

Procedural Blind Spots in Agent Evaluation: A Survey, Coverage Analysis, and New Missions for NLP

Hongliu Cao¹, Ilias Driouich¹, Eoin Thomas¹

¹Amadeus SAS

Correspondence: caohongliu@gmail.com

Abstract

Large Language Model (LLM) agents increasingly operate in multi-step, multi-turn settings that combine user interaction, tool use, state changes, and communication. Most agent evaluation still emphasizes task completion, while procedure-aware checks remain fragmented. We conduct a systematic survey and introduce $\mathcal{F}^*$, a systematic organizing map that maps scattered procedural checks into a shared audit space spanning integrity relations and quality dimensions. Applying this framework to recent agent-evaluation work reveals three procedural blind spots: most work remains outcome-focused or procedurally under-specified, procedure-aware checks mostly remain local spot-checks, and communication grounding lags action compliance. To close these gaps, we propose four new Natural Language Processing (NLP) missions around communication grounding, constraint semantics, reasoning faithfulness, and procedural equity. This survey provides a comprehensive overview of procedure-aware agent evaluation, offering a common vocabulary and coverage map that clarify what current benchmarks can certify and provide clear guidelines for future research.

1 Introduction

Large Language Model (LLM)-based agents are increasingly evaluated in tasks that unfold over multiple steps and turns: they retrieve information, call tools, modify state, navigate interfaces, and report decisions or explanations (Wang et al., 2023; Zhou et al., 2024). Unlike one-shot text generation for human review, these interactions produce procedures whose effects may be difficult to undo. A booking confirmed on the wrong date or a refund issued without authorization cannot simply be corrected by regenerating a response.

Evaluation has not kept pace. Leading benchmarks still focus on terminal state: whether a code patch resolves an issue (Jimenez et al., 2024), a web

page reaches the target state (Zhou et al., 2024), or a final answer matches a gold reference (Mialon et al., 2024). They evaluate the destination, not the trajectory. An agent that bypasses authentication, fabricates policy, or misrepresents what it did can receive the same score as one that followed every required step. This is the phenomenon of *corrupt success*: procedurally flawed behavior that produces correct-looking outcomes (Cao et al., 2026a). For example, a customer-service agent may issue the correct refund but cite a nonexistent “48-hour cooling-off policy” and never actually verify the user’s booking record. The survey question is therefore not only whether a benchmark checks procedure, but which cross-artifact consistency relation its score is licensed to certify.

The difficulty is not that procedure-aware evaluation is entirely missing. Pieces already exist across tool grounding, safety stress testing, policy conformance, and dialogue evaluation (Chen et al., 2024, 2026; Andriushchenko et al., 2025; Zhang et al., 2024; Levy et al., 2024; Li et al., 2025; Mehri and Eskenazi, 2020; Qian et al., 2025). Nearby surveys cover agent evaluation, multi-turn interaction, LLM-as-judge methods, holistic evaluation, hallucination, and fairness (Mohammadi et al., 2025; Yehudai et al., 2025; Guan et al., 2026; Gu et al., 2026; Liang et al., 2023; Ji et al., 2023; Blodgett et al., 2020). Appendix C compares the closest ones. These surveys primarily organize the literature by task, capability, environment, benchmark family, or judging method. They do not ask which consistency relation among actions, observations, communications, policies, instructions, user requests, and reasoning traces an evaluation score actually certifies. What is missing is therefore not another isolated benchmark, but a measurement layer that tells us how existing procedural checks relate, what they cover, and what they miss. Such a map separates hard integrity failures from quality degradation and makes coverage gaps explicit.

This survey provides that map through $\mathcal{F}^*$, a framework whose nine integrity pairs are derived by systematically pairing the process sub-components of $\mathcal{F}^*$ and whose three quality dimensions are motivated by service science, so that coverage gaps become nameable. We screened 3,342 candidate papers through a four-tier pipeline and manually reviewed 169 procedure-relevant papers against the framework. The resulting coverage map reveals that procedural checks cluster narrowly around action compliance, leaving communication grounding, reasoning faithfulness, and cross-session consistency largely unaudited. These gaps point to a shared need: agent evaluation must move from isolated procedure checks to procedural measurement that makes explicit which consistency relations among evidence, actions, policies, reasoning, and user-facing communication a score actually certifies. The broader NLP mission is to build the methods that make such certification possible.

