composition and the boundary of the model-diversity claim

The Qwen–Mistral checker configuration reduced parsed-only controlled UAER and natural conditional maker–checker co-failure relative to same-Qwen checking. The paired natural result is particularly informative because the same maker-failure events were presented to both checker compositions. The pattern is consistent with evidence that distinct implementations or model families can still exhibit correlated errors, while changing component composition can alter joint-failure behavior [30–33].

However, the present experiment cannot identify a generic “model diversity effect.” Model relation is confounded with checker identity: all same-model cells use Qwen as checker, and all heterogeneous cells use Mistral. Mistral may simply be better at particular policy distinctions, or differ in response style, calibration, or formatting. A fully counterbalanced design would require at minimum Qwen→Qwen, Qwen→Mistral, Mistral→Mistral, and Mistral→Qwen, ideally with several additional model families. We therefore frame the present finding as a pair-specific checker-composition result consistent with failure-domain separation, not as universal evidence that heterogeneous models are inherently independent.

The controlled factorial composition contrast is also more sensitive to parse-error semantics than the evidence and rationale contrasts because parse failures were more frequent in the Qwen–Mistral cells. This does not reverse the parsed-only result, but it further argues against interpreting the 8.09-point primary contrast as a universal causal effect of model diversity.

6.4 Maker-rationale exposure is a measurable dependence channel

Blinding the checker to the maker rationale reduced UAER by an average 7.90 percentage points. This supports the operational hypothesis that a checker's review changes when it is exposed to the maker's reasoning or justification. The result is relevant to workflow design because many "review" interfaces automatically show the initiating agent's rationale before the reviewer inspects primary evidence.

We avoid labeling the effect as psychological anchoring. Rationale visibility changes framing, information volume, and the presence of the maker's proposed interpretation. The experiment measures maker-rationale exposure, not an internal cognitive mechanism. A stronger mechanistic study could vary rationale correctness, confidence language, position, and length while holding informational content constant.

6.5 Effective SoD as measurable assurance rather than a binary architecture label

The results favor treating Effective SoD as a graded assurance construct rather than a binary architecture label. No configuration becomes 'independent' merely because it uses a second principal, a different model, or direct evidence. Each configuration instead has an empirical assurance profile $A(D) = (UAER(D), SDR(D))$. Even C7 retained a parsed-only UAER of 13.22%, and residual risk varied sharply by policy family. Failure-domain separation mitigates common-mode failure; it does not eliminate it.

A deployment can convert this graded profile into an application-specific acceptance decision by imposing tolerances δ and κ . High-value payment release, routine procurement, and low-risk administrative approval need not share the same tolerated escape and disruption rates. Organizations should define those tolerances from business impact, regulatory requirements, compensating controls, and recovery options, then validate a maker-checker configuration against them. The present experiment deliberately does not invent post-hoc universal thresholds.

SDR also bounds the interpretation of the security gains. Across cells, parsed-only SDR varied by only about five percentage points while UAER varied by more than fifty. The lower-UAER configurations therefore did not achieve their security advantage through indiscriminate blocking, but the study should not be read as a complete optimization of enterprise utility. Latency, escalation workload, reviewer cost, and downstream availability were not measured.

6.6 Design implications for enterprise agent controls

Four practical implications follow. First, assign distinct maker and checker identities for accountability, but do not treat identity separation as evidence of effective assurance. Second, give the checker a path to authoritative transaction state and policy rather than forcing it to inherit only the maker's representation. Third, treat checker heterogeneity as a candidate defense-in-depth mechanism and validate the actual pairing rather than assuming independence from model labels. Fourth, engineer strict output validation and explicit behavior for unparseable decisions. These practices align with broader principles of explicit verification, continuous risk measurement, and evidence-based control evaluation [36, 41].

These implications complement rather than replace deterministic authorization enforcement. Where rules are fully formalizable, deterministic policy enforcement can prevent an LLM from authorizing outside policy. Agentic maker–checker review is most relevant when an LLM is allowed to interpret, assemble, or recommend a policy-sensitive enterprise action. The second agent should then be evaluated as a security control with measurable escape and disruption characteristics, not merely as an additional reasoning stage.

