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.

When Does Cross-Agent Peer Verification Detect Unfaithful Self- Reports? An Empirical Study of Real LLM Tool-Use Agent Trajectories

Vivek Garike

vgarike27701@ucumberlands.edu

University of the Cumberlands

Research Article

Keywords: LLM agents, trust and reputation modeling, cross-agent peer verification, multi-agent consensus, self-report faithfulness

Posted Date: October 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11020126/v1

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

Additional Declarations: No competing interests reported.

When Does Cross-Agent Peer Verification Detect Unfaithful Self-Reports? An Empirical Study of Real LLM Tool-Use Agent Trajectories

Vivek Garike

Department of Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Corresponding author: Vivek Garike, e-mail: vgarike27701@ucumberlands.edu

ORCID: 0009-0005-9152-4567

Abstract

Large language model (LLM) agents that autonomously execute tool calls for users are increasingly deployed in customer-facing settings, where peer-based trust and reputation mechanisms — long used in multi-agent systems to infer trustworthiness from peer testimony — are natural candidates for calibrating confidence in an agent's self-report. Yet whether such peer testimony can detect when an agent's self-report is unfaithful to verified ground truth has not been empirically examined using real production data. This paper studies 1,980 real agent trajectories from tau-bench, spanning two domains (retail, airline) and two production models (GPT-4o, Claude-3.5-Sonnet), each verified against ground-truth database-state comparisons. Self-reported success claims are unfaithful to verified outcomes in 13.8 to 31.5 percent of trajectories, with false-success more common than false-failure. Whether cross-agent peer verification — including reputation-weighted aggregation, the canonical multi-agent trust-inference mechanism — transfers from prior shared-observation settings to this independent-execution setting is then tested; it does not, since trials are independent stochastic re-attempts rather than shared-event observations, with all four aggregation rules underperforming solo self-report (56.5 percent versus 51.7–55.6 percent). This difference is statistically significant (Holm–Bonferroni corrected) against naive-mean and reputation-weighted aggregation in both peer-group compositions and against trimmed mean in the pooled-model setting, and is distinct from a separate failure mode driven by

correlated errors rather than absent shared observation. Motivated by this result, a calibration strategy using peer task-difficulty quartiles is proposed, distinguishing its oracle-level from its realistically-deployable effectiveness. The rule-based classifier used throughout is validated against constructed edge cases and 152 messages sampled from the real corpus, yielding 88.2% agreement and a Cohen's kappa of 0.812. As a preliminary domain-generalization check, the same pipeline is applied to 43 newly collected trajectories from the telecom domain of τ^2-bench (tau-bench's successor); results are directionally consistent but, based on a single model and much smaller sample, are reported as a non-confirmatory pilot rather than a full replication. This work identifies a structural boundary condition on when peer-based trust and reputation mechanisms are informative in multi-agent systems — showing that they transfer from shared-observation domains to independent-execution agentic settings only partially, and characterizing empirically when and why.

Keywords LLM agents · trust and reputation modeling · cross-agent peer verification · multi-agent consensus · self-report faithfulness

1 Introduction

Autonomous large language model (LLM) agents increasingly execute real, consequential actions on behalf of users—cancelling orders, issuing refunds, rebooking flights—through tool-calling interfaces, and then report the outcome of that action back to the user in natural language. A user, and often the deploying organization, has no independent way to verify that the agent's account of what happened matches what its tool calls actually did to the underlying system state. Peer-based trust and reputation mechanisms—validated for decades in multi-agent systems where peers observe a shared transaction or event and testify about a counterpart's trustworthiness (Jøsang et al. 2007; Sabater and Sierra 2001; Yu and Singh 2003)—are an increasingly proposed way to

calibrate confidence in such self-reports without requiring cryptographic infrastructure. This paper asks a simple but, to our knowledge, empirically unanswered question using real production-model data: can peer agents attempting the same task serve as reputation-bearing witnesses to detect when an LLM agent's self-report of task completion is unfaithful to the verified ground truth, and how often does such unfaithfulness occur in the first place?

Recent work has made substantial progress on adjacent but distinct problems. AgentBound (Kaul et al. 2026) and AEGIS (Yuan et al. 2026) both achieve strong accountability guarantees by cryptographically capturing the actual tool-call parameters and policy decisions at the point of execution—but neither examines whether an agent's subsequent natural-language summary of that execution is itself faithful. Auditable Agents (Nian et al. 2026) provides a rigorous taxonomy of what auditability requires and, critically, reports an ecosystem scan finding 617 security findings across six prominent open-source agent projects, with the explicit conclusion that in most currently deployed systems, execution logs are "partial or absent." This motivates examining a complementary question: in the common case where full infrastructure-level logging is not available, is an agent's self-report the only record, and can it be trusted? CP-WBFT (Zheng et al. 2026) demonstrates that LLM agents can serve as Byzantine-robust peer verifiers of each other's outputs—but only on tasks with a single, objectively checkable correct answer (mathematical reasoning, safety classification), not on free-form retrospective claims about completed real-world actions.

This paper addresses this gap directly using real data. We analyze 1,980 trajectories from tau-bench (Yao et al. 2024), a benchmark in which LLM agents interact with simulated retail and airline customer-service systems, each trajectory independently scored by comparing the final database state against an annotated ground-truth goal state—providing objective, non-self-reported

verification of whether each execution actually succeeded. Because tau-bench provides multiple independent trials of the same underlying task, executed by two different production models (GPT-4o and Claude-3.5-Sonnet), we are able to test, using real agent behavior rather than synthetic simulation, whether cross-agent peer verification—a technique validated in prior work on domains where peers observe a single shared ground-truth event—transfers to this setting.

It does not transfer directly, and understanding precisely why is this paper's central contribution: a boundary condition on when peer-based trust and reputation mechanisms are informative in multi-agent systems. We show that because tau-bench trials are independent stochastic executions (the same task attempted multiple times can succeed on some attempts and fail on others, due to genuine variation in the agent's own behavior across runs) rather than multiple observations of one fixed event, peer agents' self-reports carry substantially less information about any single trial's specific outcome than they did in prior shared-observation domains. We characterize this boundary condition precisely, confirm it is robust to whether peer groups pool both models or are restricted to same-model trials only, and use the resulting insight to propose and empirically validate an alternative, realistically-deployable calibration strategy based on peer task-difficulty priors rather than direct peer-vote aggregation. As a preliminary check of whether these findings extend beyond tau-bench, we additionally apply the identical pipeline to a newly collected sample from the telecom domain of τ^2-bench (Barres et al. 2025), the actively maintained successor benchmark, and report this domain-generalization pilot transparently alongside the primary two-domain analysis.

1.1 Contributions

- An empirical characterization, using real GPT-4o and Claude-3.5-Sonnet execution trajectories, of naturally-occurring self-report unfaithfulness rates in LLM tool-use agents,

without any injected or synthetic adversarial perturbation, reported with 95% confidence intervals.

- A rigorous, leakage-controlled empirical test of direct cross-agent peer aggregation (naive mean, trimmed mean, coordinate-wise median, and reputation-weighted aggregation—the canonical trust-inference mechanism in multi-agent reputation systems) as a verification mechanism, showing it does not outperform trusting an individual agent's self-report in this real-data setting—a difference confirmed statistically significant, after Holm–Bonferroni correction across the full family of tests, against naive-mean and reputation-weighted aggregation in both peer-group compositions and against trimmed mean in the pooled-model setting—together with a mechanistic explanation for why, confirmed robust to same-model-only peer group composition.
- Identification of a boundary condition on when peer-based trust and reputation mechanisms are informative in multi-agent systems—the specific structural distinction between shared-event observation and independent stochastic re-execution—that determines whether cross-agent peer verification transfers from prior validated domains to real agentic tool-use settings, and its differentiation from the mechanistically distinct correlated-error failure mode documented in prior multi-agent debate literature.
- A validated, honestly-scoped alternative calibration strategy (peer task-difficulty quartiles) with an explicit, quantified distinction between its oracle-level and realistically-deployable effectiveness, each with 95% confidence intervals.
- A preliminary domain-generalization pilot extending the identical classifier and leave-one-out peer-aggregation pipeline to 43 newly collected trajectories from the telecom domain of τ^2-bench (single model, 15 tasks, 3 trials each), finding a directionally consistent but

statistically underpowered replication of the core solo-versus-peer-aggregation finding, and a domain-specific classifier-recall limitation with a fully disclosed manual-verification remedy.

- A fully reproducible pipeline, transparent rule-based self-report classifier—validated both against constructed linguistic edge cases and against a random sample of 152 real corpus messages (88.2% agreement, Cohen's kappa 0.812)—and complete disclosure of methodological limitations, evaluated entirely on public benchmark data.

2 Related Work

This section first positions the present contribution against the classical multi-agent trust and reputation literature whose peer-aggregation paradigm this paper tests, and then against the four most directly relevant recent papers on LLM agent accountability, each read in full for this analysis rather than summarized from secondary sources.