This paper makes three contributions: (i) it synthesizes fragmented agent-evaluation work into the $\mathcal{F}^*$ framework (Section 2), (ii) it provides a systematic coverage analysis of what current benchmarks can and cannot certify (Section 3), and (iii) it derives four NLP missions for procedure-aware evaluation: communication grounding, constraint semantics, reasoning diagnosis, and equitable procedure (Section 4), turning the coverage analysis into a concrete research agenda.

2 The $\mathcal{F}^*$ Framework: Integrity Pairs and Quality Dimensions

This section defines the vocabulary used to map benchmarks in Section 3. The central distinction is between a *procedure* and a *trajectory*. A trajectory is a sequential log of events: which tools were called, what messages were sent, and in what order. A procedure additionally captures the consistency relations among observations, actions, communicated content, applicable rules, and user requests. A trajectory can therefore be complete as a log while still underspecifying whether claims were grounded, reasoning reflected actual computation, or the agent served the user’s expressed needs. $\mathcal{F}^*$ formalizes procedure in this sense.

Following prior agent formalizations (Oliehoek and Amato, 2016; Yao et al., 2025; Cao et al., 2026a), we represent a complete agent procedure as

$$\mathcal{F}^* = \langle \Lambda, (\mathcal{E}_t, \mathcal{A}_t, \mathcal{O}_t, \phi_t)_{t=1}^T \rangle, \quad (1)$$

where Λ is the agent instruction set, fixed before the episode begins. At each step t , $\mathcal{E}_t$ denotes the environment state, $\mathcal{A}_t$ the agent’s action, $\mathcal{O}_t$ the observations received, and ϕ_t the agent’s internal reasoning trace (chain-of-thought or scratchpad, produced before acting but not always observable). $\mathcal{E}_t$ anchors outcome evaluation: most existing benchmarks check whether the final state $\mathcal{E}_T$ matches a goal specification. The integrity pairs below complement rather than replace that check, asking whether the *procedure* that produced $\mathcal{E}_T$ was itself trustworthy. Actions decompose into three types: reading (A_t^R), writing or state-changing (A_t^W), and communicating (A_t^C). Observations have three sources: system returns such as tool outputs and database results (O_t^{sys}), contextual documents such as policies (O_t^{ctx}), and user utterances (O_t^{usr}). Making Λ explicit matters because instruction-grounded failures are not reducible to environment state alone. An agent can act out of scope, leak its prompt, or misrepresent its role without changing the final task outcome.

The nine integrity pairs in Table 1 are derived by systematically pairing the eight process sub-components of $\mathcal{F}^*$ (excluding $\mathcal{E}$, which anchors outcome evaluation) and retaining every pair whose mismatch constitutes an attributable agent fault; Appendix B accounts for the excluded cross-products. The 28 pairs reduce to nine by excluding four categories: (i) observation–observation pairs (consistency among inputs is an environment property, not an agent fault; e.g., if tool outputs arrive contradictory, the inconsistency is environmental, but the agent’s reasoning over those contradictions is captured by P5: the fault boundary is the agent’s decision to accept or reject an unsupported premise, not the input itself), (ii) instruction–input pairs (instructions constrain what the agent *does*, not what it receives), (iii) generative pairs (a read action *produces* its observation; this is causal, not a violable consistency), and (iv) mediated pairs (user-to-action fulfillment is already checked through P9, P4, and P8 rather than independently). Most pairs operate at the sub-component level (e.g., O^{sys} vs. A^C); P5 and P6 aggregate observation and action types because the consistency question is source- or kind-independent. The framework has two layers. *Integrity* refers to pairwise consistency checks: whether communicated content is grounded in observations, claimed actions match executed actions, reasoning is grounded in evidence and faithfully drives action, instructions and

Table 1: The nine integrity pairs and three quality dimensions of the $\mathcal{F}^*$ framework. Pair names serve as identifiers in the coverage map (Section 3); the derivation and exclusion rationale appear in Section 2 and Appendix B.