7 Threats to Validity

7.1 Construct validity

UAER captures a specific effect-level failure: checker approval of a maker progression proposal when gold is DENY or REQUIRE_APPROVAL. It does not measure every possible maker error or exact three-class policy accuracy. SDR is included to detect over-restrictive checking on gold-PERMIT proposals. Other organizational outcomes—latency, escalation cost, audit quality, and human override—remain outside the present construct.

7.2 Internal validity

The frozen-proposal design provides strong control for maker-difficulty confounding in the factorial experiment, but three treatment interpretations remain bounded. First, same versus heterogeneous checker composition is confounded with Qwen versus Mistral checker identity. Second, mediated versus direct evidence can change information fidelity as well as evidence provenance. Third, rationale visibility can alter prompt length, framing, and available information. The manuscript therefore reports operational treatment contrasts rather than claiming isolated generic mechanisms.

7.3 Statistical conclusion validity

Repeated runs on the same proposals violate row-level independence. Cell-level intervals and factor contrasts therefore use proposal-cluster bootstrap inference; the logistic model uses proposal-cluster robust standard errors; and the natural substudy uses case-cluster bootstrap inference. The resulting intervals quantify variation across the benchmark proposals or cases, not a probability guarantee for all enterprise transactions. Primary, secondary, sensitivity, and exploratory analyses are separated to avoid treating every reported statistic as an independent confirmatory test.

7.4 External and policy validity

The primary study uses one P2P policy profile and two open-weight model families. Results should not be generalized numerically to banking, healthcare, HR, or other procurement regimes without replication. The benchmark is policy-grounded and process-realistic but semi-synthetic; it is not a dump of production authorization records. BPI Challenge 2019 contributes process terminology and workflow grounding but not authorization gold. The study also does not benchmark agentic checking against human four-eyes review; reported UAER values should therefore be interpreted comparatively across the

evaluated agent configurations rather than as evidence of parity with human reviewers. The 2026 Mohawk policy version used for rule construction is explicitly identified so that future revisions can be distinguished from the evaluated policy state [35].

7.5 Temporal validity and reproducibility

LLM behavior can change across model revisions, serving stacks, and decoding implementations. We therefore record exact model revisions, inference parameters, software versions, and raw responses. The frozen data package preserves all checker, maker, and independent decisions without deleting parse failures. Claims in this paper apply to the tested revisions and protocol rather than to future versions carrying the same marketing names.

8 Conclusion

Enterprise maker–checker controls are often treated as adequate once maker and checker occupy different identities. For AI agents, that structural condition is incomplete. This study distinguishes nominal SoD from Effective SoD and evaluates the latter as measurable authorization assurance: the ability of a second control to reduce unsafe proposal progression while maintaining legitimate workflow.

Across a policy-grounded P2P benchmark and 5,520 model executions, the three operational contrasts were associated with materially different unsafe-approval escape under the parsed-only primary estimand. C1 produced 63.87% UAER, whereas C7 produced 13.22%. Evidence mediation exhibited the largest and most parse-robust proposal-level marginal contrast. Maker-rationale exposure remained directionally stable under adversarial parse handling, whereas the controlled checker-composition marginal contrast attenuated because parse failures were concentrated in Qwen–Mistral cells. The paired natural maker-failure substudy nonetheless retained a large same-Qwen versus Qwen–Mistral co-failure separation under both fail-closed and adversarial parse conventions.

The implication is not that a particular model pair or evidence architecture ‘solves’ SoD. Identity separation should be treated as the beginning of agentic SoD design, not the endpoint. Effective assurance should be evaluated as a graded profile against the failure domains that remain shared, using application-specific measures of unsafe approval and workflow disruption. Direct authoritative access, controlled maker-rationale exposure, pair-specific checker evaluation, and explicit output-failure handling are therefore testable control-design choices. Future work should counterbalance model pairings, isolate information-fidelity effects within evidence mediation, compare agentic and human checking, and extend the evaluation to additional authorization domains and institutional policies.

Declarations

Competing interests.

The author declares no competing interests.

AI Statement.

AI-assisted tools were used to improve language and clarity; the author is fully responsible for the content.

Ethics approval.