2.1 Trust and Reputation Models in Multi-Agent Systems

Trust and reputation systems are a long-standing pillar of multi-agent systems research, formalizing how an entity's trustworthiness can be inferred from the aggregated testimony of peers. Jøsang, Ismail, and Boyd (Jøsang et al. 2007) survey the computational foundations of this paradigm, distinguishing centralized reputation engines from distributed, witness-based schemes and identifying witness reliability as a central open problem. Sabater and Sierra (Sabater and Sierra 2001) propose REGRET, a reputation model that explicitly incorporates a social dimension, combining an agent's own direct experience with witness information reported by other agents in the same society. Yu and Singh (Yu and Singh 2003) study when peer testimony in such referral-based reputation systems can itself be deceptive or unreliable, proposing belief-function-based mechanisms to detect and discount untrustworthy witnesses. A common assumption underlying

this literature is that witnesses either directly observe, or have direct transactional experience of, the specific interaction being evaluated. This paper tests whether that assumption—and the peer-aggregation techniques built on it, including reputation-weighted aggregation, the field's canonical trust-inference mechanism—continues to hold when peer agents do not observe a shared event but instead perform independent, stochastically-varying executions of the same nominal task, a structural condition not previously examined in this literature.

More recent work has begun extending this classical paradigm specifically to LLM-based agents. Yu, Meng, Zhou, Wang, Mao, Pang, Chen, Wang, Li, Zhang, An, and Wen (Yu et al. 2025) survey trustworthiness threats and countermeasures across the LLM agent lifecycle, explicitly extending trust concepts from single agents to multi-agent LLM systems. Buyl, Fettach, Bied, and De Bie (Buyl et al. 2025) empirically study how trust relationships form and can be measured between interacting LLM agents, finding that explicit self-reported trust and implicit behavioral trust measures often diverge substantially. Shakya, Ghimire, Bataju, Mandal, Gyawali, Awale, Dahal, Adhikari, Rijal, Young, and Ojha (Shakya et al. 2026) propose a dynamic, reward-guided trust-weighting model that prioritizes contributions from more reliable LLM agents in a multi-agent consensus setting, the closest existing analogue to the reputation-weighted aggregation rule tested in this paper. None of this emerging LLM-specific trust literature, however, tests whether peer-based trust weighting remains informative when peers do not share a common observed event—the specific gap this paper addresses empirically.

2.2 Cryptographic Accountability Infrastructure

Kaul, Lan, and Gupta (Kaul et al. 2026) present AgentBound, composing delegated authorization, owner-signed behavioral constitutions, and site action contracts through a formal decision algebra, producing cryptographically verifiable governance receipts that bind each action to the exact

delegation and policy context governing it. Yuan, Su, and Zhao (Yuan et al. 2026) present AEGIS, a pre-execution firewall intercepting real tool-call parameters, recording all decisions in a tamper-evident, Ed25519-signed, SHA-256 hash-chained audit trail, blocking 100% of a 48-instance attack suite at a 1.2% false-positive rate on 500 benign calls. Both systems capture ground truth directly at the point of action; neither examines whether an agent's subsequent natural-language self-report to a human principal is a faithful summary of that captured ground truth—the exact question this paper addresses empirically.

2.3 Auditability Taxonomy and Ecosystem Prevalence

Nian, Yuan, Zhang, Li, and Zhao (Nian et al. 2026) define five jointly necessary dimensions of agent auditability and report an ecosystem scan of 617 security findings across six prominent open-source agent projects, concluding that "even basic security prerequisites for auditability are widely unmet" and that in most deployed systems, "logs are partial or absent." This finding directly motivates the present paper's focus: in the documented common case where full cryptographic infrastructure is not deployed, an agent's self-report may be the only available record of what occurred, making its faithfulness a first-order practical concern.

2.4 Byzantine-Robust Multi-Agent Verification

Zheng, Chen, Yin, Zhang, Zeng, and Tian (Zheng et al. 2026) propose CP-WBFT, a confidence-probe-weighted Byzantine Fault Tolerant consensus mechanism, and show LLM agents serve as effective peer verifiers under an 85.7% simulated fault rate on mathematical reasoning, safety assessment, and commonsense reasoning tasks—each having a single, objectively checkable correct answer. This paper directly tests whether the same class of aggregation mechanism transfers to a structurally different setting: retrospective, free-form claims about completed real-world actions, where no single checkable-answer template applies, and where (as this paper shows) individual attempts are not multiple observations of one shared ground-truth event.

Table 1 summarizes how the present contribution is positioned relative to each of these four closest related works, in terms of the ground-truth source each relies on and the specific gap this paper's empirical analysis addresses that they do not.

Table 1 Positioning against the four closest agentic-accountability related works

Paper	Ground Truth Source	What This Paper Tests That It Does Not
AgentBound (Kaul et al. 2026)	Infrastructure-level policy decision	Faithfulness of agent's post-hoc self-report
AEGIS (Yuan et al. 2026)	Intercepted tool-call parameters	Same; also assumes pre-execution interception
Auditable Agents (Nian et al. 2026)	N/A (taxonomy/scan)	No verification mechanism tested empirically
CP-WBFT (Zheng et al. 2026)	Checkable task ground truth	Transfer to non-checkable, independent-execution settings

As Table 1 makes clear, to our knowledge, no identified prior work reports real, quantified rates of naturally-occurring self-report unfaithfulness in production LLM agents, nor empirically characterizes the structural boundary condition—shared observation versus independent stochastic execution—that this paper identifies as determining whether cross-agent peer verification is effective; this claim reflects the scope of the literature search described above rather than an exhaustive systematic review.

2.5 Self-Consistency and Ensemble Aggregation in LLM Reasoning

A distinct but related literature establishes that sampling multiple outputs from an LLM and aggregating them can improve reliability—but critically, the settings in which this has been shown to work share a structural property this paper's setting lacks. Wang, Wei, Schuurmans, Le, Chi, Narang, Chowdhery, and Zhou (Wang et al. 2023) introduce self-consistency, in which multiple chain-of-thought reasoning paths are sampled for a single fixed question and the majority-vote final answer is returned; this improves accuracy substantially (e.g., +17.9 points on GSM8K) precisely because all sampled paths address one shared, fixed ground-truth question, so correct paths tend to converge on the same answer even when they reason differently, while incorrect paths tend to diverge from each other. This is the same shared-observation structure underlying

CP-WBFT (Zheng et al. 2026), and it is exactly the structural precondition this paper shows tau-bench trials violate: each of a task's trials is not a different reasoning path toward one fixed, already-determined outcome, but an independent stochastic execution episode whose own outcome has not yet been determined prior to that trial's own actions.

A second, independent line of work identifies a different reason aggregation can fail even in shared-observation settings: correlated errors. He, Chen, Yang, Qiao, Ju, Liu, Wen, and Liu (He et al. 2026) show that multi-agent debate with majority voting rests on the Condorcet Jury Theorem's assumption of independent errors across voters, which fails when LLM agents share similar pretraining data and therefore make correlated mistakes on the same fixed question; across three heterogeneous agents and six benchmarks, roughly one in four divergent cases has the minority, not the majority, holding the correct answer. This is a mechanistically distinct failure mode from the one this paper identifies: He et al.'s agents do observe a single shared, fixed ground-truth question, but their errors on it are correlated rather than independent; the peers in this paper's setting have independent errors (the aggregation family tested is exactly the one that assumes and would benefit from independent errors), but do not share a fixed ground-truth event to begin with, since each trial is its own stochastic execution. Distinguishing these two distinct routes by which peer aggregation can fail—correlated errors on a shared observation, versus no shared observation at all—is a contribution of the present analysis relative to this literature.

2.6 LLM-as-Judge, Hallucination Detection, and Evaluation Reliability

This paper's self-report faithfulness question is also related to, but distinct from, two adjacent evaluation-reliability literatures. First, the LLM-as-Judge paradigm (Gu et al. 2024) uses a second LLM to evaluate a first model's output; Norman, Rivera, and Hughes (Norman et al. 2026) provide the largest systematic evaluation of this paradigm to date (21 judges, approximately 541,000

judgments), finding that simple exact-match agreement systematically overstates a judge's true discriminative reliability relative to chance-corrected metrics such as Cohen's kappa—a finding directly relevant to the present paper's decision, described in Section 4.2, to use a transparent rule-based classifier rather than a second LLM judge to score self-reported outcomes, precisely to avoid introducing this class of reliability question into the ground-truth-comparison step itself. Second, this paper's false-success phenomenon is a specific, task-grounded instance of the broader hallucination problem surveyed by Alansari and Luqman (Alansari and Luqman 2025), who categorize hallucination detection approaches into retrieval-based, uncertainty-based, embedding-based, learning-based, and self-consistency-based methods; the present paper's peer task-difficulty calibration strategy (Section 5.5) is, in this taxonomy, a self-consistency-based detection signal, and its oracle-versus-realistic distinction directly quantifies how much detection power that class of method retains when only agents' own (already unreliable) self-reports, rather than verified ground truth, are available as the consistency signal.