#	Name	Relation	Failure detected
P1	Data grounding	$O^{sys} - A^C$	Fabricated facts
P2	Execution compliance	$A^C - (A^R, A^W)$	Unexecuted claimed actions
P3	Policy grounding	$O^{ctx} - A^C$	Hallucinated policy
P4	Policy compliance	$O^{ctx} - (A^R, A^W)$	Unauthorized action
P5	Reasoning grounding	$O - \phi$	Fabricated premises
P6	Reasoning compliance	$\phi - A$	Sycophancy, deceptive alignment
P7	Instruction grounding	$\Lambda - A^C$	Role misrepresentation, prompt leak
P8	Instruction compliance	$\Lambda - (A^R, A^W)$	Out-of-scope behavior
P9	User fulfillment	$O^{usr} - A^C$	Ignored user intent
Q ₁	Efficiency (steps, cost, latency)		
Q ₂	Resilience (error recovery, robustness)		
Q ₃	Interaction quality (burden, tone, clarity)		

policies are respected, and user requests are fulfilled. *Quality* refers to efficiency, resilience, and interaction quality, motivated by service science models of operational cost, error recovery, and customer effort (Parasuraman et al., 1988). Integrity gives necessary conditions for procedural trustworthiness; quality asks how well the agent performs once those conditions are considered.

Table 1 groups these pairs into execution, policy, reasoning, instruction, and user-fulfillment categories, alongside three quality dimensions. Each pair names a relationship between component types; the temporal scope of the check (same-step, cumulative, or trajectory-wide) depends on the constraint applied. Two notation choices reflect design decisions. P6 uses A (all action types) because reasoning should causally drive every output including communication; P8 uses (A^R, A^W) because the communication-specific instruction check is already covered by P7. P9 checks user fulfillment through communication ($O^{usr} - A^C$); whether the agent also *acted* on the request is already covered by P2, P4, and P8. The pairs are analytic lenses, not claims that real failures map to exactly one pair. A benchmark may touch multiple relations, and Section 3 maps partial or artifact-limited evidence conservatively. The framework is a systematic organizing map for the surveyed literature, not a claimed-complete taxonomy. Its contributions do not depend on the nine pairs being the only possible decomposition, and hold if pairs are split, merged, or added. Applicability is risk-tiered: high-stakes deployments may audit most applicable pairs, while lower-risk settings (sandboxed games,

story generation) require fewer, because communication integrity (P1/P2) and instruction leakage (P7) remain auditable even when policy constraints (P4) do not apply.

3 What Current Agent Evaluation Covers

Having defined $\mathcal{F}^*$, we next use it as a mapping instrument to ask what current agent evaluations can actually certify. We focus on which procedural artifacts they collect, which integrity and quality dimensions they score, and which claims remain unsupported.

3.1 Survey Protocol and Coverage Map

We conducted a systematic scoping review of agent-evaluation work (full protocol in Appendix D). Searching ACL Anthology, OpenReview and arXiv yielded 3,342 unique records. Three successive filters (procedural-signal keyword screening, $\approx 1,030$; strict eligibility, 360; AI-assisted rubric screening, 169) produced 169 candidates for full-text manual review, of which 54 provided enough procedural evidence to map at least one integrity pair or quality dimension. The remaining 115 papers (68%) reference or analyze procedural properties but measure them only through task outcome. Table 2 maps this set; papers with no procedural measurement are excluded rather than shown as blank rows.

Our annotation manual assigns $\checkmark$ for explicit audits, $\sim$ for partial or indirect evidence, $\diamond$ for outcome-based adversarial stress tests, and blanks when the paper gives no usable evidence; ambiguous cases receive the weaker label. We include $\diamond$ because these outcome-based tests indirectly check procedure by asking whether a constraint survives attack. We count them separately because attack success rates show whether a constraint failed, not which procedural mechanism failed. Appendix E gives the annotation criteria, exclusion principles, and evaluation-paradigm definitions. For counts, each paper is counted once per category, $\checkmark$ and $\sim$ form direct coverage, and $\diamond$ is reported separately. The coverage map yields three key findings.

