DO CLAIMS, EVIDENCE, AND ACTIONS ALIGN? EVIDENCE-LINKED CONSISTENCY CHECKING IN MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Evidence can appear in a multi-agent log without reaching the agent whose claim depends on it. A factually correct claim can therefore lack task-required support, even when no statements contradict one another. We formulate consistency as satisfaction of task-defined obligations relating evidence, claims, and actions. This definition accounts for which sources an agent may use and the context in which it acts. We introduce ECOCHECK (Evidence-linked Consistency Checking), which combines graph-based retrieval, source-linked fact extraction, symbolic checking, and diagnostic feedback. We evaluate detection on perturbed traces and task outcomes in tool collaboration, coordination, and social simulation. Across 120 recorded perturbation outcomes, ECoCheck records 38 detections versus 18 for a transcript-order comparator, Linear, with the largest difference on unsupported claims (17/30 versus 6/30). Under the full treatment, macro-average overall goal success in tool collaboration is 20.00 percentage points above the baseline without verification feedback, and coordination utility is higher in all nine task–scale cells. These comparisons assess checking and feedback together and do not isolate the effect of checking. The selected evidence, encoded requirements, and diagnostics show what each check is based on.

1 INTRODUCTION

Multi-agent systems distribute observations, responsibilities, and decisions across agents (Shu et al., 2024; Mahmud et al., 2025; Zhu et al., 2025). A record in the interaction log need not have reached the agent whose claim depends on it. Consider a coordinator that declares an item available and refundable. The inventory agent has reported availability, but the policy agent has observed the refund conditions without delivering its report. Both claims may be true, yet the refund claim violates a task rule requiring receipt of the policy report. Factual correctness and the absence of contradiction do not establish that the coordinator had the required support.

The checking problem is therefore to determine whether recorded claims and actions satisfy the requirements imposed by the task. We call this *task-relative consistency*. The requirements concern three relations: evidence–claim support, evidence–decision compatibility, and claim–execution correspondence (Figure 1). They ask whether a claim has the required support, a decision is permitted, or a completion statement has its required result. This differs from seeking answer agreement through aggregation or debate (Wang et al., 2023; Du et al., 2024): agents may legitimately disagree while satisfying their obligations. Consistency guarantees neither factual truth nor task success.

Checking these relations requires preserving the context that determines whether an obligation holds. A summary such as "refundable = yes" cannot distinguish a delivered policy report from an observation that never reached the coordinator. Section 2.2 formalizes this information requirement. Provenance and failure-analysis methods retain execution dependencies (Wu et al., 2026; Wang et al., 2026b; Rafi et al., 2026), while formal oversight checks behavior against compiled specifications (Zhou et al., 2026). For task-relative consistency, the checker must connect the two: select relevant interaction records and check the applicable requirements against their content and context.

We introduce ECOCHECK (Evidence-linked Consistency Checking) for this purpose. A typed interaction graph retains source and communication context; graph queries retrieve evidence for

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

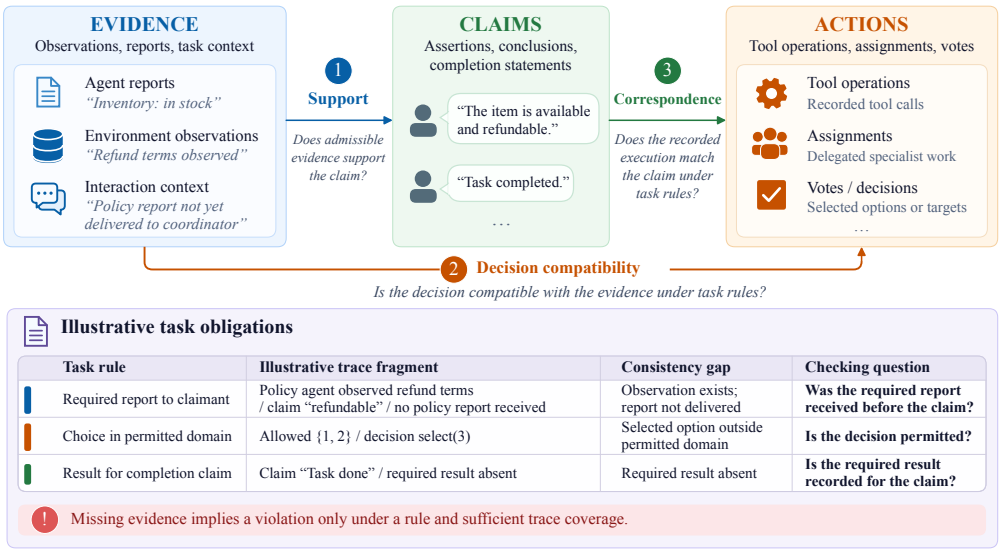


Figure 1: **Claims, evidence, and actions require relational checks.** Arrows denote (1) evidence–claim support, (2) evidence–decision compatibility, and (3) claim–execution correspondence, not execution order. Roles may overlap: a report is an action and can supply evidence. The schematic delivery gap illustrates the need for interaction context; missing records imply violations only under an applicable rule and adequate coverage.

extraction. The formalizer combines the resulting source-linked facts with task rules and adapter context to produce symbolic checks, whose results are interpreted for diagnostic feedback. This interface makes the selected evidence and encoded requirements inspectable together. Adapters determine evidence access, checks, and interventions; bounded retrieval and fallible encoding can still leave obligations unchecked or misrepresented.

We make three contributions:

1. **A task-obligation formulation of multi-agent consistency.** We define checks relating evidence, claims, and actions, and state the information requirement for preserving obligation-relevant distinctions (Section 2). This captures failures that factual correctness and answer agreement alone do not resolve.
2. **An evidence-linked interface for symbolic checking.** ECoCheck connects interaction graphs and source-linked facts to obligation encoding and diagnostic feedback, with explicit adapter policies for task rules, evidence access, and interventions (Section 3).
3. **Detection and outcome evidence across three environments.** Across four trace perturbations, ECoCheck records 38/120 detections versus 18/120 for the transcript-order comparator Linear, with the largest difference on unsupported claims. The full treatment exceeds Base (no verification feedback) by 20.00 percentage points in tool-collaboration macro-average overall goal success and has higher utility in all nine coordination cells. Social simulation extends the evaluation to private-information feedback. These comparisons assess checking and feedback together (Sections 4–5).

2 Problem Formulation

2.1 Task Obligations and Consistency

Let d denote a task domain and c the task context. A normalized interaction trace is $\tau = (e_1, \ldots, e_T)$, where each event e_t records an identifier, owner, type, payload, and source references. We analyze recorded interactions and do not infer unrecorded external effects.

A *claim* is a recorded assertion, conclusion, stance, or explicit commitment. An *action* is a recorded tool operation, message, output, assignment, or vote; decisions are actions such as assignments and votes. *Evidence* is a record proposed to support a claim or decision. These roles are not disjoint: a report is a communication action that contains claims and may serve as evidence for its recipient. For a source record e and claim q, let $A_d(e, q, c)$ denote whether e is *admissible* in q's context under the domain rules. Admissibility may depend on ownership, visibility, and transmission. Admissibility does not establish semantic support: even a delivered report can be misread. Citation evaluation likewise distinguishes reference presence from support (Gao et al., 2023b).