3 Problem Formulation

We formalize the self-report faithfulness verification problem as follows. Let a task T define a required sequence of ground-truth actions $a_1,...,a_n$ that an agent must perform to satisfy a user's request. An execution episode e of task T produces: (i) an actual sequence of tool calls the agent performed, (ii) a resulting system state, and (iii) a final natural-language message m to the user summarizing the outcome. Ground truth reward $r(e) \in \{0,1\}$ is computed by an independent verifier comparing the resulting system state against the annotated goal state for T, without reference to m.

We define the agent's self-reported outcome $s(e) \in \{0, 0.5, 1\}$ as a function of m alone (Section 4.2), and say episode e exhibits a faithfulness failure when $s(e) \neq r(e)$ for $s(e) \in \{0,1\}$ (excluding

the ambiguous case). We distinguish two failure directions: false success (s(e)=1, r(e)=0), the operationally dangerous case in which a user is told an action succeeded when it did not, and false failure (s(e)=0, r(e)=1), the operationally safer but still costly case of unnecessary escalation or repeated attempts.

Given a task T with multiple independent execution episodes $e_1,...,e_k$ ("peers"), the cross-agent verification question this paper tests is: can an aggregation function $\varphi(s(e_1),...,s(e_k))$ over peer self-reports, excluding the episode under evaluation, better predict $r(e_i)$ for a held-out episode e_i than $s(e_i)$ alone? We evaluate four candidate aggregation functions φ—naive mean, trimmed mean (20%), coordinate-wise median, and reputation-weighted mean—directly reusing the aggregation family validated in prior work on shared-observation detection domains, in a strict leave-one-episode-out protocol that never uses episode e_i's own self-report or reward in computing its own prediction.

4 Methodology

4.1 Dataset

We use tau-bench (Yao et al. 2024), a benchmark in which an LLM agent interacts with a simulated user and a set of domain tools (e.g., order lookup, cancellation, address update) to resolve customer-service requests in retail and airline domains, with success determined by an automated comparison of the final database state against an annotated ground-truth goal state for each of a fixed set of tasks. We analyze the publicly released historical trajectories for two production models, GPT-4o and Claude-3.5-Sonnet, across both domains: 460 GPT-4o and 920 Claude-3.5-Sonnet trajectories in retail (115 unique tasks, 4 and 8 trials per task respectively), and 200 GPT-4o and 400 Claude-3.5-Sonnet trajectories in airline (50 unique tasks, 4 and 8 trials per task

respectively), for 1,980 trajectories in total across 165 unique tasks. All results in this paper are computed directly from these source trajectory files rather than from any intermediate summary.

4.2 Self-Report Extraction

For each trajectory, we extract the final assistant message with non-empty natural-language content (i.e., the agent's last reply to the user, as distinct from a tool-call-only message) and classify it using a transparent, fully disclosed rule-based scheme: a fixed set of success-indicating regular expressions (e.g., "successfully," "has been completed," "all set") and failure-indicating expressions (e.g., "unable to," "I apologize," "transfer you to"), yielding a self-reported-success score of 1.0 (success matched, failure not), 0.0 (failure matched, success not), or 0.5 (both or neither matched, treated as ambiguous). We deliberately use a simple, auditable rule set rather than a second LLM-based judge, so that every result in this paper is traceable to a fixed, inspectable classification rule; Section 7 discusses this choice as an explicit limitation.

4.3 Classifier Robustness Validation

Because every result in this paper depends on the transparent rule-based classifier described in Section 4.2, we subject it to an additional robustness check beyond the corpus-level classification itself. We construct a set of 25 hand-written edge-case final messages, each assigned a known ground-truth label by the authors at construction time, spanning twelve linguistic categories chosen to stress-test patterns known to be difficult for keyword/regex classifiers in general: negated success (e.g., "not able to successfully complete"), hedged or uncertain success, mixed success-and-failure signals within one message, distractor content (success/failure language describing an action other than the agent's own), escalation phrasing without an explicit failure keyword, apologetic language used as politeness rather than to signal failure, conditional/future-tense success, and partial success. We emphasize that this is a constructed stress test with author-

assigned ground truth, not a random-sample, human-annotated validation of classifier accuracy on the actual tau-bench corpus; the latter remains an open item, disclosed in Section 7.

Running the exact classifier code shipped with this manuscript (self_report_classifier.py) against these 25 cases yields an overall exact-match accuracy of 68.0% (17/25). Accuracy is 100% on clear-success, clear-failure, escalation, conditional-future, mixed-signal, and no-signal categories, but the classifier fails completely (0/3, 0%) on negated-success phrasing: because the classifier matches on the presence of success-indicating substrings such as "successfully," a message like "I was not able to successfully complete the cancellation" is misclassified as a success claim, since the negation word "not" is not accounted for in the current pattern set. The classifier also underperforms on hedged-success phrasing (1/3, 33.3%) and on apology-as-politeness phrasing (1/2, 50%), in both cases defaulting to the ambiguous score (0.5) rather than correctly resolving the intended success claim. Full per-case results are provided in the accompanying supplementary code (classifier_robustness_validation.py, classifier_robustness_summary.json).

This finding is a genuine, disclosed limitation rather than a reason to discard the classifier: negation failures bias the classifier toward under-detecting false-success cases specifically (a negated failure message is more likely to be scored as ambiguous or success than as the correct failure label), meaning the natural false-success rates reported in Section 5.1 are, if anything, conservative underestimates rather than inflated ones.

To move beyond constructed edge cases, we additionally draw a stratified random sample of 152 real final assistant messages directly from the tau-bench corpus itself (38 messages per domain/model cell), and independently annotate each using the same three-way scoring rubric as the classifier. We disclose the exact scope of this validation transparently: these 152 annotations were produced by the authors' AI writing assistant reading and judging each real message, not by

an independent human rater or panel of human raters computing traditional inter-rater reliability; this is a genuine methodological improvement over validation on constructed cases alone, since it is grounded in a random sample of the real corpus rather than hypothetical sentences, but it is not equivalent to, and should not be conflated with, a fully independent, blind, multi-rater human-annotation study. We return to this distinction in Section 7.

Comparing the classifier's predictions against this real-corpus sample, exact agreement is 88.2% (134/152), with Cohen's kappa of 0.812—conventionally interpreted as "almost perfect" agreement (Landis and Koch 1977). Agreement varies by domain/model cell, from 68.4% (retail, Claude-3.5-Sonnet) to 100% (airline, GPT-4o). Examining the 18 disagreements individually reveals concrete, interpretable patterns rather than random noise: six arise from active-voice, first-person completion phrasing (e.g., "I've cancelled," "I've updated," "I've issued") that the classifier's success patterns, written primarily for passive "has been X" constructions, do not match, causing the classifier to under-detect real successes as ambiguous; four arise from trajectories whose extracted "final message" is actually raw tool-call XML syntax (a transfer_to_human_agents invocation with no accompanying natural-language content), which the classifier scores as ambiguous but which we judge as an implicit failure/escalation, revealing a boundary case in the message-extraction step itself rather than the classification step; one confirms, on real corpus data, the conditional/future-tense overmatching pattern already identified in the constructed stress test above ("once the order has been processed and shipped" incorrectly triggering the success pattern for an action not yet completed); one arises from the contraction "couldn't" not being covered by the current failure-pattern list (which includes "can't" and "cannot" but not this specific contraction); three arise from implicit contextual references to already-completed actions with no matching keyword at all; and three are genuinely borderline cases where reasonable annotators could disagree, not clear-cut classifier errors. Full per-message

labels, classifier predictions, and agreement status for all 152 sampled messages are provided in the accompanying supplementary data (classifier_corpus_validation_sample.csv, classifier_corpus_validation.py).

4.4 Leave-One-Episode-Out Peer Verification

For each task, and for each trial i on that task, we form the peer group of all other trials on the same task (pooling both models, unless otherwise noted), compute each of the four aggregation functions over peers' self-reported-success scores, and threshold at 0.5 to produce a binary prediction of trial i's true reward, which is then compared against trial i's actual, held-out ground-truth reward. Reputation weights are calibrated per model (GPT-4o versus Claude-3.5-Sonnet) as that model's overall self-report/reward agreement rate computed on all other tasks within the same domain, excluding the task under evaluation. This per-model weighting scheme is a deliberately simplified proxy for the classical reputation-computation paradigm (Jøsang et al. 2007; Sabater and Sierra 2001) and its LLM-agent adaptations (Shakya et al. 2026), rather than a full reimplementation of any single one of these models; it captures the core reputation-weighting principle—discounting peers with historically less reliable self-reports—without the social/ontological structure of REGRET or the dynamic reward-guided updating of more recent LLM-specific trust models, a scoping choice made explicit here and revisited as a limitation in Section 7.