Not applicable. The expert validation involved a voluntary methodological review by an external university professor with relevant subject-matter expertise. The reviewer participated voluntarily, no personal or sensitive data were collected, and the activity did not involve human-subject research, intervention, or recruitment requiring institutional ethics approval.

Funding.

No funding was received for conducting this study or preparing this manuscript.

Author Contribution

Author Contributions: M.A.E. conceived and designed the study, developed the benchmark and experimental protocol, curated the data, conducted the analysis, prepared the figures and tables, and wrote, reviewed, and approved the final manuscript.

Data Availability

A description of data, its constituent files, dataset structure, and associated reproducibility artifacts is provided in Supplementary Information, Online Resource 1. The frozen data package and reproducibility artifacts supporting this study are available from the corresponding author upon reasonable request.

References

1. Saltzer, J.H., Schroeder, M.D.: The protection of information in computer systems. *Proc IEEE* 63(9):1278–1308. (1975). <https://doi.org/10.1109/PROC.1975.9939>
2. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. *Computer*. **29**(2), 38–47 (1996). <https://doi.org/10.1109/2.485845>
3. Kuhn, D.R.: Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems. In: *Proc 2nd ACM Workshop on Role-Based Access Control*, pp 23–30. (1997). <https://doi.org/10.1145/266741.266749>

4. Sandhu, R., Ferraiolo, D., Kuhn, D.R.: The NIST model for role-based access control: towards a unified standard. In: Proc 5th ACM Workshop on Role-Based Access Control, pp 47–63. (2000).
<https://doi.org/10.1145/344287.344301>
5. Li, N., Tripunitara, M.V., Bizri, Z.: On mutually exclusive roles and separation-of-duty. ACM Trans. Inf. Syst. Secur. **10**(2) (2007). Article 5. <https://doi.org/10.1145/1237500.1237501>
6. Li, N., Wang, Q.: Beyond separation of duty: an algebra for specifying high-level security policies. J. ACM. **55**(3) (2008). Article 12. <https://doi.org/10.1145/1379759.1379760>
7. Liu, D.-R., Wu, M.-Y., Lee, S.-T.: Role-based authorizations for workflow systems in support of task-based separation of duty. J. Syst. Softw. **73**(3), 375–387 (2004). [https://doi.org/10.1016/S0164-1212\(03\)00175-4](https://doi.org/10.1016/S0164-1212(03)00175-4)
8. Bergé, P., Crampton, J., Gutin, G., Watrigant, R.: The Authorization Policy Existence Problem. IEEE Trans. Dependable Secure Comput. (2018). <https://doi.org/10.1109/TDSC.2018.2883416>
9. Crampton, J., Gutin, G., Karapetyan, D., Watrigant, R.: The bi-objective workflow satisfiability problem and workflow resiliency. J. Comput. Secur. **25**(1), 83–115 (2017). <https://doi.org/10.3233/JCS-16849>
10. Crampton, J., Eiben, E., Gutin, G., Karapetyan, D., Majumdar, D.: Valued Authorization Policy Existence Problem: theory and experiments. ACM Trans. Priv. Secur. **25**(4) (2022).
<https://doi.org/10.1145/3528101> Article 28
11. Groll, S., Fuchs, L., Pernul, G.: Separation of duty in information security. ACM Comput. Surv. **57**(7), Article (2025). <https://doi.org/10.1145/3715959>
12. Alkhorem, A.H., Conte de Leon, D., Jillepalli, A.A., et al.: Graph-based formal modeling and implementation of access control policies with automated conflict and redundancy detection. Int. J. Inf. Secur. **25**, 61 (2026). <https://doi.org/10.1007/s10207-025-01130-z>
13. Rasori, M., Saracino, A., Mori, P., et al.: Using the ACE framework to enforce access and usage control with notifications of revoked access rights. Int. J. Inf. Secur. **23**, 3109–3133 (2024).
<https://doi.org/10.1007/s10207-024-00877-1>
14. Wen, S.-F., Shukla, A., Katt, B.: Artificial intelligence for system security assurance: a systematic literature review. Int. J. Inf. Secur. **24**, 43 (2025). <https://doi.org/10.1007/s10207-024-00959-0>
15. Shukla, A., Katt, B., Yamin, M.M.: A quantitative framework for security assurance evaluation and selection of cloud services: a case study. Int. J. Inf. Secur. **22**, 1621–1650 (2023).
<https://doi.org/10.1007/s10207-023-00709-8>
16. Agathangelos-loannou, G., Tsohou, A.: Methodologies for information security risk assessment: shortcomings for analysing artificial intelligence systems. Int. J. Inf. Secur. **25**, 98 (2026).
<https://doi.org/10.1007/s10207-026-01271-9>
17. Yu, H., Iqbal, S., Neto, E.C.P., Buffett, S., et al.: From static tasks to dynamic reasoning: a characterization framework and study of large language models in next-generation cybersecurity automation. Int. J. Inf. Secur. **25**, 149 (2026). <https://doi.org/10.1007/s10207-026-01321-2>