3.2 Key Findings

Finding 1: Most reviewed agent-evaluation work remains outcome-focused or procedurally under-specified. The first pattern comes from the filtering procedure itself. Most of the 169 reviewed papers did not qualify for Table 2. Many excluded works, including SWE-bench (Jimenez

Table 2: Coverage of integrity pairs and quality dimensions across the 54 qualified procedure-aware works. Reviewed papers with no procedural measurement are excluded rather than shown as blank rows. ✓ = explicit audit; ◊ = adversarial resilience (outcome-based); ~ = partial/indirect; blank = no evidence. **Eval:** S = single-turn single-step, MS = multi-step, MT = multi-turn, MS+MT = both. Q_1 = Efficiency, Q_2 = Resilience, Q_3 = Interaction Quality.

Work	Domain	Eval	Exec.		Policy		Reason.		Instr.		User	Quality		
			P1	P2	P3	P4	P5	P6	P7	P8	P9	Q_1	Q_2	Q_3
TRACE (deep research) (Chen et al., 2026)	Research	MS	✓									✓	✓	
TD-EVAL (Acikgoz et al., 2025)	Dialogue	MS+MT	✓		~									
GSAR (Kamelhar, 2026)	AIOps	MS	✓											
Scalable Oversight (Kenton et al., 2024)	AI oversight	MT	~										◊	
Multi-Faceted (Shourya et al., 2026)	Dialogue	MT	~											~
M3-Bench (Xie et al., 2026)	Social	MS+MT		✓			~							~
Process Eval. (Gritta et al., 2026)	General	MS									~			
Beyond State Machines (Garigipati et al., 2026)	Telecom	MS		~		~								
GuideBench (Diao et al., 2025)	General	S			✓									
CoPriva (Chang et al., 2025)	Security	S			✓									
ST-WebAgentBench (Levy et al., 2024)	Web	MS				✓								
SOPBench (Li et al., 2025)	SOP	MS				✓								
SOP-Bench (Nandi et al., 2026)	SOP	MS				✓								
Co. Policy Adherence (Zwerdling et al., 2025)	Enterprise	MS+MT				✓								
MANTRA (Anand et al., 2026)	Enterprise	MS				✓								
CCTU (Ye et al., 2026b)	Tool use	MS				✓								
AudAgent (Zheng et al., 2026)	Security	MS				✓								
Near-Miss (Rabinovich et al., 2026)	Cust. svc.	MS+MT				✓								
Solver-Aided (Winston et al., 2026)	Cust. svc.	MS+MT				✓								
Claw-Eval (Ye et al., 2026a)	General	MS				~							✓	
TransLaTeX (Lyn and Graham, 2025)	Sci. format	MS				~								
SOP-Maze (Wang et al., 2026)	SOP	S				~								
Beyond IVR (Balaji et al., 2026)	Cust. svc.	MT				~								
FM SO.P (Huang et al., 2026)	SOP	S				~								
Policy Red-Teaming (Nakash et al., 2025)	Safety	MS+MT				◊								
ASB (Zhang et al., 2025)	General	MS				◊				◊				
FinVault (Yang et al., 2026)	Finance	MS				◊				◊				
Ariadne (Khanzadeh, 2026)	Reasoning	S					✓	✓						
TRACE (tool-aug.) (Kim et al., 2025)	Tool use	MS					✓					✓		
SeekBench (Shao et al., 2025)	Search	MS					✓						✓	
A-CC (Waqas et al., 2026)	Tool use	MS												
CoT-Control (Chen et al., 2025a)	Reasoning	S						~						
AgentIF (Qi et al., 2025)	General	S							✓	✓				
Agent-SafetyBench (Zhang et al., 2024)	Safety	MS							~	~				
ToolEmu (Ruan et al., 2024)	Safety	MS								✓				
AgentDojo (Debenedetti et al., 2024)	General	MS								◊				
AgentDyn (Li et al., 2026)	General	MS								◊				
AgentHarm (Andriushchenko et al., 2025)	Safety	MS								◊				
CIBER (Ba et al., 2026)	Code	MS								◊				
τ -bench (Yao et al., 2025)	Cust. svc.	MS+MT				~					✓			
τ^2 -bench (Barres et al., 2025)	Cust. svc.	MS+MT				~					✓			
HotelQuEST (Hadad et al., 2026)	Hotels	MS		✓							✓	✓		
Mind2Web 2 (Gou et al., 2025)	Web	MS		✓							✓			
TRAJECT-Bench (He et al., 2025a)	Tool use	MS		~							✓	✓	~	
Userbench (Qian et al., 2025)	General	MS+MT									✓			
Vitabench (He et al., 2025b)	General	MS+MT									✓			
COMPASS (Qin et al., 2026)	General	MS+MT									✓			
TRIP-Bench (Shen et al., 2026)	General	MS+MT									✓			
INSURE-Dial (Kulkarni et al., 2026)	Healthcare	MT									~			
Online-Mind2Web (Xue et al., 2025)	Web	MS										✓		
USR (Mehri and Eskenazi, 2020)	Dialogue	S												✓
TCQA ² (Chen et al., 2025b)	Gaming	MT												✓
WikiHowAgent (Pei et al., 2025)	Education	MT												✓
Interaction Smells (Zhang et al., 2026)	Code	MT												~