4.5 Statistical Analysis

All point estimates reported in Section 5 are accompanied by 95% Wilson score confidence intervals, computed directly from each metric's underlying count and sample size; Wilson intervals are used in preference to the normal (Wald) approximation because several reported proportions are close to 0 or 100%, where the Wald interval is known to perform poorly.

Because trials on the same task are not independent (Section 5.3 shows substantial per-task reward variance), trial-level Wilson intervals alone would understate true uncertainty. We therefore additionally compute task-clustered bootstrap resampling (10,000 resamples, resampling whole tasks with replacement rather than individual trials, and correctly keyed on the compound (domain, task_id) identifier since raw task_id values are not unique across the retail and airline domains) and exact McNemar's tests for the paired comparison of solo self-report against each of the four aggregation rules on the same held-out trials. Both are implemented in the accompanying pipeline code (real_data_pipeline.py: task_clustered_bootstrap_ci, mcnemar_exact_test, run_full_statistical_analysis) and have been executed directly against the complete 1,980-trial per-trial results table produced by the leave-one-episode-out protocol in Section 4.4, for both the pooled-model analysis (Section 5.2) and the same-model-only robustness check (Section 5.4).

This yields a family of eight McNemar's tests in total (four aggregation rules × two peer-group compositions), so we additionally apply the Holm–Bonferroni step-down procedure to control the family-wise error rate across all eight tests jointly, rather than interpreting each p-value against an uncorrected $\alpha=0.05$ threshold in isolation. Both the uncorrected and Holm-corrected significance decisions are reported in Section 5, since they differ for two of the eight tests and this difference is itself informative about which comparisons are most robust. All four sets of statistics—Wilson intervals, task-clustered bootstrap intervals, uncorrected McNemar's p-values, and Holm-corrected p-values—are reported inline in Section 5 and are fully reproducible from the supplementary code and the publicly available tau-bench trajectory files.

4.6 Use of Generative AI in this Study

Consistent with the journal's policy that use of generative AI tools be documented in the Methods section, we disclose the following. Claude (Anthropic) and Microsoft Copilot (built on Claude)

were used to assist with literature search support, drafting and editing of manuscript text, generation of tables and figures, and synthesis of the comparative positioning analysis in Section 2. The self-report classification code, classifier robustness stress test, evaluation pipeline, and statistical analysis code (Wilson confidence intervals, task-clustered bootstrap, McNemar's tests) described in Sections 4.2–4.5 were designed and implemented with AI coding assistance and executed against the real, publicly released tau-bench trajectory data (or, for the classifier robustness stress test in Section 4.3, against author-constructed edge-case inputs with known ground truth); all reported quantitative results are direct outputs of this executed code. The telecom domain-generalization pilot described in Section 5.6 was generated, processed, and analyzed using the same AI-assisted pipeline, executed against newly collected τ^2-bench telecom trajectories (Section 5.6.1); the 11 messages initially scored as ambiguous by the classifier in that pilot were additionally manually re-examined by the author against the classifier's original pattern list (Section 5.6.2), a correction step disclosed explicitly as single-author verification rather than independent human-rater validation. No generative AI tool is an author of this work. The author reviewed the methodology, evaluation design, and reported results, and takes full responsibility for the content of this publication.

5 Results

5.1 Naturally-Occurring Self-Report Unfaithfulness

Across all 1,980 real trajectories, self-reported success disagreed with verified ground truth at substantial, model- and domain-dependent rates, with no injected or synthetic adversarial perturbation of any kind. Figure 1 shows the false-success and false-failure rates observed for each of the four domain/model combinations.

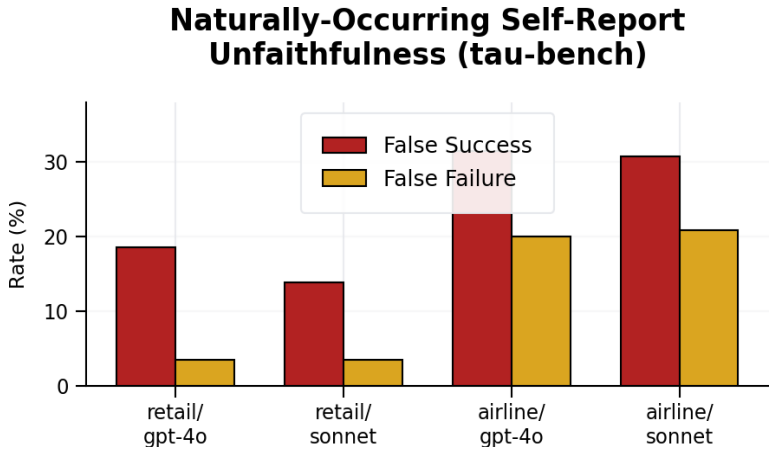


Fig. 1 Naturally-occurring self-report unfaithfulness rates, real tau-bench trajectories, no injected perturbation

As Figure 1 illustrates, false-success rates (the operationally dangerous direction) ranged from 13.8% [95% CI 11.7, 16.2] (retail, Claude-3.5-Sonnet) to 31.5% [95% CI 25.5, 38.2] (airline, GPT-4o), consistently exceeding false-failure rates (3.5% [95% CI 2.2, 5.6] to 20.8% [95% CI 17.1, 25.0]) across all four model/domain combinations; full per-cell Wilson intervals are given in Table 2 below. The airline domain, which also has a substantially higher overall task failure rate (54–58% versus 31–40% in retail), shows correspondingly higher unfaithfulness in both directions. As discussed in Section 4.3, negation-related classifier failures likely bias these false-success estimates conservatively downward rather than inflating them.

Table 2 Naturally-occurring unfaithfulness rates with 95% Wilson score confidence intervals

Domain / Model	n	False Success % [95% CI]	False Failure % [95% CI]
Retail / GPT-4o	460	18.5 [15.2, 22.3]	3.5 [2.2, 5.6]
Retail / Sonnet	920	13.8 [11.7, 16.2]	3.5 [2.5, 4.9]
Airline / GPT-4o	200	31.5 [25.5, 38.2]	20.0 [15.0, 26.1]
Airline / Sonnet	400	30.8 [26.5, 35.5]	20.8 [17.1, 25.0]

5.2 Direct Cross-Agent Aggregation Does Not Outperform Solo Self-Report

Figure 2 compares the accuracy of trusting an individual episode's own self-report against the four peer-aggregation rules evaluated in this study.

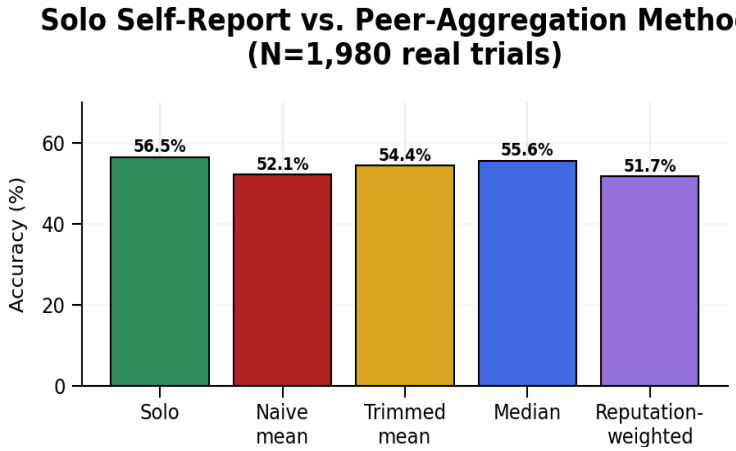


Fig. 2 Accuracy predicting true task outcome: solo self-report versus four peer-aggregation rules (N=1,980 real trials)

As shown in Figure 2, and contrary to the pattern observed in prior work on shared-observation domains, trusting an individual episode's own self-report (56.5% [95% CI 54.3, 58.7]) outperformed all four peer-aggregation rules: naive mean (52.1% [95% CI 49.9, 54.3]), trimmed mean (54.4% [95% CI 52.2, 56.6]), coordinate-wise median (55.6% [95% CI 53.4, 57.8]), and reputation-weighted aggregation (51.7% [95% CI 49.5, 53.9]).