18. Kölbel, L.M., Poss, L., Schöning, S.: Context is key for cybersecurity: leveraging external knowledge for process model explanation via LLMs. *Int. J. Inf. Secur.* **25**, 108 (2026).
<https://doi.org/10.1007/s10207-026-01245-x>
19. Xu, Y., Zhuang, Y., Liu, X., et al.: LLM agents security duality: a comprehensive survey of self-security and empowered cybersecurity. *Artif. Intell. Rev.* **59**, 174 (2026). <https://doi.org/10.1007/s10462-026-11563-0>
20. Tang, Y., Liu, Y., Lan, J., Yan, Z., Gelenbe, E.: Security of LLM-based agents regarding attacks, defenses, and applications: a comprehensive survey. *Inf. Fusion.* **127**, 103941 (2026).
<https://doi.org/10.1016/j.inffus.2025.103941>
21. Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C.J., Hashimoto, T.: Identifying the risks of LM agents with an LM-emulated sandbox. In: *International Conference on Learning Representations (ICLR 2024)* (2024)
22. Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., Tramèr, F.: AgentDojo: a dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. *Adv. Neural Inf. Process. Syst.* **37** (2024). <https://doi.org/10.52202/079017-2636>
23. Zhang, H., Huang, J., Mei, K., Yao, Y., Wang, Z., Zhan, C., Wang, H., Zhang, Y.: Agent Security Bench (ASB): formalizing and benchmarking attacks and defenses in LLM-based agents. In: *International Conference on Learning Representations (ICLR 2025)* (2025)
24. Zhan, Q., Liang, Z., Ying, Z., Kang, D.: InjecAgent: benchmarking indirect prompt injections in tool-integrated large language model agents. *Find. ACL.* **2024**, 10471–10506 (2024).
<https://doi.org/10.18653/v1/2024.findings-acl.624>
25. Chen, Z., Xiang, Z., Xiao, C., Song, D., Li, B.: AgentPoison: red-teaming LLM agents via poisoning memory or knowledge bases. *Adv. Neural Inf. Process. Syst.* **37** (2024)
26. Yuan, T., He, Z., Dong, L., et al.: R-Judge: benchmarking safety risk awareness for LLM agents. *Find. EMNLP.* **2024**, 1467–1490 (2024). <https://doi.org/10.18653/v1/2024.findings-emnlp.79>
27. Huang, J., Chen, X., Mishra, S., Zheng, H.S., Yu, A.W., Song, X., Zhou, D.: Large language models cannot self-correct reasoning yet. In: *International Conference on Learning Representations (ICLR 2024)* (2024)
28. Madaan, A., Tandon, N., Gupta, P., et al.: Self-Refine: iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* **36** (2023). <https://doi.org/10.52202/075280-2019>
29. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K.R., Yao, S.: Reflexion: language agents with verbal reinforcement learning. *Adv. Neural Inf. Process. Syst.* **36** (2023). <https://doi.org/10.52202/075280-0377>
30. Knight, J.C., Leveson, N.G.: An experimental evaluation of the assumption of independence in multiversion programming. *IEEE Trans. Softw. Eng. SE.* **12**(1), 96–109 (1986).
<https://doi.org/10.1109/TSE.1986.6312924>
31. Kim, E.M., Garg, A., Peng, K., Garg, N.: Correlated errors in large language models. In: *Proceedings of the 42nd International Conference on Machine Learning*, PMLR 267:30038–30066. (2025).