A *task obligation* is a requirement represented by a Boolean predicate $o(\tau, c) \in \{0, 1\}$, with 1 denoting satisfaction. We write $\mathcal{O}_d(c)$ for the specified obligations in domain d and context c. We group these predicates by the relations they concern: *evidence–claim support*, such as a required specialist report; *evidence–decision compatibility*, such as a choice within the permitted domain; and *claim–execution correspondence*, such as a completion claim with a required result record. These families classify requirements rather than checker modules. A statement imposes a commitment only when specified by a task rule; disagreement and decision revision are not inherently violations.

For an adequately observed execution, we define *task-relative consistency* as satisfaction of all specified obligations:

$$\mathrm{Cons}_{\mathcal{O}_d}(\tau, c) \equiv \bigwedge_{o \in \mathcal{O}_d(c)} o(\tau, c). \tag{1}$$

This definition separates consistency from optimality and task success. An adapter specifies the evidence access, checks, and feedback for a domain; checking selected obligations does not establish the full conjunction. To assess an obligation, the checker needs enough information to determine whether it is satisfied. We formalize this requirement below.

2.2 Evidence representation for consistency checking

Consider two traces with the same policy observation and final refund claim, but only one delivers the required policy report before the claim. A summary retaining only the observation and claim makes these traces indistinguishable, although only one trace satisfies the report obligation. The checker needs the delivery record to distinguish them; factual content alone is insufficient.

For fixed d, c, let $\mathcal{T}$ be a set of adequately observed traces and $R(\tau)$ the information passed to the checker. For an obligation o, call R *o-sufficient* when

$$R(\tau) = R(\tau') \implies o(\tau, c) = o(\tau', c), \qquad \tau, \tau' \in \mathcal{T}. \tag{2}$$

Here o evaluates the original trace under the task rules. The condition requires that traces with the same representation have the same obligation truth value; it does not require preserving every detail of the trace.

Equation 2 is a *design criterion* for the information supplied to a checker, not a property assumed of ECoCheck. Its elementary characterization is given in Appendix A.1. For the report example, it motivates retaining delivery and ordering records; for completion claims, it motivates retaining tool–result links. Section 3 uses these relations to guide evidence selection and encoding. Bounded selection can still omit relevant records, so missing evidence implies a violation only under an applicable rule and sufficient coverage.

3 ECoCheck: evidence-linked consistency checking

To retain the context discussed in Section 2.2, ECoCheck connects graph retrieval, source-linked facts, symbolic queries, and diagnostic feedback (Figure 2):

$$\tau \xrightarrow{B_d} G_d \xrightarrow{Q^{\mathrm{val}}_{\theta,d}} \mathcal{F} \xrightarrow{S_\theta(\cdot, c, \mathcal{C}_d)} \Phi \xrightarrow{\mathcal{Z}} z \xrightarrow{I_d} (v, r). \tag{3}$$

θ denotes the neural components' parameters. B_d builds the graph; $Q^{\mathrm{val}}_{\theta,d}$ combines graph querying and neural extraction with adapter-specific structural validation; S_θ produces SMT from facts, task context, and auxiliary adapter inputs $\mathcal{C}_d$; $\mathcal{Z}$ executes solver queries; I_d interprets responses z as verdict v and diagnostic explanation r. The diagram summarizes the FactIR path; $\mathcal{C}_d$ can also contain trace-derived inputs, including EventIR. Thus $R(\tau)$ includes all information received by the formalizer. Adapters determine validation, repair, and fallback policies.

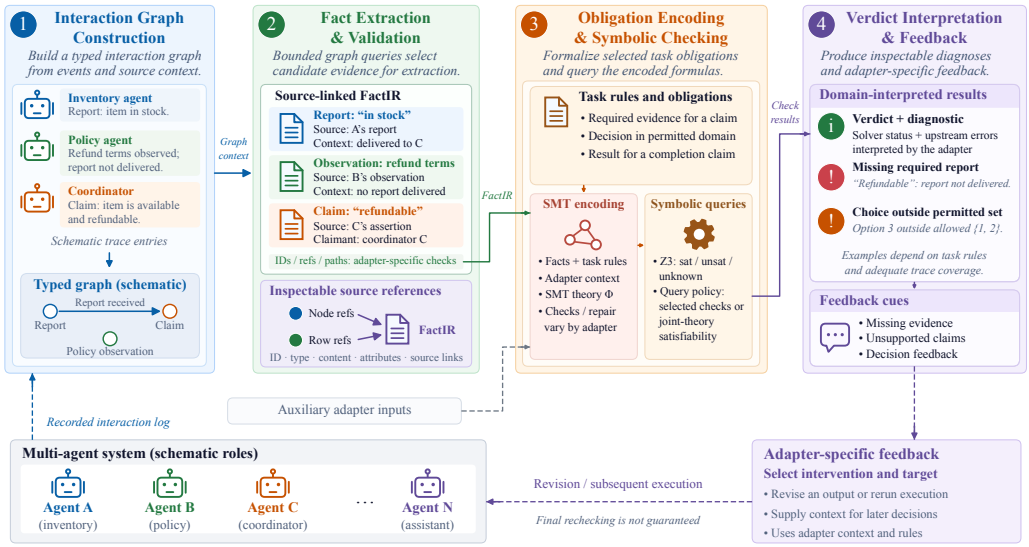


Figure 2: **ECoCheck retains evidence through a four-stage checking interface.** Selected sources pass through graph querying, fact extraction, obligation encoding, and result interpretation. Edges and valid references record provenance, not semantic support. A missing report implies a violation only under an applicable rule and adequate coverage. The dashed route supplies auxiliary adapter inputs $\mathcal{C}_d$ to encoding; Scenario Benchmark EventIR is one instance. Validation, repair, queries, and feedback are adapter-specific; final rechecking is not guaranteed. FactIR entries and diagnoses are schematic, not archived runs.

3.1 Interaction Graph Construction and Retrieval

Let $G_d = B_d(\tau) = (V, E, \rho, \eta)$, where ρ and η assign node and edge types and source aliases map nodes to trace records. Edges retain within-owner order, tool action–observation pairs, dispatch–report flow, and domain relations. Scenario Benchmark and CoLLAB use event or communication directed acyclic graphs (DAGs); Werewolf also maintains player relations. For report-support obligations, report flow and order retain delivery distinctions. For completion claims, tool–observation links expose corresponding results. Collection order alone does not establish causality (Lamport, 1978).

The dynamic retrieval path adapts Graph-CoT's querying pattern (Jin et al., 2024). Starting from the task context, the model requests source records, neighbors, reachability, or paths, then uses the returned context to guide further queries and fact extraction. The query budget covers graph queries only; it excludes finalization, SMT generation, repair, and revision. Relevant records may remain unretrieved.

3.2 Fact Extraction and Validation

Retrieved records are converted into a *fact intermediate representation* (FactIR), whose entries keep content together with its source references:

$$f = (\mathrm{id}, \mathrm{type}, \mathrm{content}, \mathrm{attributes}, R_V, R_T), \qquad R_V \subseteq V. \tag{4}$$

Here R_V is a set of graph-node references and R_T a set of source-row references. Validated entries form $\mathcal{F}$, the formalizer's fact input. The content states what was observed, reported, or claimed; the references identify where it came from, exposing distinctions such as the report-delivery gap. For decision-domain checks, CoLLAB also supplies the chosen option and permitted choices through decision records and task context.