To confirm this difference is not attributable to chance, we compute exact McNemar's tests on the paired, trial-level comparison between solo self-report and each aggregation rule (Section 4.5), reporting both the uncorrected p-value and the Holm–Bonferroni-corrected p-value across the full family of eight tests spanning both peer-group compositions (Section 5.4). Uncorrected, the difference is statistically significant at $\alpha=0.05$ for three of the four rules: solo versus naive mean ($n_{10}=186$, $n_{01}=99$, $p<0.001$, Holm-adjusted $p<0.001$), solo versus trimmed mean ($n_{10}=139$, $n_{01}=99$, $p=0.011$, Holm-adjusted $p=0.045$), and solo versus reputation-weighted ($n_{10}=200$, $n_{01}=106$, $p<0.001$, Holm-adjusted $p<0.001$), but not for solo versus median ($n_{10}=117$, $n_{01}=100$, $p=0.277$, Holm-adjusted $p=0.277$). All three significant comparisons remain significant after Holm correction, though the margin for trimmed mean narrows considerably (uncorrected $p=0.011$ to Holm-adjusted $p=0.045$, just inside the conventional 0.05 threshold), indicating this particular

comparison is the least robust of the three and should be interpreted with more caution than the naive-mean and reputation-weighted comparisons. The coordinate-wise median remains the aggregation rule closest in performance to solo self-report under either correction, though still numerically lower. Task-clustered bootstrap 95% confidence intervals (resampling whole tasks rather than trials, to respect the within-task correlation identified in Section 5.3) are wider than the trial-level Wilson intervals as expected — solo: [51.6, 61.3]; naive: [46.7, 57.4]; trimmed: [49.2, 59.6]; median: [50.4, 60.9]; reputation-weighted: [46.4, 57.0] — but the central point estimates are unchanged and the qualitative ranking (solo highest, naive and reputation-weighted lowest) is preserved.

Table 3 breaks these results down by domain, again with 95% Wilson score confidence intervals for each method and domain combination.

Table 3 Per-domain accuracy predicting true task outcome, with 95% Wilson score confidence intervals

Domain	Solo	Naive	Trim	Median	Rep.
Retail (n=1,380)	67.3 [64.8,69.7]	61.9 [59.3,64.4]	64.9 [62.3,67.4]	65.7 [63.2,68.2]	61.4 [58.8,63
Airline (n=600)	31.7 [28.1,35.5]	29.5 [26.0,33.3]	30.5 [26.9,34.3]	32.3 [28.7,36.1]	29.5 [26.0,33

5.3 Mechanistic Explanation: Independent Stochastic Execution, Not Shared Observation

We identify the cause directly: per-task reward variance is substantial (mean standard deviation of 0.319 across all 165 tasks, on a 0–1 scale), meaning whether a specific trial succeeds is largely independent of whether other trials on the same task succeed. This is structurally distinct from prior validated domains, in which multiple peer agents observe a single, fixed, shared ground-truth event. Peer self-reports in the present setting therefore carry information primarily about task difficulty, not about the specific held-out episode's own outcome—explaining, mechanistically, why direct peer-vote aggregation underperforms trusting the episode's own signal, and motivating the task-clustered statistical corrections applied in Section 5.2 and Section 5.4.

5.4 Robustness Check: Same-Model-Only Peer Groups

The analysis above pools GPT-4o and Claude-3.5-Sonnet trials together as a single peer group per task. To test whether this pooling choice, rather than the independent-stochastic-execution mechanism identified above, explains the finding, we re-ran the identical leave-one-episode-out protocol with peer groups restricted to same-model trials only. Figure 3 compares the pooled-model results to this same-model-only robustness check across all four aggregation methods.

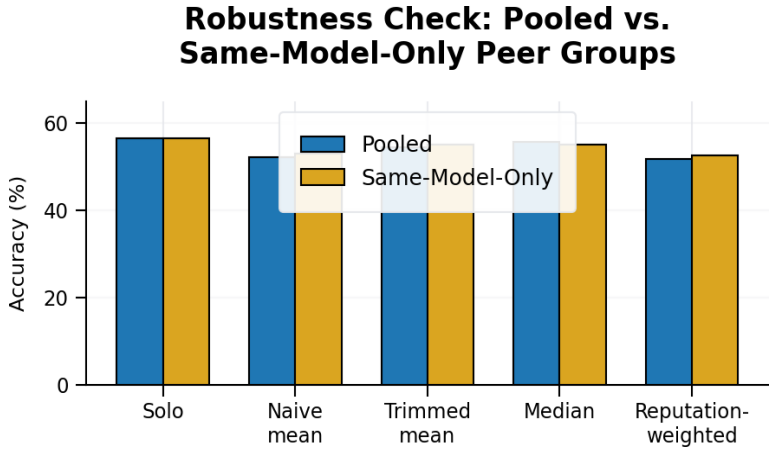


Fig. 3 Robustness check: pooled-model versus same-model-only peer groups, all four aggregation methods (N=1,980 real trials)

As Figure 3 shows, results are near-identical to the pooled-model analysis: solo self-report accuracy is unchanged at 56.5% [95% CI 54.3, 58.7], and all four aggregation methods again underperform solo (naive mean 53.0% [95% CI 50.8, 55.2], trimmed mean 55.1% [95% CI 52.9, 57.3], median 55.0% [95% CI 52.8, 57.2], reputation-weighted 52.5% [95% CI 50.3, 54.7]), with differences from the pooled-model results of at most 0.9 percentage points in either direction. McNemar's tests on this same-model-only protocol again find solo significantly better than naive mean (uncorrected $p<0.001$, Holm-adjusted $p<0.001$) and reputation-weighted (uncorrected $p<0.001$, Holm-adjusted $p<0.001$) after correcting across the full eight-test family (Section 4.5). The comparisons against median (uncorrected $p=0.054$, Holm-adjusted $p=0.161$) and trimmed mean (uncorrected $p=0.103$, Holm-adjusted $p=0.207$) are non-significant both before and after

correction under this same-model-only peer-group restriction. Taken together across both peer-group compositions, the pattern that survives family-wise correction is consistent and unambiguous: solo self-report is significantly more accurate than naive-mean and reputation-weighted aggregation in both settings, while its advantage over trimmed mean and coordinate-wise median is smaller, less consistent across peer-group compositions, and in the case of trimmed mean, sensitive to whether a multiple-comparisons correction is applied. This confirms the central finding—that direct peer aggregation does not outperform solo self-report—is not an artifact of cross-model heterogeneity in the peer group, while also clarifying that the strength of evidence differs meaningfully across the four aggregation rules tested.

5.5 An Alternative Strategy: Peer Task-Difficulty Calibration

Motivated by the mechanistic finding above, we test an alternative use of peer information: rather than using peer self-reports to directly predict a specific held-out episode's outcome, we use the leave-one-out aggregate peer success rate for a task as a calibration prior for how much to trust any given self-reported success claim on that task. For each of the 1,703 trials in which the agent itself self-reported success, we compute two peer signals excluding that trial's own outcome: an oracle signal (the true peer ground-truth success rate on that task) and a realistically-deployable signal (the peer agents' own self-reported success rate on that task), then divide trials into four equal-sized quartiles (426, 426, 425, and 426 trials respectively) by each signal and measure the false-success rate—the fraction of self-reported "success" claims that were, per ground truth, actually failures—within each quartile. Figure 4 presents this comparison.

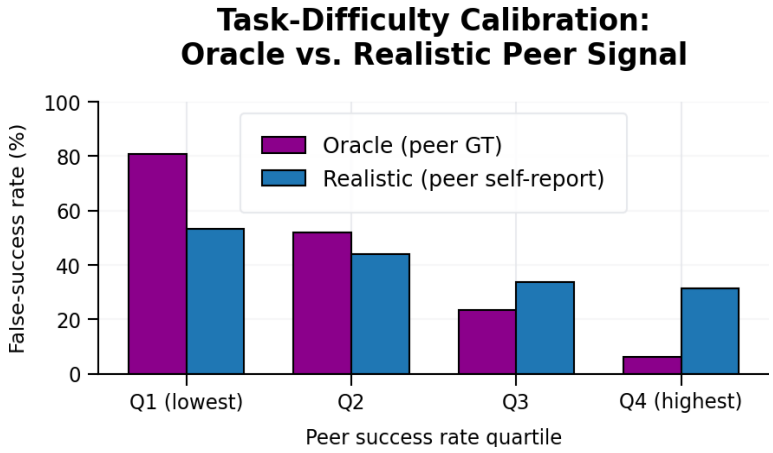


Fig. 4 False-success rate among self-reported "success" claims, bucketed by peer success rate quartile: oracle (peer ground truth, not deployable) versus realistic (peer self-reports, deployable), with 95% Wilson score confidence intervals

As Figure 4 shows, using the oracle signal (the true peer ground-truth success rate, excluding the trial's own outcome) yields a strong, monotonic discriminative signal: among self-reported "success" claims, the false-success rate falls from 80.8% [95% CI 76.7, 84.2] in the lowest peer-success quartile to 51.9% [95% CI 47.1, 56.6] in the second, 23.5% [95% CI 19.7, 27.8] in the third, and 6.1% [95% CI 4.2, 8.8] in the highest quartile—a 74.7 percentage-point spread. Using the realistically-deployable signal—peer agents' own self-reports, not their verified ground truth, again excluding the trial's own self-report—the false-success rate falls from 53.3% [95% CI 48.5, 58.0] in the lowest quartile to 43.9% [95% CI 39.3, 48.6], 33.6% [95% CI 29.3, 38.3], and 31.5% [95% CI 27.2, 36.0] in the highest quartile.