et al., 2024), WebArena (Zhou et al., 2024), GAIA (Mialon et al., 2024), TheAgentCompany (Xu et al., 2025), and ScienceAgentBench (Chen et al., 2025c), score terminal-state correctness (patch acceptance, element match, exact answer) rather than procedural integrity. Such violations are invisible to outcome scoring when the final state remains correct, so outcome-focused evaluation cannot distinguish trustworthy success from corrupt success.

Finding 2: Procedure-aware evaluation remains mostly spot-checking. Within Table 2, procedure-aware checks are sparse and localized.

We call this pattern spot-checking: each work typically verifies one local part of $\mathcal{F}^*$ rather than reporting across the full set of integrity pairs and quality dimensions. The non-empty cells cluster around a few relations: execution checks based on tool outputs and action traces (P1 has 8 direct works; P2 has 3) and policy-compliance checks based on explicit policies or Standard Operating Procedures (SOPs; P4 has 17 direct works plus 3 adversarial studies). In contrast, reasoning, instruction, and communication-grounding dimensions are thinner because they require reasoning traces, deployment-instruction records, user-need speci-

fications, or cross-view scoring procedures that many papers do not report. Quality dimensions are also rarely integrated with integrity reporting: Q_1 appears in 5 direct works, Q_2 in 5 direct plus 1 adversarial study, and Q_3 in 6 direct works.

No work in Table 2 checks more than two integrity pairs even counting partial and adversarial labels. Without a procedural-space map, these gaps are not merely unaddressed but *unnameable*: no vocabulary exists to identify what was never checked. Cross-pair diagnosis is therefore structurally impossible: a P4 check is insufficient if P3 is unaudited (the agent may comply with policy in actions but fabricate it in communication), and detecting a P1 failure is harder without P5 to reveal whether reasoning ignored the evidence. The gap here is in coverage *reporting*, not coverage *extent*: a benchmark that audits one or two pairs is legitimate if it declares which pairs it excludes and why. The problem is that no work in Table 2 makes such a declaration, so the scope of its procedural claims remains implicit and cross-benchmark comparison is impossible.

Finding 3: Communication-grounding checks lag action-compliance checks. Among these spot-checks, Table 2 exposes a directional asymmetry within matched integrity categories: evaluations more often check what agents do than whether their communications are grounded. Policy compliance (P4) is covered by 17 direct works and 3 adversarial stress tests, whereas policy grounding (P3) is covered by only 3 direct works. Instruction compliance (P8) is covered by 4 direct works and 6 adversarial stress tests, whereas instruction grounding (P7) is covered by only 2 direct works. A customer-service agent could correctly block an unauthorized refund (P4 ✓) while fabricating a nonexistent company policy as justification (P3 unaudited). A benchmark that scores action compliance alone would miss this communication failure.

3.3 Category Patterns

To unpack these three findings, we now summarize how each category is currently measured: what existing works certify, and what remains outside their scope. Appendix E gives the full annotation protocol and ambiguous-case decisions.