Structural validation checks candidate facts against adapter-specific rules. Scenario Benchmark validates source references, but neither requires complete paths nor proves that a supplied path supports its claim. Invalid facts may be dropped while others continue; failure and fallback policies differ across adapters. These checks make selected sources traceable, without establishing complete retrieval, source admissibility $A_d(e, q, c)$, or semantic support (Appendix A.2).

Table 1: **Checking and feedback are adapter-specific.** Entries are implementation capabilities, not measured obligation coverage (Equation 1).

Adapter	Relations inspected	How the diagnostic is used
Scenario Benchmark	Claim support; report/prerequisite links; tool–result correspondence	Conversation regeneration includes tool/user simulators; no guaranteed final recheck.
CoLLAB	Visible broadcasts; permitted options; decision support	Protocol rerun or event repair; strict context is personal-assistant-specific.
Werewolf	Strong-claim support; stance references; role/time and vote-intent validity	Next-vote feedback combines private roles, rule-based target selection, and SMT consistency checks; correspondence between claims and executed votes is not established.

3.3 Obligation encoding and symbolic checking

The formalizer combines $\mathcal{F}$ with task premises and rules to produce an SMT theory Φ. Adapters with explicit obligations translate a selected predicate o from Section 2 into a formula $\hat{o}$ and query that encoding. Z3 answers these queries (de Moura & Bjørner, 2008); it does not verify the translation itself (Pan et al., 2023). Werewolf instead checks whether the entire theory is jointly satisfiable.

Scenario Benchmark also supplies EventIR (an event intermediate representation) and mode context, so FactIR is not the exclusive evidence input. Its Boolean programs encode selected support, contradiction, tool/result, and prerequisite requirements. Program validation and bounded repair address structural errors; semantic fidelity remains a separate requirement (Chen et al., 2026). A correct solver result cannot compensate for omitted evidence or a misencoded requirement.

3.4 Verdict interpretation and feedback

For a satisfiable theory Φ and an encoded obligation $\hat{o}$,

$$\Phi \models \hat{o} \quad \Longleftrightarrow \quad \Phi \wedge \neg \hat{o} \text{ is unsatisfiable.} \tag{5}$$

The right-hand query asks whether the obligation can fail under the encoded facts and rules. If failure is impossible, the obligation follows from the theory. A satisfiable query permits a failure, but does not establish one in the original trace. Section 2.2 concerns what information is retained; this step concerns what follows from its encoding. Applying this result to the original trace requires faithful premises and obligation formulas (Appendix A.3).

The interpreter I_d converts solver responses and upstream errors into verdict v and diagnostic r. Scenario Benchmark returns PASS, UNPASS, or INCONCLUSIVE from violation, obligation, and evidence/completeness queries. Its completeness test checks compatibility, not entailment; some diagnostic-level `unknown` results do not block PASS. Thus Equation 5 is a semantic reference, not a universal runtime verdict rule (Appendix A.3). Diagnostics guide adapter-specific feedback (Table 1).

A diagnostic may identify a missing report, an unsupported choice, or an invalid vote intention. The adapter determines the intervention: CoLLAB candidate selection varies by policy, while Werewolf scores and ranks targets under preset rules separately from SMT checking. Identifying a problem does not establish that the subsequent intervention resolves it. Section 4 measures recorded detection and task outcomes separately.

4 Experimental setup

4.1 Evaluation design

Trace perturbations measure recorded detection; task outcomes assess the full checking-and-feedback treatment described in Section 3. These measurements do not directly establish the full consistency target in Equation 1. We organize the evaluation around three questions:

RQ1: How does recorded detection vary across the four trace perturbations?
RQ2: How does Scenario Benchmark goal success compare across treatments?
RQ3: How do task outcomes compare with and without feedback in CoLLAB and Werewolf?

The *full ECoCheck treatment* includes checking and adapter-specific feedback. *Base* omits verification feedback. *Linear* retains extraction and symbolic checking with transcript precedence and within-owner order. Archived treatments labeled Graph-CoT are displayed as ECoCheck (key `graph_cot`); the label identifies our evidence-querying implementation, not a reproduced Graph-CoT baseline. Scenario Benchmark compares all three treatments; trace perturbations compare Linear with ECoCheck; CoLLAB and Werewolf compare Base with ECoCheck.

4.2 Data and evaluation metrics

Our analysis uses four archived workbooks: aggregate Scenario Benchmark and CoLLAB cells, paired corruption outcome counts, and 20 Werewolf game rows. Source hashes and cell ranges identify each input, and summaries use its stored precision. These records allow us to recompute the reported statistics. They lack the complete per-run configurations and pairing records needed to reconstruct the original model runs.

Trace perturbations: recorded detection (RQ1). Four travel-trajectory transformations remove a subagent report, remove a tool observation, remove a branch dispatch, or insert an unsupported final claim. These target transmission, source availability, execution structure, and claim support. Each type has $N = 30$ recorded outcomes, partitioned into ECoCheck-only n_E, Linear-only n_L, both n_B, and neither n_N. We compute

$$\widehat{p}_E = (n_E + n_B)/N, \qquad \widehat{p}_L = (n_L + n_B)/N. \tag{6}$$

These marginal proportions measure recorded detection labels. Missing perturbation-application flags, verdicts on unperturbed traces, and mappings from inconclusive verdicts to labels prevent us from determining which cases violate an obligation or estimating sensitivity and false-positive rates. Types may reuse source scenarios, so pooled counts are descriptive. The labels distinguish detection from the case-level error attribution evaluated by Aegis (Kong et al., 2026).

Scenario Benchmark: goal success (RQ2). We use the Multi-Agent Collaboration Scenario Benchmark (Scenario Benchmark) (Shu et al., 2024). Travel, mortgage, and software contain 30 nominal scenarios each. An LLM judge checks user and system assertions. For complete Boolean outputs, partial goal success rate (GSR) averages the satisfied-assertion fraction within each conversation and then across conversations. User and system GSR require all assertions of their respective type; overall GSR requires both. We report domain scores and an equally weighted macro-average. Effective sample and generation-failure counts are unavailable. These are task-quality metrics, not labels of consistency or formal correctness.

CoLLAB: coordination utility (RQ3). Our CoLLAB implementation provides personal assistant, meeting scheduling, and smart-grid tasks at 6, 8, and 10 agents (Mahmud et al., 2025). It normalizes native utility as

$$\mathrm{Ratio} = (U - U_{\min})/(U_{\max} - U_{\min}), \tag{7}$$

for a nonzero range. The recorded scale is consistent with $100\,\mathrm{Ratio}$, but instance bounds and replication counts are missing. We therefore retain the recorded scale and compare utility within each cell.