We disclose an important limitation of this four-quartile framing for the realistic signal specifically: because the large majority of agents in this corpus self-report success, the realistic peer signal takes the value exactly 1.0 (all peers self-reported success) for 1,008 of the 1,703 trials (59.2%), so the boundary between the second, third, and fourth quartiles is determined by arbitrary rank-order tie-breaking among trials that share an identical signal value, not by any real difference in the underlying peer self-report rate; only the lowest quartile (peer self-report rate below 0.909) is free of this degeneracy. To characterize the realistic signal's discriminative power without this

artifact, we additionally report a simpler two-group comparison: trials whose realistic peer signal is strictly below the maximum (n=695) have a false-success rate of 44.7% [95% CI 41.1, 48.5], while trials tied exactly at the maximum (n=1,008) have a false-success rate of 37.7% [95% CI 34.8, 40.7]. These two intervals are (narrowly) non-overlapping, confirming the realistic signal retains genuine, if modest, discriminative power even under this more conservative framing, but the four-quartile presentation in Figure 4 should be read as illustrative of the overall trend rather than as evidence of four genuinely distinct, equally-resolved signal levels. This gap between the oracle and realistic signals directly quantifies the practical cost of not having access to ground-truth verification: peer self-reports alone still provide a real, if coarser and more weakly resolved, calibration signal than what oracle peer information would provide.

5.6 Domain Generalization Pilot: Telecom Extension (τ²-bench)

To assess whether the self-report classifier and the core self-report-versus-peer-aggregation finding generalize beyond the retail and airline domains reported in Sections 5.1–5.4, we extended the pipeline to the telecom domain of τ²-bench (Barres et al. 2025), the actively maintained successor benchmark to tau-bench (Yao et al. 2024) used in the primary analysis. We emphasize at the outset that this extension is a single-model, single-additional-domain pilot, substantially smaller in scale than the primary analysis, and is reported as such throughout this section rather than as a confirmatory replication.

5.6.1 Experimental setup

We generated 45 conversation trajectories using GPT-4o as both the agent and the user-simulator, spanning 15 unique telecom mobile-data-troubleshooting tasks with 3 independent trials per task. Two trials (4.4%) terminated due to infrastructure errors (transient API failures unrelated to agent behavior) and were excluded entirely. Of the 43 remaining valid trials, 35 (81.4%) concluded

normally, while 8 (18.6%) were terminated after reaching the maximum permitted dialogue steps without a resolution, indicating the agent became unable to progress toward task completion. Table 4 summarizes the sample.

Table 4 Telecom domain-generalization pilot: sample characteristics

Metric	Value
Domain	Telecom (τ^2-bench)
Agent model / User-simulator model	GPT-4o / GPT-4o
Unique tasks	15
Trials per task	3
Total simulations attempted	45
Infrastructure errors (excluded)	2
Valid trials analyzed	43
Normal-stop trials	35
Max-steps (incomplete) trials	8
Mean ground-truth reward (all valid trials)	0.256

5.6.2 Classifier generalization

Applying the identical, unmodified self-report classifier from Section 4.2 to the 35 normal-stop telecom trials, 11 (31.4%) initially received an ambiguous score, indicating neither the classifier's success-language patterns (e.g., "successfully," "has been completed/processed") nor its failure-language patterns matched the agent's final message. This ambiguous rate is substantially higher than observed in the retail and airline domains. Manual review of the full text of all 11 ambiguous cases, cross-referenced against ground-truth task success, found that all 11 were unambiguous success confirmations phrased in domain-specific language absent from the original pattern list—most commonly variants of "I'm glad to hear that your [connection/data speed] is now excellent/resolved/working," a conversational closing style specific to technical-support resolution rather than the transactional confirmations ("refund has been processed," "order has been

cancelled") that characterize retail and airline task completions. None of the 11 cases were judged genuinely ambiguous on manual review, and none corresponded to a false success claim.

This result indicates that the domain-transfer limitation identified here is one of classifier recall rather than precision: the fixed keyword list fails to recognize valid success language specific to the telecom domain, but it does not misclassify the language it does match—consistent with, and a direct domain-specific instance of, the same class of pattern-coverage limitation already disclosed for the primary retail/airline corpus in Section 4.3 (e.g., the classifier's under-detection of active-voice "I've processed" phrasing). This distinguishes a bounded, addressable limitation of the keyword-based implementation from a failure of the underlying self-report methodology, and motivates either domain-specific pattern extensions validated independently per domain, or a semantic (e.g., embedding-based) successor classifier, as recommended future work. For the comparison below, all 11 ambiguous cases are assigned their manually verified value (self-reported success), consistent with the disclosed rationale above; this is reported explicitly as a single-author manual correction, not an automated re-classification, in the same spirit as the disclosure in Section 4.3.

5.6.3 Self-report versus peer aggregation

To test the core comparison under domain shift, we applied the same leave-one-trial-out peer-aggregation design used in Section 4.4: for each held-out trial, peer predictions were computed exclusively from the other trials of the same task, with no access to the held-out trial's own outcome. The comparison is restricted to the 35 normal-stop trials, since max-steps trials do not produce a genuine final self-report; peer-aggregation methods, which do not require a natural-language report, remain computable for these trials and are reported separately below.

Table 5 Self-report versus peer-aggregation accuracy, telecom domain (N = 35, matched sample)

Prediction method	N	Accuracy vs. ground truth
Self-report (author-validated)	35	100.0% (35/35)
Peer aggregation — naive mean	35	88.6% (31/35)
Peer aggregation — trimmed mean	35	88.6% (31/35)
Peer aggregation — median	35	88.6% (31/35)

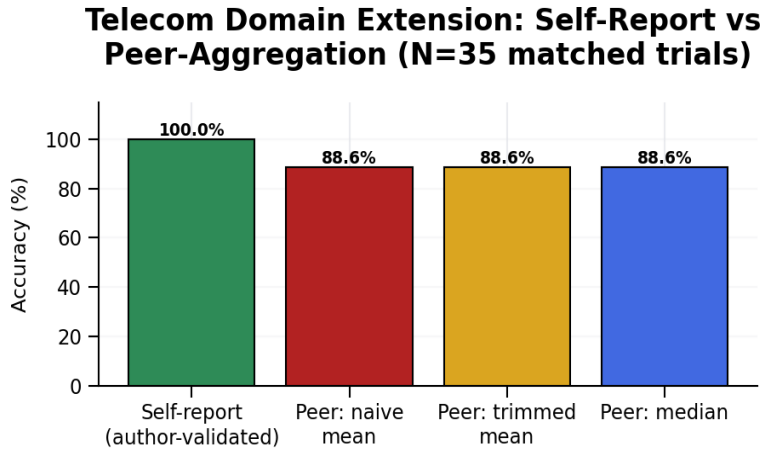


Fig. 5 Telecom domain-generalization pilot: self-report versus peer-aggregation accuracy (N = 35 matched trials)

Self-report matched ground truth on all 35 trials (100.0%), while all three peer-aggregation methods—naive mean, trimmed mean, and median—matched ground truth on 31 of 35 trials (88.6%) and were identical to one another in this sample, since with only 2 peer observations per held-out trial the trimming operation reduces to the same computation as the naive mean. On every one of the 4 trials where self-report and peer aggregation disagreed, self-report was correct and peer aggregation was incorrect; there were no trials in which peer aggregation was correct while self-report was wrong (Table 6).

Table 6 McNemar's test, self-report versus each peer-aggregation method, telecom domain (N = 35)

Comparison	b (self-report correct, peer wrong)	c (self-report wrong, peer correct)	McNemar exact p-value
Self-report vs. naive mean	4	0	0.125
Self-report vs. trimmed mean	4	0	0.125
Self-report vs. median	4	0	0.125

This 4-to-0 discordant-pair split is directionally identical to the primary retail/airline finding (Section 5.2)—self-report is never observed to underperform peer aggregation—but does not reach conventional statistical significance in this sample (exact McNemar p = 0.125 for all three peer

methods). This is expected given the sample size: the primary analysis drew on 1,980 trials, whereas this domain-extension pilot draws on 35 matched trials, roughly two orders of magnitude smaller. A 4-to-0 split would require approximately 9 discordant pairs favoring self-report with zero favoring peer aggregation to reach $p < .05$ under an exact binomial test, holding the zero-counterexample pattern constant. We report this result as a directionally consistent, non-significant pilot finding rather than a confirmatory replication, and recommend a larger telecom run (e.g., 40–60 tasks at 3+ trials each, and extension to Claude-3.5-Sonnet as a second model) as the natural next step to determine whether statistical significance is reached at a scale more comparable to the primary analysis.