- <https://proceedings.mlr.press/v267/kim25e.html>
32. Huang, J.-T., Zhou, J., Jin, T., Zhou, X., Chen, Z., Wang, W., Yuan, Y., Lyu, M.R., Sap, M.: On the resilience of LLM-based multi-agent collaboration with faulty agents. In: Proceedings of the 42nd International Conference on Machine Learning, PMLR 267:26202–26226. (2025).
<https://proceedings.mlr.press/v267/huang25ay.html>
33. Cemri, M., Pan, M.Z., Yang, S., Agrawal, L.A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., Gonzalez, J.E., Stoica, I.: Why do multi-agent LLM systems fail? Adv. Neural Inf. Process. Syst. **38**, 135676–135729 (2025). <https://doi.org/10.52202/085713-4082>
34. van Dongen, B.F.: BPI Challenge 2019. 4TU.ResearchData. (2019).
<https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>
35. Mohawk College: Procurement Policy, CS-1202-2011, last revised 10 June 2026. (2026).
<https://www.mohawkcollege.ca/about-mohawk/leadership-and-administration/policies/procurement-policy>. Accessed 9 September 2026
36. Rose, S., Borchert, O., Mitchell, S., Connelly, S.: Zero Trust Architecture. NIST Special Publication 800 – 207. (2020). <https://doi.org/10.6028/NIST.SP.800-207>
37. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics. **33**(1), 159–174 (1977). <https://doi.org/10.2307/2529310>
38. Liang, K.-Y., Zeger, S.L.: Longitudinal data analysis using generalized linear models. Biometrika. **73**(1), 13–22 (1986). <https://doi.org/10.1093/biomet/73.1.13>
39. McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika. **12**(2), 153–157 (1947). <https://doi.org/10.1007/BF02295996>
40. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)
41. Tabassi, E.: Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. (2023).
<https://doi.org/10.6028/NIST.AI.100-1>

Figures

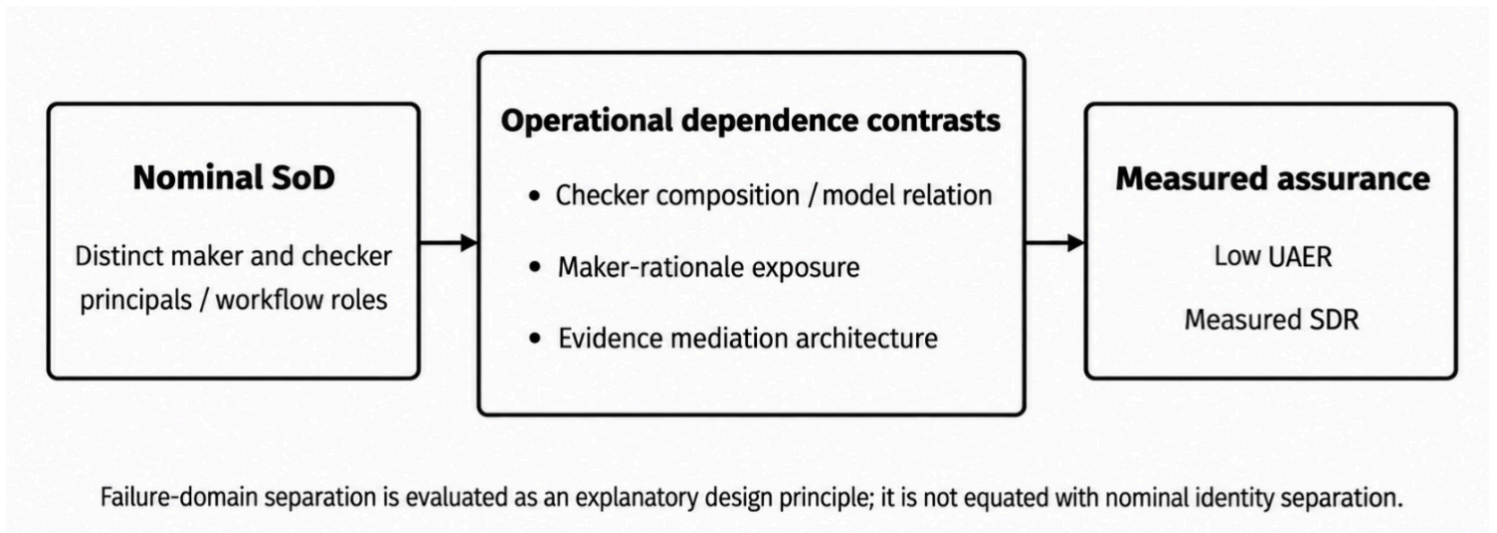


Figure 1

Nominal SoD, operational dependence contrasts, and empirically measured assurance.