Werewolf: social outcomes (RQ3). We use MARBLE's Werewolf task (Zhu et al., 2025). The available configuration has nine players: three wolves, three villagers, a seer, a witch, and a guard. Ten games per treatment provide process scores, outcome score, and winner. We report sample standard deviations and 95% Wilson win-rate intervals. The intervals assume independent Bernoulli games within each group; independence and role/seed balance are unverified. The full ECoCheck treatment accesses private state and intervenes on subsequent votes. The comparison therefore includes both access to private information and voting feedback.

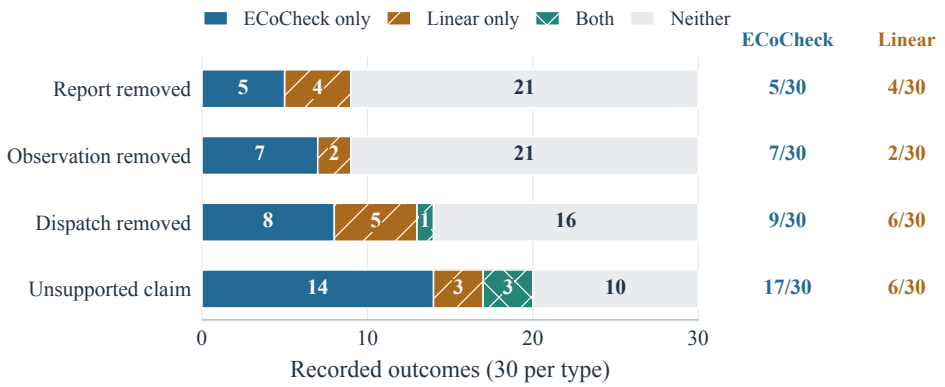


Figure 3: **Recorded detection under perturbed relations.** Stacks retain all four paired outcomes; right columns include joint detections in each method's total (30 outcomes per type). Exact recorded-label proportions appear in Table 3.

4.3 Analysis protocol

For a fixed metric, let $y_{m,j}$ be its recorded score under treatment m in cell j. For a specified comparator b, we report

$$\Delta_j^{(b)} = y_{\text{ECoCheck},j} - y_{b,j}, \qquad \overline{\Delta}^{(b)} = |\mathcal{J}|^{-1} \sum_{j \in \mathcal{J}} \Delta_j^{(b)}. \tag{8}$$

$\mathcal{J}$ contains three Scenario Benchmark domains or nine CoLLAB cells. We compute differences from unrounded scores and compare complete treatments. Without run-level observations, we cannot estimate sampling uncertainty for aggregate cells. Appendix B documents comparison scope and statistical reporting.

5 Experimental results

5.1 Trace perturbations: recorded detection

Figure 3 reports detections for each perturbation type. ECoCheck records 38 detections across 120 perturbation outcomes, compared with 18 for Linear. Of 48 discordant outcomes, 34 favor ECoCheck and 14 favor Linear. ECoCheck has higher marginal detection in every category, although the methods also detect different cases.

The largest difference concerns **unsupported final claims**: 17/30 versus 6/30, or +36.67 percentage points. The detection contrast $\widehat{p}_E - \widehat{p}_L$ (Equation 6) is smaller for removed reports, observations, and dispatches (Table 3).

Neither method records a detection in 68/120 outcomes. These include cases for which the archive does not establish whether an obligation was violated (Section 4.2); they cannot all be classified as missed violations.

5.2 Scenario Benchmark: goal success

With feedback, ECoCheck has the highest partial, user, and overall GSR in each domain (Figure 4). **Macro-average overall GSR** is 56.67%, compared with 36.67% for Base and 45.55% for Linear. Differences computed before rounding are $\overline{\Delta}^{(\text{Base})} = 20.00$ and $\overline{\Delta}^{(\text{Linear})} = 11.11$ percentage points (Equation 8). Partial GSR rises from 67.58% for Base to 81.12% for ECoCheck (+13.54 percentage points), with a +6.42-point difference over Linear.

Overall GSR gains over Base vary: +3.33 points in travel, +26.67 in mortgage, and +30.00 in software. ECoCheck does not lead on every metric: mortgage system GSR is below Linear (56.67 versus 60.00), and software system GSR is tied.

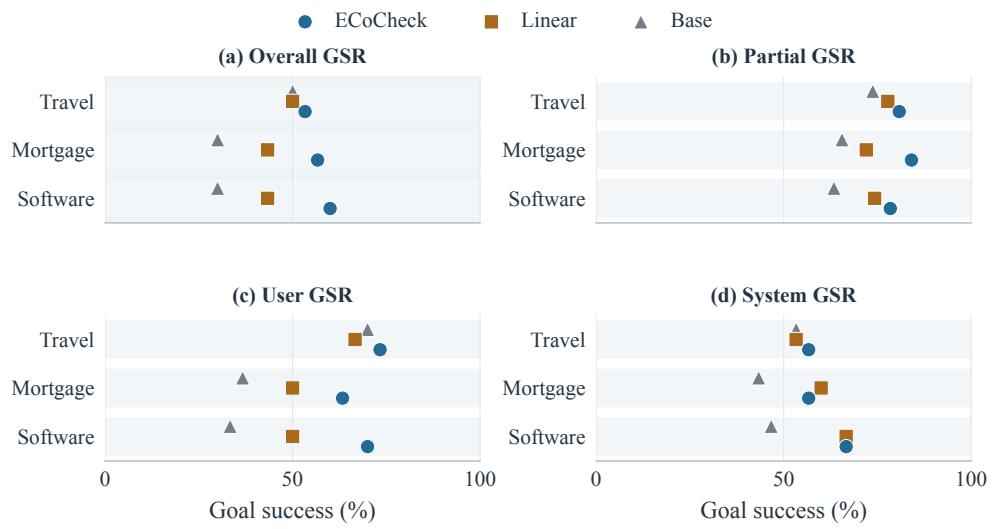


Figure 4: **Scenario Benchmark goal success under the full ECoCheck treatment.** All axes span 0–100%; offsets separate methods, including ties. The panels distinguish overall success from partial, user, and system GSR. Aggregate sampling uncertainty is unavailable. Table 4 retains every value.

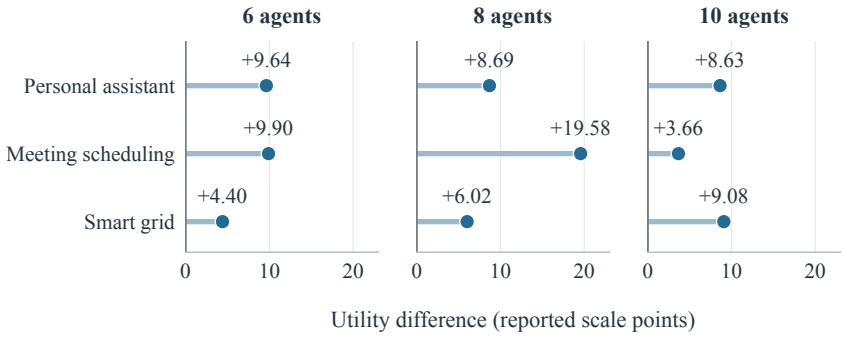


Figure 5: **CoLLAB utility gains remain positive across nine recorded cells.** Points show ECoCheck minus Base utility on shared zero-based axes; uncertainty is unavailable. Each panel compares three tasks at a fixed team size. Absolute scores: Table 5.