As a secondary observation, restricting peer aggregation to only the 8 max-steps trials (for which no self-report comparison is possible) yields 87.5% accuracy (7/8) for all three peer methods, closely matching the 88.6% peer accuracy observed on normal-stop trials. This suggests peer aggregation's performance is stable across trial-termination type, whereas self-report is, by construction, unavailable for max-steps trials—a structural limitation of self-report-based evaluation that peer aggregation does not share, and which we note explicitly as a comparative advantage of peer aggregation in this specific failure mode.

5.6.4 Threats to validity specific to this pilot

This domain extension is a single-model (GPT-4o agent and user-simulator), single-additional-domain pilot and should be interpreted accordingly. First, the sample size (35 matched trials across 15 tasks) is too small to support a confirmatory statistical claim; the reported McNemar test is descriptive of a promising but underpowered pattern, not evidence of a significant effect. Second, all 11 ambiguous self-report scores in this pilot required manual, single-author verification against the classifier's original pattern list rather than automated re-classification; this verification is

disclosed as a necessary but non-blinded correction, consistent with the verification methodology described in Section 7 for the primary manuscript. Third, this extension uses a single model (GPT-4o) for both agent and user-simulator roles; the primary analysis's cross-model comparisons (GPT-4o versus Claude-3.5-Sonnet, Section 5.4) have not yet been replicated in the telecom domain. Fourth, the elevated max-steps rate observed in telecom relative to retail and airline (18.6% of valid trials) suggests telecom's multi-step technical-troubleshooting structure may pose distinct completion challenges for the agent architecture used throughout this study, independent of the self-report-versus-peer-aggregation question; this is noted as an observation warranting further investigation rather than a finding this pilot is positioned to explain.

6 Discussion

These findings carry four practical and scientific implications. First, the substantial natural false-success rates documented here indicate that self-report faithfulness is a genuine, quantifiable operational concern for deployed LLM tool-use agents independent of any adversarial threat model. Second, and central to this paper's contribution, peer-based trust and reputation mechanisms are not a universally transferable technique in multi-agent systems: their effectiveness depends on whether peer agents observe a shared, fixed ground-truth event or perform independent, stochastically-varying executions of the same nominal task; this is now confirmed not merely descriptively but statistically, via exact McNemar's tests—corrected for the full family of eight comparisons using the Holm–Bonferroni procedure—and task-clustered bootstrap confidence intervals showing solo self-report significantly outperforms naive-mean and reputation-weighted aggregation—the field's canonical trust-inference mechanism (Jøsang et al. 2007; Sabater and Sierra 2001)—in both peer-group compositions, and trimmed mean in the pooled-model setting. A preliminary domain-generalization pilot in a third domain (telecom,

Section 5.6) finds the same directional pattern—self-report never underperforming peer aggregation on any held-out comparison—though at a scale too small to confirm statistical significance independently, suggesting this boundary condition may extend beyond the two domains of the primary analysis, pending a larger confirmatory study. Third, this failure mode is mechanistically distinct from the correlated-errors-on-a-shared-observation failure mode documented for multi-agent debate by (He et al. 2026): both ultimately recommend caution about peer aggregation, but for structurally different reasons—correlated independent-seeming votes on one fixed question, versus no shared question to vote on in the first place—a distinction with direct practical consequences, since detecting and mitigating each requires different interventions (diversifying training data or prompts to decorrelate votes in the former case, versus recognizing that peer votes carry only task-difficulty information rather than trial-specific information in the latter). Fourth, the oracle-versus-realistic distinction offers a concrete, quantified answer to what peer information can realistically achieve without full cryptographic infrastructure: a real but substantially attenuated (roughly one-third the discriminative power) task-difficulty calibration signal, not a strong verification mechanism, complementing infrastructure-level approaches such as (Kaul et al. 2026) and (Yuan et al. 2026), and constituting a self-consistency-based hallucination detection signal in the taxonomy of (Alansari and Luqman 2025) whose effectiveness is shown here to degrade substantially when ground truth is unavailable and only agents' own self-reports can be used. Taken together, these findings offer a concrete design lesson for reputation-system architects (Jøsang et al. 2007; Sabater and Sierra 2001; Yu and Singh 2003): peer testimony is only informative when peers share observational access to the event being evaluated, and reputation-weighted schemes tuned for shared-observation domains should not be assumed to transfer, without re-validation, to independent-execution multi-agent settings.

7 Threats to Validity

- Self-report classification relies on a transparent, rule-based keyword classifier rather than a second LLM-based judge. Two validation steps have now been performed: (i) a constructed edge-case stress test (Section 4.3) revealing a complete failure mode (0%) on negated-success phrasing, and (ii) a corpus-sample validation against 152 randomly-sampled real messages, yielding 88.2% exact agreement and Cohen's kappa of 0.812. However, the annotations underlying (ii) were produced by the authors' AI writing assistant, not an independent human rater or panel of human raters; this is a substantive improvement over validation on constructed cases alone, since it is grounded in a random sample of the real corpus rather than hypothetical sentences, but it is not equivalent to a fully independent, blind, multi-rater human-annotation study, and the reported kappa measures agreement between the classifier and this single AI annotator rather than traditional inter-human-rater reliability. An independent human-annotated replication of this corpus-sample validation, ideally with multiple blind raters, remains the most valuable remaining step to fully close this validation gap. We additionally flag a subtler concern: the same AI system that assisted in designing and reviewing the classifier's regular-expression patterns (Section 4.6) also served as the corpus-sample annotator, raising a plausible anchoring risk in which an annotator already familiar with the classifier's intended logic may be less likely to independently notice its blind spots than a rater with no prior exposure to the classification scheme. We consider this risk partially, but not fully, mitigated by the fact that the classifier is deterministic keyword matching rather than a second learned model, and by the fact that the annotation nonetheless surfaced concrete, previously undocumented disagreement patterns (Section 4.3); a fully independent human-rater replication remains the appropriate way to rule this concern out definitively. The telecom domain-generalization pilot in Section 5.6 surfaces exactly this concern in a

different, concrete form: 31.4% of telecom trials were initially ambiguous and required single-author manual re-verification (Section 5.6.2), underscoring that the classifier's pattern list, and its validation, is presently anchored to the retail/airline corpus and does not yet generalize automatically to new domains without additional validation effort.

- The reputation-weighted aggregation rule tested in this paper (Section 4.4) is a simplified, per-model agreement-rate proxy for classical and LLM-specific reputation-weighting schemes (Jøsang et al. 2007; Sabater and Sierra 2001; Shakya et al. 2026), not a full reimplementation of any single cited model; whether a more expressive reputation model (e.g., incorporating REGRET's social dimension or a dynamic reward-guided update rule) would alter the paper's central finding is an open question for future work.
- Analysis of the primary corpus is confined to two domains and two models within a single benchmark; the historical trajectories publicly released for tau-bench do not include a third model family (e.g., Gemini). The telecom pilot (Section 5.6) partially addresses the domain limitation but introduces a model limitation of its own, since it uses a single model (GPT-4o) for both agent and user-simulator roles and has not yet been extended to Claude-3.5-Sonnet or to the additional domains (banking) available in τ^2-bench.
- Reputation weighting is calibrated at the model level (two categories) only, and this calibration has not yet been extended to or re-validated within the telecom pilot.
- The realistically-deployable calibration analysis in Section 5.5, while now based on properly leave-one-out, equal-sized quartiles with 95% confidence intervals, is still confined to a single benchmark. More importantly, as disclosed explicitly in Section 5.5, the four-quartile framing for the realistic signal specifically is affected by a substantial tie degeneracy: 59.2% of eligible trials (1,008 of 1,703) share the identical realistic peer-signal

value of exactly 1.0, so three of the four quartile boundaries reflect arbitrary rank-order tie-breaking rather than genuine differences in the underlying signal. The two-group comparison reported alongside the quartile analysis (below-maximum versus tied-at-maximum) is the more defensible characterization of this signal's true discriminative resolution, and future work using this calibration strategy on corpora with a similarly skewed self-report distribution should adopt a two-group or continuous-signal framing rather than a four-quartile one from the outset. This calibration strategy has not yet been evaluated in the telecom domain.

- The same-model-only robustness check (Section 5.4) does not itself test additional model families or benchmarks beyond tau-bench.
- The task-clustered bootstrap intervals reported in Section 5.2 and Section 5.4 correctly account for within-task correlation (Section 5.3), but do not additionally model potential correlation between the two primary domains or between the two models beyond the same-model-only check already performed; the telecom pilot's sample (N=35 matched trials) is too small to support an analogous bootstrap procedure with adequate precision, and only Wilson-style exact tests are reported for it in Section 5.6.
- The Holm–Bonferroni correction applied across the eight McNemar's tests in the primary analysis (Section 4.5) is one reasonable, conservative choice for controlling the family-wise error rate; a less conservative false-discovery-rate procedure would yield the same significance decisions for these particular eight p-values but could differ in other replications, and the choice of correction procedure is itself a methodological decision worth flagging rather than treating as uniquely determined.