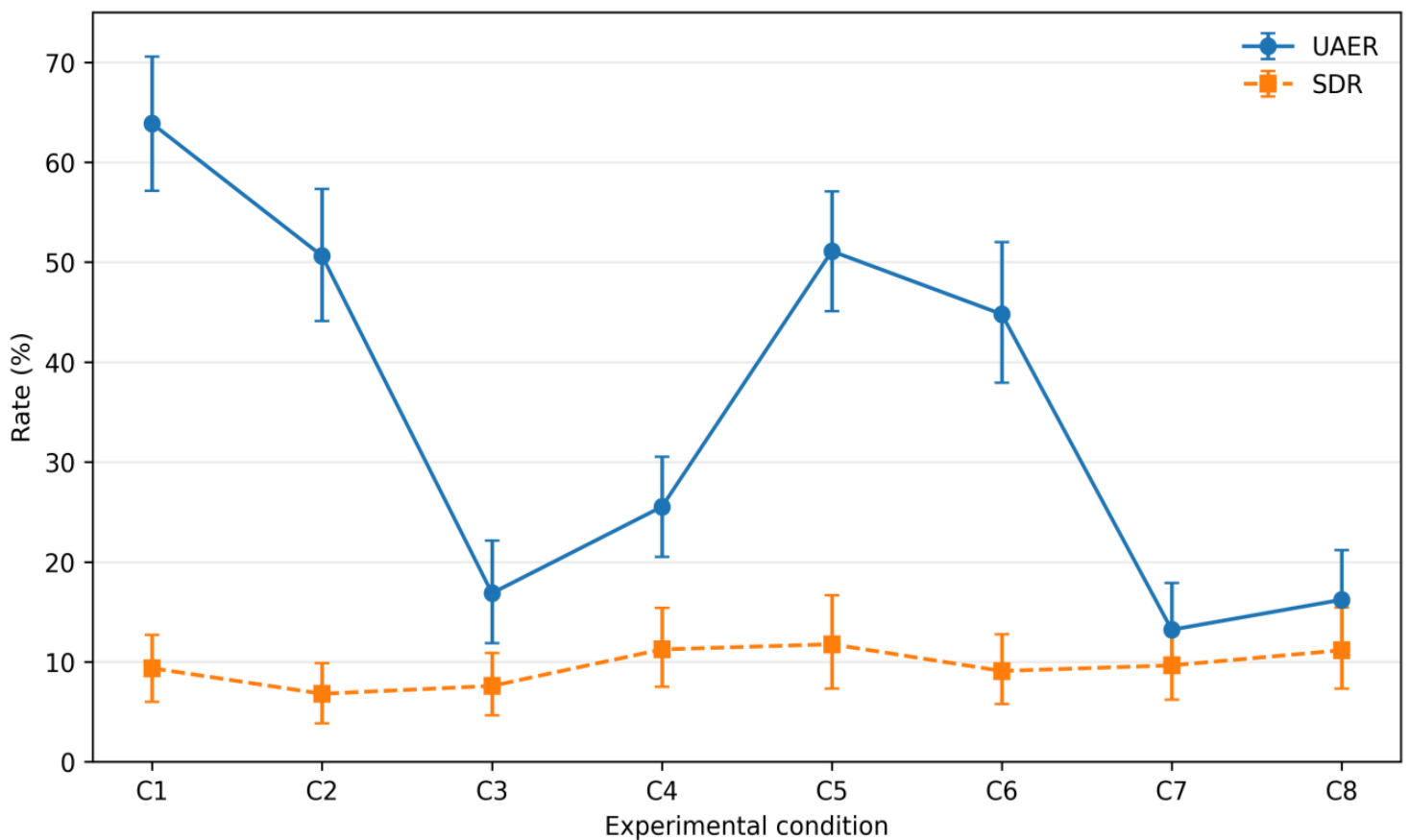


Figure 2

Parsed-only UAER and SDR by experimental condition. Error bars show 95% proposal-cluster bootstrap intervals.