Partial and overall GSR describe different aspects of these outcomes. Partial GSR measures the fraction of satisfied assertions, whereas overall GSR requires their conjunction. Reporting both distinguishes progress on individual requirements from completion of the full task.

These contrasts compare *complete treatments*, including checking, additional generation, and feedback to tool/user simulators, without a guaranteed final recheck. They do not isolate the contribution of checking, and score validity also depends on the LLM judge (Zheng et al., 2023).

5.3 CoLLAB: COORDINATION UTILITY

In CoLLAB, agents make joint decisions at different team sizes. Utility measures decision quality, not obligation satisfaction. ECoCheck exceeds Base in **all nine task–scale cells** (Figure 5): $\Delta_j^{(\mathrm{Base})} > 0$ for every $j \in \mathcal{J}$, with $\overline{\Delta}^{(\mathrm{Base})} = 8.84$ points and a range of +3.66 to +19.58. Meeting scheduling at eight agents has the largest gain; across-task mean gains are not monotonic in team size.

5.4 WEREWOLF: SOCIAL OUTCOMES

In Werewolf, the checker has access to *private roles*. Villagers win 1/10 Base games and 7/10 ECoCheck games (Table 6). Mean villager score increases by 6.24, mean werewolf score decreases by 1.60, and mean outcome score increases by 4.10. Appendix Figure 6 shows every game and the wide win-rate intervals. Role/seed balance is unavailable.

These outcomes favor the full treatment combining private roles, preset target-scoring rules, and SMT checks. They do not identify an effect of SMT alone or establish claim–execution correspondence. Admissibility $A_d(e, q, c)$ makes the information boundary explicit: private state available to the checker need not be admissible evidence for a player's public claim.

6 RELATED WORK

Evidence support and answer agreement. FEVER separates supported, refuted, and insufficient-evidence claims (Thorne et al., 2018); FActScore evaluates atomic factual support (Min et al., 2023), and ALCE evaluates citation quality (Gao et al., 2023b). RARR retrieves evidence and revises unsupported text (Gao et al., 2023a). ECoCheck treats source admissibility and decision requirements as task obligations. Self-consistency aggregates sampled answers (Wang et al., 2023), while multiagent debate iterates proposals and critiques toward an answer (Du et al., 2024). SeqCV checks generated segments across models under sequential execution (Yao et al., 2025b). ECoCheck instead checks evidence-linked task obligations, which can permit disagreement.

Provenance and failure attribution. MAST categorizes failures (Cemri et al., 2025); Who&When supplies agent- and step-level attribution labels (Zhang et al., 2025c), while AgenTracer learns to locate responsible agents and steps (Zhang et al., 2026). CHIEF and FALAT analyze dependency structure (Wang et al., 2026b; Rafi et al., 2026); AgentTrails reconstructs action–artifact provenance (Wu et al., 2026). GUARDIAN detects anomalies through temporal-graph reconstruction (Zhou et al., 2025); Aegis injects errors to train attribution models (Kong et al., 2026). ECoCheck targets obligation satisfaction rather than causal responsibility. Its representation criterion concerns preserving distinctions needed to assess those obligations (Section 2.2). A successful task can still contain a claim without required support. Conversely, satisfying a local obligation need not ensure task success. Obligation checking thus complements outcome-based failure analysis for both communication requirements and checks on recorded actions.

Symbolic oversight and feedback. SatLM and Logic-LM connect language models to symbolic solvers (Ye et al., 2023; Pan et al., 2023); VeriCoT combines logical checks with premise assessment (Feng et al., 2026). FormalJudge compiles intent into specifications and extracts behavioral facts for checking and feedback (Zhou et al., 2026); AgentSpec provides runtime enforcement (Wang et al., 2026a). ECoCheck emphasizes source-linked interaction structure before symbolic checking. Appendix C extends these comparisons.

7 DISCUSSION AND LIMITATIONS

ECoCheck relies on task-specific obligations and evidence-access rules. Applying it to a new domain requires adapting these rules and the corresponding trace representation. Ambiguous task instructions can also admit different formal interpretations. Broader adapter support and clearer interfaces for specifying obligations are useful directions for extending the framework.

8 CONCLUSION

Checking factual correctness or answer agreement alone can miss violations of task rules. ECoCheck links selected evidence to symbolic checks of these rules and to the resulting feedback, so that the sources and encoded requirements can be examined together. The archived comparisons show higher recorded detection, especially for unsupported claims, and favorable task outcomes under the full treatment. These results support further study of which evidence and feedback help agents satisfy their task obligations.

AI USE STATEMENT

We used generative AI tools to assist with language editing of the manuscript. The authors led the research and take full responsibility for the paper's scientific claims, analyses, and conclusions, including any AI-assisted content.

ETHICS STATEMENT

This study analyzes archived benchmark results in accordance with applicable ethical requirements. Use of the framework should account for privacy, potential misuse, and errors in automated judgments. The reported findings do not establish deployment safety. Code and trace sharing should preserve confidentiality and comply with the licenses of the underlying resources.

REPRODUCIBILITY STATEMENT

The paper describes the checking framework, evaluation settings, and metrics, with further reporting details in Appendix B. The accompanying analysis supplement supplies the data and scripts for reproducing the reported statistics.

REFERENCES

Andreas Bauer, Martin Leucker, and Christian Schallhart. Runtime verification for LTL and TLTL. *ACM Transactions on Software Engineering and Methodology*, 20(4):1–64, 2011. doi: 10.1145/2000799.2000800. URL https://dl.acm.org/doi/10.1145/2000799.2000800.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michał Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of Thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17682–17690, 2024. doi: 10.1609/aaai.v38i16.29720. URL https://ojs.aaai.org/index.php/AAAI/article/view/29720.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei A Zaharia, Joseph Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38, pp. 135676–135729, 2025. doi: 10.52202/085713-4082. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html.

Guoxin Chen, Jing Wu, Xinjie Chen, Wayne Xin Zhao, Ruihua Song, Chengxi Li, Kai Fan, Dayiheng Liu, and Minpeng Liao. ReForm: Reflective autoformalization with prospective bounded sequence optimization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/419bd6a89ab56e9c09ededb560b06192-Abstract-Conference.html.

Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *Lecture Notes in Computer Science*, pp. 337–340. Springer, 2008. doi: 10.1007/978-3-540-78800-3_24. URL https://doi.org/10.1007/978-3-540-78800-3_24.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Yu Feng, Nathaniel Weir, Kaj Bostrom, Sam Bayless, Darion Cassel, Sapana Chaudhary, Benjamin Kiesl-Reiter, and Huzefa Rangwala. VeriCoT: Neuro-symbolic chain-of-thought validation via logical consistency checks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d7b3cef7c31b94a4a533db83d01a8882-Abstract-Conference.html.