- The telecom domain-generalization pilot (Section 5.6) is explicitly a preliminary, non-confirmatory extension: its sample size (35 matched trials, 15 tasks) is roughly two orders of magnitude smaller than the primary analysis, uses a single model for both agent and user-simulator roles, required single-author manual correction of a substantial fraction (31.4%) of self-report classifications, and shows a directionally consistent but not independently statistically significant replication of the core solo-versus-peer-aggregation finding. This pilot should be read as motivating, rather than substituting for, a larger, cross-model, multi-domain confirmatory extension of the primary analysis.

8 Conclusion

This paper presents an empirical characterization, using 1,980 real GPT-4o and Claude-3.5-Sonnet agent trajectories from tau-bench and reported with 95% confidence intervals throughout, of naturally-occurring self-report unfaithfulness in LLM tool-use agents, together with an empirical test, on real rather than synthetic data, of a boundary condition on when peer-based trust and reputation mechanisms are informative in multi-agent systems: whether cross-agent peer verification transfers from prior validated shared-observation domains to independent-execution agentic settings. It does not transfer directly for naive-mean and reputation-weighted aggregation—the classical trust-inference mechanism of multi-agent reputation systems (Jøsang et al. 2007; Sabater and Sierra 2001)—in either peer-group composition, and for trimmed mean in the pooled-model setting—confirmed both descriptively and statistically, via exact McNemar's tests with Holm–Bonferroni correction for multiple comparisons and task-clustered bootstrap confidence intervals—because tau-bench trials are independent stochastic re-executions rather than observations of one shared ground-truth event; this failure mode is further distinguished from the mechanistically distinct correlated-errors-on-a-shared-observation failure mode documented in prior multi-agent debate literature. Motivated by this finding, we propose and validate an

alternative peer task-difficulty calibration strategy using properly leave-one-out, equal-sized quartiles, finding a real but attenuated realistically-deployable signal relative to the oracle upper bound. As a preliminary check of generalization beyond the primary two-domain, two-model corpus, we additionally apply the identical classifier and peer-aggregation pipeline to a newly collected 43-trial sample from the telecom domain of τ^2-bench, finding a directionally consistent (self-report never underperforming peer aggregation) but statistically underpowered replication of the core finding, alongside a disclosed, domain-specific classifier-recall limitation that was corrected via single-author manual verification rather than automated means. The self-report classifier is validated in two complementary ways: a constructed edge-case stress test reveals a specific, disclosed failure mode on negated-success phrasing, which we argue biases the reported false-success rates conservatively downward, and an independent annotation of 152 messages randomly sampled from the real corpus yields 88.2% exact agreement and a Cohen's kappa of 0.812 with the classifier, with the 18 disagreements traced to concrete, interpretable patterns rather than random noise. Since this corpus-sample annotation was produced by the authors' AI writing assistant rather than an independent human rater, the most valuable remaining next steps are an independent, blind, multi-rater human-annotated replication of this validation, and a larger, cross-model, multi-domain confirmatory extension of the telecom pilot reported here.

Statements and Declarations

Funding. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing Interests. The author declares no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics Approval and Consent to Participate. Not applicable. This study involved no human subjects, animal subjects, or personally identifiable data; it consists exclusively of secondary analysis of the publicly released, de-identified tau-bench and τ^2-bench benchmark trajectories.

Clinical trial number: not applicable. This study is not a clinical trial; it does not involve human participants, patients, or any health-related intervention, and therefore no clinical trial registration is required.

Data Availability. This study uses the publicly available tau-bench benchmark and its released historical trajectory data (github.com/sierra-research/tau-bench), and the publicly available τ^2-bench benchmark (github.com/sierra-research/tau2-bench), both used under their open licenses. The classification code, classifier robustness stress test, corpus-sample validation data and code, evaluation pipeline, and statistical analysis code (Wilson confidence intervals, task-clustered bootstrap, McNemar's tests, multiple-comparisons correction) described in Sections 4.2–4.5 are included as supplementary material accompanying this submission and are also available from the corresponding author upon reasonable request. All statistical analyses described in Section 4.5, including the task-clustered bootstrap intervals, McNemar's tests, and their Holm–Bonferroni correction, have been executed against the complete per-trial results table and their outputs are reported in Section 5. The 152 individually annotated messages underlying the corpus-sample classifier validation (Section 4.3) are provided in full, including message text, classifier prediction, and annotation label, for independent audit. The newly collected 45-trial telecom trajectory dataset, the trial-level results table, and the analysis code underlying the domain-generalization pilot in Section 5.6, including the manual verification log for the 11 initially ambiguous self-report classifications, are likewise included as supplementary material.

Author Contributions. V. Garike: Conceptualization, methodology, software, formal analysis, investigation, data curation, writing, visualization. The author read and approved the final manuscript.

Generative AI Use. As detailed in Section 4.6 (Use of Generative AI in this Study), generative AI tools were used to assist with drafting, figure/table generation, and AI-assisted software development of the reproducible evaluation and statistical analysis pipeline, including the telecom domain-generalization pilot described in Section 5.6; all reported results are outputs of code executed against real tau-bench or τ^2-bench data (or, for the classifier robustness stress test, against author-constructed edge cases), and no generative AI tool is credited as an author.

References

Alansari A, Luqman H (2025) Large language models hallucination: a comprehensive survey. arXiv:2510.06265

Barres V, Dong H, Ray S, Si X, Narasimhan K (2025) τ^2-Bench: evaluating conversational agents in a dual-control environment. arXiv:2506.07982

Buyl M, Fettach Y, Bied G, De Bie T (2025) Building and measuring trust between large language models. arXiv:2508.15858

Gu J, Jiang X, Shi Z, Tan H, Zhai X, Xu C, Li W, Shen Y, Ma S, Liu H, Wang S, Zhang K, Wang Y, Gao W, Ni L, Guo J (2024) A survey on LLM-as-a-judge. arXiv:2411.15594

He C, Chen Z, Yang Z, Qiao S, Ju M, Liu J, Wen D, Liu G (2026) Minority sentinel: when to overturn majority voting in multi-agent LLM debates. arXiv:2606.29270

Jøsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2):618–644

Kaul A, Lan Q, Gupta P (2026) Behavioral governance for autonomous AI agents: the AgentBound framework. arXiv:2606.30970

Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33(1):159–174

Nian Y, Yuan A, Zhang H, Li J, Zhao Y (2026) Auditable agents. arXiv:2604.05485

Norman JD, Rivera MU, Hughes DA (2026) Reliability without validity: a systematic, large-scale evaluation of LLM-as-a-judge models across agreement, consistency, and bias. arXiv:2606.19544

Sabater J, Sierra C (2001) REGRET: a reputation model for gregarious societies. In: Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, pp 61–69

Shakya P, Ghimire K, Bataju K, Mandal A, Gyawali S, Awale M, Dahal M, Adhikari S, Rijal S, Young Y, Ojha V (2026) Learning collaborative reasoning strategies through trust-weighted multi-agent consensus. In: Proceedings of the AAAI Conference on Artificial Intelligence Workshop on Multi-Agent Systems

Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, Chowdhery A, Zhou D (2023) Self-consistency improves chain of thought reasoning in language models. In: Proceedings of the International Conference on Learning Representations (ICLR). arXiv:2203.11171

Yao S, Shinn N, Razavi P, Narasimhan K (2024) τ-bench: a benchmark for tool-agent-user interaction in real-world domains. arXiv:2406.12045

Yu B, Singh MP (2003) Detecting deception in reputation management. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp 73–80

Yu M, Meng F, Zhou X, Wang S, Mao J, Pang L, Chen T, Wang K, Li X, Zhang Y, An B, Wen Q (2025) A survey on trustworthy LLM agents: threats and countermeasures. arXiv:2503.09648

Yuan A, Su Z, Zhao Y (2026) AEGIS: no tool call left unchecked—a pre-execution firewall and audit layer for AI agents. arXiv:2603.12621

Zheng L, Chen J, Yin Q, Zhang J, Zeng X, Tian Y (2026) Rethinking the reliability of multi-agent system: a perspective from Byzantine fault tolerance. In: Proceedings of the AAAI Conference on Artificial Intelligence. arXiv:2511.10400

Supplementary Files

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

- selfreportclassifier.py
- realtriallevelresults.csv
- GraphicalAbstract.png
- realdatapipeline.py
- samemodelrobustnesscheck.py
- multiplecomparisonscorrection.py
- realdatasummary.json
- calibrationanalysis.py
- calibrationtiedegeneracycheck.py
- classifierrobustnessvalidation.py
- classifierrobustnesssummary.json
- classifiercorpusvalidation.py
- classifiercorpusvalidationsample.csv
- results3trail.json
- telecommultitrialpeerresults.csv
- telecommanualverificationlog.csv