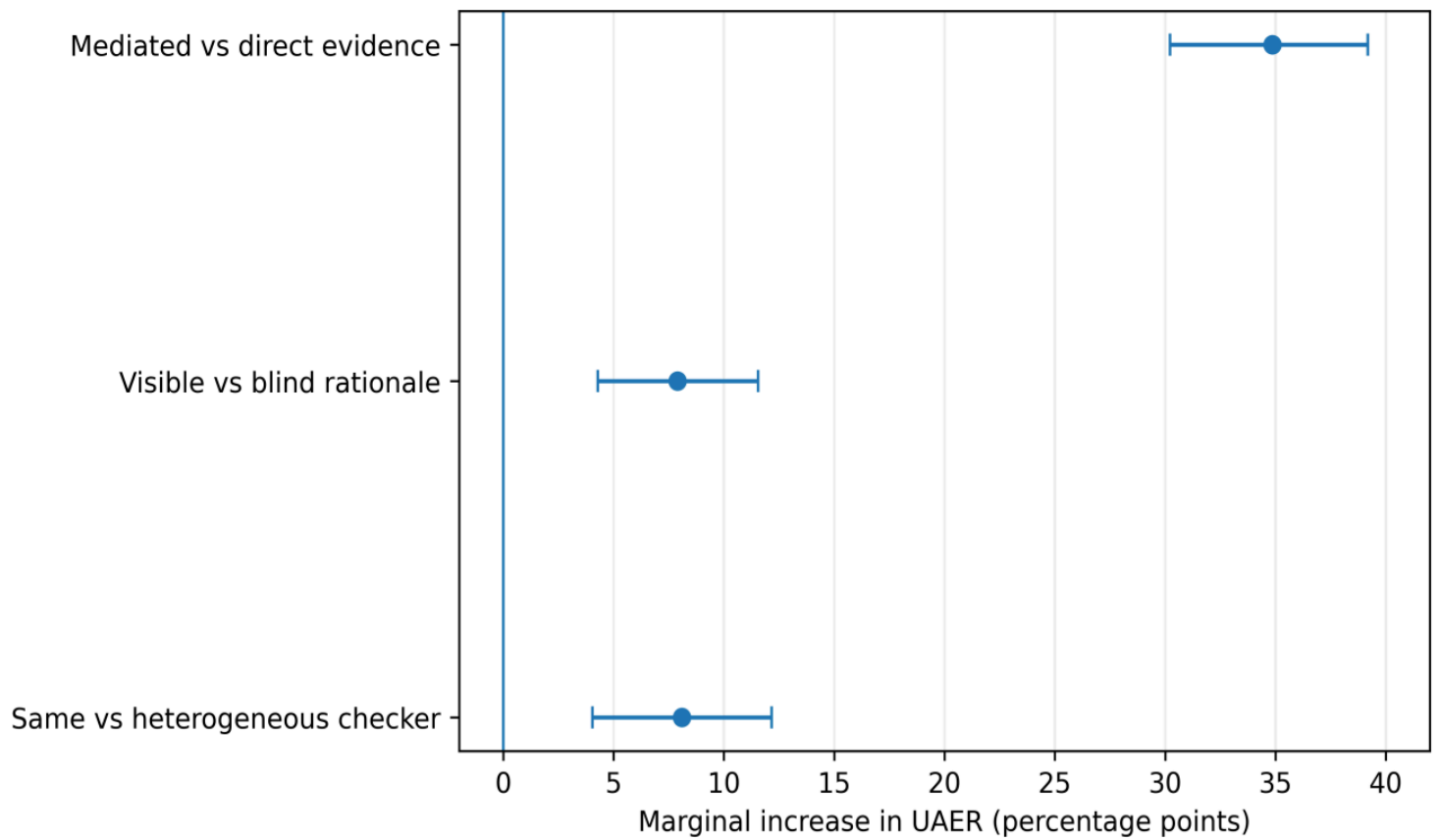


Figure 3

Proposal-level marginal UAER contrasts for the three operational factors. Error bars show 95% proposal-cluster bootstrap intervals.