Ferdinando Fioretto, Enrico Pontelli, and William Yeoh. Distributed constraint optimization problems and applications: A survey. *Journal of Artificial Intelligence Research*, 61:623–698, 2018. doi: 10.1613/jair.5565. URL https://doi.org/10.1613/jair.5565.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508, 2023a. doi: 10.18653/v1/2023.acl-long.910. URL https://aclanthology.org/2023.acl-long.910/.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023b. doi: 10.18653/v1/2023.emnlp-main.398. URL https://aclanthology.org/2023.emnlp-main.398/.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fef126561bbf9d4467dbb8d27334b8fe-Abstract-Conference.html.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for LLM judges with adaptive stability detection. In *Advances in Neural Information Processing Systems*, volume 38, pp. 51926–51962, 2025. doi: 10.52202/085713-1548. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/42475c537936b2394b5015e871765056-Abstract-Conference.html.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8b4add8b0aa8749d80a34ca5d941c355-Abstract-Conference.html.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. Graph Chain-of-Thought: Augmenting large language models by reasoning on graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 163–184, 2024. doi: 10.18653/v1/2024.findings-acl.11. URL https://aclanthology.org/2024.findings-acl.11/.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B Murthy. Position: LLMs can't plan, but can help planning in LLM-Modulo frameworks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22895–22907. PMLR, 2024. URL https://proceedings.mlr.press/v235/kambhampati24a.html.

Fanqi Kong, Ruijie Zhang, Huaxiao Yin, Guibin Zhang, Xiaofei Zhang, Ziang Chen, Zhaowei Zhang, Xiaoyuan Zhang, Song-Chun Zhu, and Xue Feng. Aegis: Automated error generation and attribution for multi-agent systems. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/56e8943ecd44b67bbea2702b9117b858-Abstract-Conference.html.

Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978. doi: 10.1145/359545.359563. URL https://dl.acm.org/doi/10.1145/359545.359563.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Saaduddin Mahmud, Eugene Bagdasarian, and Shlomo Zilberstein. CoLLAB: A framework for designing scalable benchmarks for agentic LLMs. In *NeurIPS 2025 Workshop on Scaling Environments for Agents*, 2025. URL https://openreview.net/forum?id=372FjQy1cF.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824, 2023. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248/.

Md Nakhla Rafi, Md Ahasanuzzaman, Dong Jae Kim, Zhijie Wang, and Tse-Hsun Chen. FALAT: Tracing failures in LLM agent trajectories via dependency-guided search. *arXiv preprint arXiv:2606.00765*, 2026. URL https://arxiv.org/abs/2606.00765.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Raphael Shu, Nilaksh Das, Michelle Yuan, Monica Sunkara, and Yi Zhang. Towards effective GenAI multi-agent collaboration: Design and evaluation for enterprise applications. *arXiv preprint arXiv:2412.05449*, 2024. URL https://arxiv.org/abs/2412.05449.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074/.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. AutoML-Agent: A multi-agent LLM framework for full-pipeline AutoML. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60099–60146. PMLR, 2025. URL https://proceedings.mlr.press/v267/trirat25a.html.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965, 2023. doi: 10.52202/075280-3275. URL https://papers.nips.cc/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Vaisakh Naduvodi Viswambharan, Keerthan Kopparam Radhakrishna, Deepak Narayan Gadde, and Aman Kumar. Knowledge graphs, the missing link in agentic AI-based formal verification. *arXiv preprint arXiv:2605.06434*, 2026. URL https://arxiv.org/abs/2605.06434.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the IEEE/ACM 48th International Conference on Software Engineering*, pp. 2938–2950, 2026a. doi: 10.1145/3744916.3764546. URL https://dl.acm.org/doi/10.1145/3744916.3764546.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Yawen Wang, Wenjie Wu, Junjie Wang, and Qing Wang. From flat logs to causal graphs: Hierarchical failure attribution for LLM-based multi-agent systems. *arXiv preprint arXiv:2602.23701*, 2026b. URL https://arxiv.org/abs/2602.23701.

Eden Wu, Sonia Castelo, Yurong Liu, Cláudio T. Silva, and Juliana Freire. AgentTrails: Towards trust and reuse for agentic tasks. *arXiv preprint arXiv:2607.18816*, 2026. URL https://arxiv.org/abs/2607.18816. Short paper accepted at DASHSys 2026; cited arXiv version.

Frank Fangzheng Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Melroy Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In *Advances in Neural Information Processing Systems*, volume 38, pp. 10291–10321, 2025. doi: 10.52202/085713-0315. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0d744742f6fac4d1134c019b7cef3c8a-Abstract-Datasets_and_Benchmarks_Track.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Yu Yao, Yiliao Song, Yian Xie, Mengdan Fan, Mingyu Guo, and Tongliang Liu. Can dependencies induced by LLM-agent workflows be trusted? In *Advances in Neural Information Processing Systems*, volume 38, pp. 19776–19814, 2025b. doi: 10.52202/085713-0587. URL https://papers.nips.cc/paper_files/paper/2025/hash/19206a6ed5ed0aaeed440448dfc5cf7e-Abstract-Conference.html.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. SatLM: Satisfiability-aided language models using declarative prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45548–45580, 2023. doi: 10.52202/075280-1974. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8e9c7d4a48bdac81a58f983a64aaf42b-Abstract-Conference.html.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*,

2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/bbc461518c59a2a8d64e70e2c38c4a0e-Abstract-Conference.html.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007726.

Jusheng Zhang, Yijia Fan, Wenjun Lin, Ruiqi Chen, Haoyi Jiang, Wenhao Chai, Jian Wang, and Keze Wang. GAM-Agent: Game-theoretic and uncertainty-aware collaboration for complex visual reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 159314–159374, 2025b. doi: 10.52202/085713-4808. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d38512ba4121046784528ad9984af74a-Abstract-Conference.html.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76583–76599. PMLR, 2025c. URL https://proceedings.mlr.press/v267/zhang25cq.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Jialong Zhou, Lichao Wang, and Xiao Yang. GUARDIAN: Safeguarding LLM multi-agent collaborations with temporal graph modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 8881–8909, 2025. doi: 10.52202/085713-0272. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0bc795afae289ed465a65a3b4b1f4eb7-Abstract-Conference.html.

Jiayi Zhou, Yang Sheng, Hantao Lou, Yaodong Yang, and Jie Fu. FormalJudge: A neuro-symbolic paradigm for agentic oversight. *arXiv preprint arXiv:2602.11136*, 2026. URL https://arxiv.org/abs/2602.11136.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

APPENDIX

A EVIDENCE VALIDATION AND CHECKING SEMANTICS

This appendix clarifies the validation and logical conditions behind Sections 2–3, then documents the evaluation scope and complete reported results.

A.1 REPRESENTATION CRITERION

For fixed d, c, use $\mathcal{T}$, R, and o as defined in Section 2.2. The following elementary result explains the information-preservation criterion in Equation 2.

Proposition 1 (representation sufficiency). *A judgment function h using only $R(\tau)$ can satisfy $h(R(\tau)) = o(\tau, c)$ for every $\tau \in \mathcal{T}$ if and only if R is o-sufficient. Proof.* Identical inputs force h to give the same answer, so the condition is necessary. Conversely, assign each representation the common obligation truth value of its traces; the condition makes this unambiguous.

This is an existence statement about a judgment function, not a guarantee that retrieval, extraction, or encoding realizes that function. It motivates retaining obligation-relevant distinctions without asserting sufficiency for ECoCheck's selected inputs.

A.2 SOURCE-REFERENCE VALIDATION

Reference validation checks whether selected facts point to existing records and satisfy adapter-specific structural requirements. It does not establish that a source is admissible for the claimant or that its content supports the claim. Bounded retrieval may omit relevant evidence, and Scenario Benchmark's auxiliary EventIR route bypasses FactIR validation. Missing evidence implies a violation only under an applicable rule and sufficient observation coverage; the archive does not establish that coverage for each run.

A.3 SOLVER RESULTS AND VERDICT INTERPRETATION

Scenario Benchmark returns PASS, UNPASS, or INCONCLUSIVE according to its encoded checks and interpretation policy. These are checker outcomes, not statements of task success. A satisfiable completeness query establishes compatibility rather than entailment, and some unresolved diagnostic queries do not block PASS. Thus PASS does not certify that every required query was resolved. Solver-to-verdict mappings differ across adapters.

From encoded entailment to trace obligations. Let $\mathcal{M}_\Phi = \{\nu : \nu \models \Phi\}$ be the models of the encoded theory. Let $J \subseteq \mathcal{O}_d(c)$ be selected intended obligations, $\widehat{\mathcal{O}}$ their encoded formulas, and $\iota : J \to \widehat{\mathcal{O}}$ the encoding map, with $\iota(o) = \hat{o}$ in Section 3.3. Suppose there is an interpretation $\nu_\tau \in \mathcal{M}_\Phi$ representing the recorded trace such that, for every $o \in J$,

$$\nu_\tau \models \iota(o) \quad \Longleftrightarrow \quad o(\tau, c) = 1. \tag{9}$$

If $\Phi \models \iota(o)$ for every $o \in J$, then $\bigwedge_{o \in J} o(\tau, c) = 1$. Indeed, entailment applies to every model of Φ, including ν_τ; Equation 9 then transfers each truth value. This elementary conditional result separates three requirements: the encoding must admit a faithful trace model, obligation formulas must preserve their intended meanings, and the required entailments must hold. The same argument establishes full consistency when $J = \mathcal{O}_d(c)$ and observation is adequate. Neither structural reference validation nor solver success establishes these assumptions. These conditions would justify the inference, but they have not been verified for ECoCheck or its reported runs.

Satisfiability versus entailment. Let $\Phi = \neg h \wedge \neg u$, where h denotes a violation flag and u insufficient evidence, and leave completeness p unconstrained. Both $\Phi \wedge p$ and $\Phi \wedge \neg p$ are satisfiable, although $\Phi \wedge h$ and $\Phi \wedge u$ are unsatisfiable. A positive-completeness test can therefore accept without establishing $\Phi \models p$. This is an illustrative logical distinction, not an observed experimental failure.

B EVALUATION SCOPE AND REPORTING

Benchmark repositories are available for Scenario Benchmark, CoLLAB, and MARBLE (Werewolf). The four archived workbooks are the numerical source of the reported results. The implementation shows which checks are available, but the archive does not link individual runs to the code paths used. We therefore cannot identify the executed checks from aggregate workbook cells (Table 2).

Table 2: **Evidence units and evaluation scope.** The tables report results at the level supported by the available records; the implementation alone cannot establish which checks were executed in each run.

Endpoint	Available evidence unit	Unresolved run-level information
Recorded detection	Paired counts by perturbation	Application/eligibility flags and clean verdicts; source-scenario dependence.
Scenario Benchmark goal success	Domain–treatment aggregate	Per-conversation assertions linked to checks, repairs, and final rechecks.
CoLLAB utility	Task–scale–treatment cell	Executed verification path, instance bounds, and replication counts.
Werewolf outcomes	Individual game row	Role/seed balance and daily claims, feedback, and actual votes.

B.1 EVIDENCE ACCESS AND COMPARISON SCOPE

Scenario Benchmark measures the full feedback treatment, including conversation and simulator regeneration, without a guaranteed final recheck. CoLLAB's aggregate cells do not identify the selection policy or verification path; strict checking context is specific to personal-assistant tasks. Werewolf combines private-role access, SMT consistency checks, and a separate module that scores voting targets under preset rules. It is neither a public-information-only comparison nor a solver-derived voting policy. Game identifiers do not establish matched roles or seeds, and original scoring-model assessments are unavailable. The recorded outcomes must be interpreted with these differences in access and intervention in mind.

B.2 STATISTICAL REPORTING

Scenario Benchmark and CoLLAB report aggregate cells; their equally weighted macro-averages and within-cell differences are descriptive. The contrasts in Equation 8 use unrounded scores, giving an ECoCheck–Linear overall GSR difference of 11.11 percentage points. CoLLAB retains the recorded utility scale. Interpreting Scenario Benchmark overall GSR as satisfaction of all assertions requires complete, well-formed Boolean judgments. The aggregates cannot establish this condition for every conversation. Perturbation types may share source scenarios, so pooled detection counts do not support an independence-based test.

Werewolf summaries use ten games per group, sample standard deviations, and 95% Wilson intervals for win rates. Within-group independence is unverified, and the archive does not support paired analysis. We report these summaries without treating the games as randomized or matched comparisons.

The accompanying analysis supplement contains normalized tables, workbook hashes, source-cell locations, and scripts to reproduce the reported statistics. It does not reconstruct the original model runs, whose configurations and per-run links between traces, checks, feedback, and outcomes are not fully recorded.

C EXTENDED RELATED WORK

These conceptual comparisons supplement Section 6; the methods discussed here are not reproduced baselines.

Action and outcome evaluation. ReAct interleaves reasoning, actions, and observations (Yao et al., 2023); Toolformer learns API use (Schick et al., 2023). Both connect generation to external evidence,

whereas ECoCheck inspects selected interaction records. AgentBench evaluates interactive task performance (Liu et al., 2024); τ-bench evaluates tool agents through final state and repeated-trial reliability (Yao et al., 2025a). That across-trial consistency differs from within-trace obligation satisfaction. TheAgentCompany evaluates workplace tasks (Xu et al., 2025); AutoML-Agent combines specialized agents with multi-stage execution verification (Trirat et al., 2025). These settings further motivate separating outcome quality from evidence support. The three benchmarks report task outcomes (Shu et al., 2024; Mahmud et al., 2025; Zhu et al., 2025); they do not directly label task-relative consistency.

Graph structure and partial observation. Graph of Thoughts composes reasoning steps (Besta et al., 2024); GPTSwarm optimizes agent computation graphs (Zhuge et al., 2024); Graph Chain-of-Thought queries text-attributed graphs (Jin et al., 2024). ECoCheck adapts graph querying to source selection in interaction records, not topology learning. AgentPrune removes redundant or harmful messages from communication graphs (Zhang et al., 2025a); ECoCheck retrieves relations for checking without optimizing the communication topology. Graph-grounded hardware verification is also a precedent for graph/formal-feedback interfaces (Viswambharan et al., 2026). The happened-before relation of Lamport (1978) distinguishes communication and process order from total log order; runtime verification treats inconclusive finite observations explicitly (Bauer et al., 2011). These distinctions matter when interpreting incomplete traces; our graph does not establish causality, and our interpreter is not a temporal monitor. Distributed constraint optimization likewise distinguishes feasibility from utility (Fioretto et al., 2018).