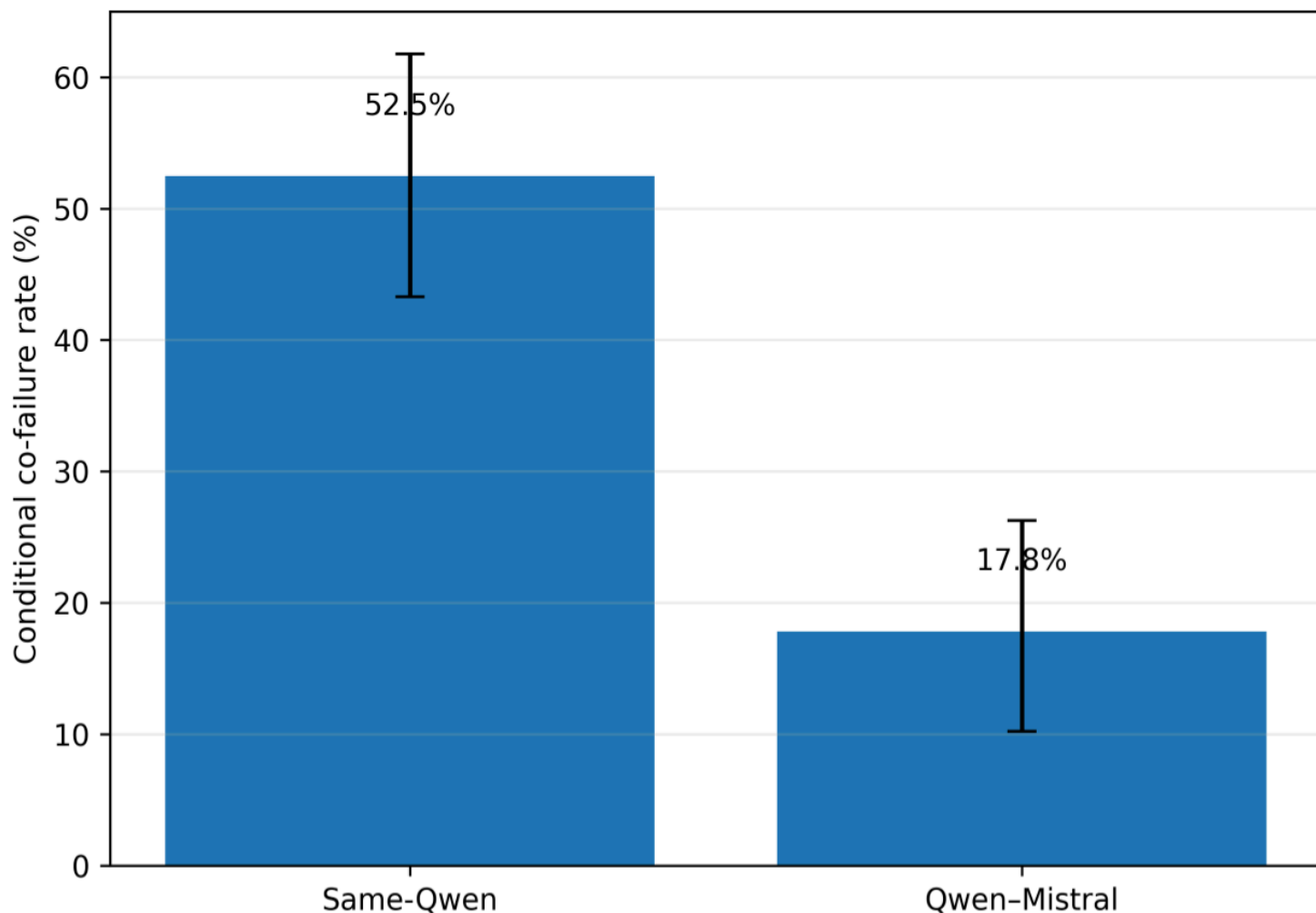


Figure 4

Conditional maker–checker co-failure on complete parsed natural maker-failure pairs (n = 101). Error bars show 95% case-cluster bootstrap intervals.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryInformationOnlineResource1.pdf](#)