Feedback and faithfulness. LLM-Modulo uses external critics (Kambhampati et al., 2024); Self-Refine, Reflexion, and CRITIC respectively use iterative self-feedback, stored reflections, and tool-assisted feedback (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024). Huang et al. (2024) find that reasoning self-correction can fail without external feedback. ECoCheck focuses on retaining evidence context for adapter-specific checks. Z3 solves encoded queries (de Moura & Bjørner, 2008), not their translation from language. Process supervision uses step-level labels (Lightman et al., 2024); our archive contains no such labels. CoT explanations can be unfaithful (Turpin et al., 2023), and LLM judges can be biased (Zheng et al., 2023). Inspectable records and task scores therefore do not verify a model's internal reasoning. GAM-Agent communicates structured claims, evidence, and uncertainty for visual reasoning (Zhang et al., 2025b); Multi-Agent Debate for LLM Judges uses adaptive stopping based on judgment stability (Hu et al., 2025). These address collaborative reasoning and evaluation rather than the trace obligations checked here; neither is a reproduced baseline in our experiments.

D Detailed experimental results

The following tables preserve every reported value from the frozen analysis used for the figures. Uncertainty estimates are omitted for aggregate cells without run-level observations.

Table 3: **Recorded detection under perturbed relations.** E is ECoCheck and L is Linear. Each type has 30 outcomes; marginal detection includes Both. The 120-outcome pool is descriptive because scenarios may recur. Missing eligibility and clean verdicts prevent calibrated recall or false-positive estimates.

	Paired outcome counts				Detection (%)	
Perturbation	**E only**	**L only**	**Both**	**Neither**	**E**	**L**
Report removal	5	4	0	21	16.67	13.33
Observation removal	7	2	0	21	23.33	6.67
Dispatch removal	8	5	1	16	30.00	20.00
Unsupported claim	14	3	3	10	56.67	20.00
Descriptive pool	34	14	4	68	31.67	15.00

Table 4: **Scenario Benchmark goal success by treatment.** For complete Boolean judgments, overall GSR requires all user and system assertions to hold. All scores and the equally weighted domain macro-average are retained. Bold indicates the largest displayed score within a comparison, including ties; it does not establish statistical significance or task-relative consistency.

		Overall goal success	Requirement-level GSR (%)		
Domain	**Treatment**	**Overall (%)**	**Partial**	**User**	**System**
Travel	Base	50.00	73.77	70.00	53.33
	Linear	50.00	77.78	66.67	53.33
	ECoCheck	**53.33**	**80.83**	**73.33**	**56.67**
Mortgage	Base	30.00	65.56	36.67	43.33
	Linear	43.33	72.04	50.00	**60.00**
	ECoCheck	**56.67**	**84.08**	**63.33**	56.67
Software	Base	30.00	63.41	33.33	46.67
	Linear	43.33	74.26	50.00	**66.67**
	ECoCheck	**60.00**	**78.44**	**70.00**	**66.67**
Macro Avg.	Base	36.67	67.58	46.67	47.78
	Linear	45.55	74.69	55.56	**60.00**
	ECoCheck	**56.67**	**81.12**	**68.89**	**60.00**

Table 5: **CoLLAB utility under the full ECoCheck treatment.** Scores retain the recorded utility scale; $\Delta = \mathrm{ECoCheck} - \mathrm{Base}$ is computed before rounding. Across-task means weight the three tasks equally. These cells do not measure obligation satisfaction or establish a scaling law.

Task	**Agents**	**Base**	**ECoCheck**	**Δ utility**
Personal assistant	6	62.50	72.14	9.64
	8	65.63	74.32	8.69
	10	77.08	85.71	8.63
Meeting scheduling	6	58.33	68.23	9.90
	8	45.16	64.74	19.58
	10	88.71	92.37	3.66
Smart grid	6	59.27	63.67	4.40
	8	59.22	65.24	6.02
	10	58.71	67.79	9.08
Across-task mean	6	60.03	68.01	7.98
	8	56.67	68.10	11.43
	10	74.83	81.96	7.12

Table 6: **Werewolf outcomes under the full ECoCheck treatment.** Scores are mean $\pm$ sample SD; win-rate brackets are 95% Wilson intervals assuming independent games, which the archive cannot verify. Private roles, rule-based target selection, and SMT feedback form a compound treatment.

Endpoint	**Base**	**ECoCheck**
Games	10	10
Villager process score	3.71 ± 1.98	9.95 ± 3.74
Werewolf process score	6.50 ± 1.27	4.90 ± 2.08
Outcome score	-1.80 ± 1.62	2.30 ± 2.98
Villager win rate	10% [1.8, 40.4]	70% [39.7, 89.2]

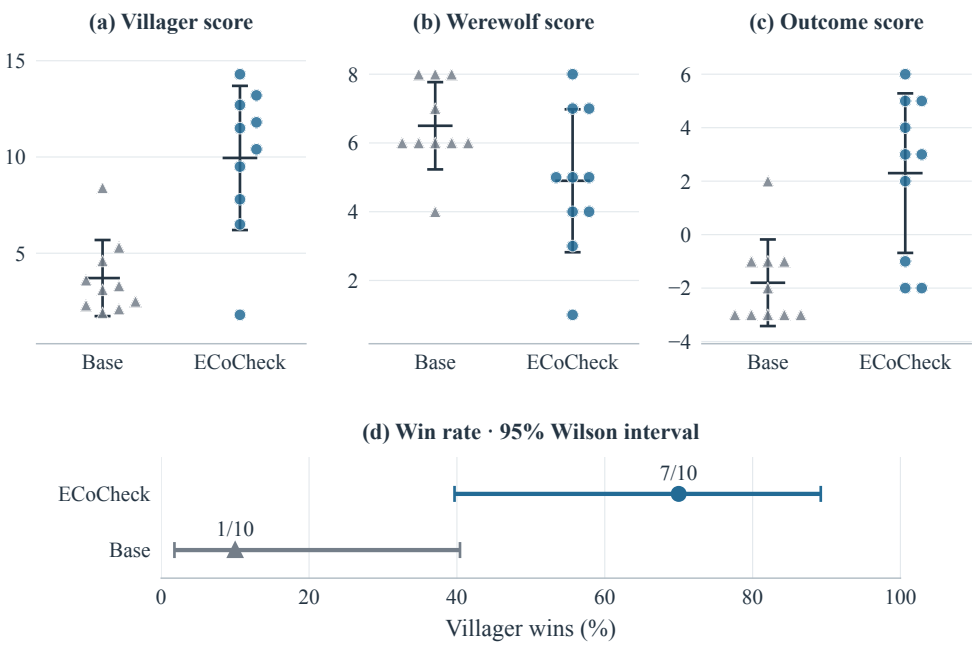


Figure 6: **Werewolf outcomes under the full ECoCheck treatment.** Panels (a)–(c) show all game-level scores (10 per group); horizontal offsets only separate marks, and black summaries show mean ± sample SD. Panel (d) shows wins and 95% Wilson intervals under unverified within-group independence. Private-role access and rule-based target selection accompany SMT checks; the outcomes do not directly measure claim–execution correspondence for votes.