Execution integrity (P1–P2). Execution integrity is well represented because P1–P2 checks can be anchored in runtime artifacts that benchmarks already generate. For data grounding (P1),

several works cross-check agent outputs against runtime observations: claim-level Natural Language Inference (NLI) entailment against cited evidence (Chen et al., 2026), per-turn database-result comparison (Acikgoz et al., 2025), and incident-resolution verification against system telemetry (Kamelhar, 2026), with partial checks in oversight and dialogue settings (Kenton et al., 2024; Shourya et al., 2026). For execution compliance (P2), the challenge shifts to whether what the agent *said it did* matches what it *actually did*: speech–action consistency scoring (Xie et al., 2026), compliance checklists detecting hallucinated tool use (Gritta et al., 2026), and action-trace matching against procedural models (Garigipati et al., 2026). These works share a diagnostic conclusion: step-level execution audits surface failures that task-completion metrics systematically miss. They remain local checks, however: an agent that grounds every citation correctly (P1 ✓) may still misrepresent which policy licensed its actions (P3 unaudited).

Policy integrity (P3–P4). Policy compliance (P4) is the largest single-pair cluster. The dominant approach is *rule-based action compliance*: benchmarks supply explicit constraints and score whether agent actions satisfy them, either programmatically (Levy et al., 2024; Li et al., 2025; Nandi et al., 2026; Lyn and Graham, 2025; Anand et al., 2026; Ye et al., 2026b; Winston et al., 2026) or via LLM judges (Zwerdling et al., 2025; Zheng et al., 2026; Rabinovich et al., 2026). A related sub-theme tests whether compliance survives adversarial pressure (Nakash et al., 2025; Zhang et al., 2025; Yang et al., 2026), and several works evaluate compliance indirectly through outcome-based proxies (Ye et al., 2026a; Wang et al., 2026; Balaji et al., 2026; Huang et al., 2026). Their shared diagnostic value is detecting policy-violating successes that task-completion metrics reward, such as completing a booking without verifying required slots or exporting data without a General Data Protection Regulation (GDPR) check. These works also differ substantially in what “policy” means, from short slot-filling rules (Ye et al., 2026b) through branching SOPs (Li et al., 2025; Wang et al., 2026) to lengthy enterprise regulations (Anand et al., 2026; Zwerdling et al., 2025) (Appendix G). However, this cluster is almost entirely action-side: P4 has 17 direct works whereas policy grounding (P3) has only 3 (Chang et al., 2025; Diao et al., 2025; Acikgoz et al., 2025). This ratio is one concrete instance

of Finding 3: the field checks what agents *do* about policy far more than whether they *communicate* it faithfully.

Reasoning integrity (P5–P6). Reasoning integrity asks whether the agent’s internal reasoning is grounded in evidence (P5) and whether it causally drives the agent’s actions (P6). P5 targets factual premises asserted as observations or derived facts, not every entertained proposition: proof by contradiction, abduction, and forward planning are compatible with P5 when conjectural status is preserved; P5 fires only when an unsupported fact enters the reasoning as if established. Evidence-bank methods flag reasoning steps that introduce unsupported claims (Kim et al., 2025; Shao et al., 2025). Counterfactual interventions reveal *causal decoupling*: agents whose intermediate reasoning is negated still produce the same answer (Khanzadeh, 2026). Assertion-conditioned compliance shows that false premises can be silently incorporated while task success remains high (Waqas et al., 2026). Controllability work finds that reasoning traces resist manipulation independently of outputs, with controllability decreasing further with longer reasoning and more RL training (Chen et al., 2025a). These works collectively show that reasoning traces can be checked for evidence grounding and causal faithfulness. Each certifies one local trace property, however; none audits trajectory-level reasoning integrity across tools, policies, instructions, and user-facing communication.

Instruction integrity (P7–P8). Instruction integrity asks whether the agent remains faithful to its deployment instructions (Λ). Two sub-themes emerge. *Direct auditing* is rare. AgentIF (Qi et al., 2025) certifies output and tool-use constraints against explicit instruction specifications (P7 ✓, P8 ✓). ToolEmu (Ruan et al., 2024) certifies safe action compliance in simulated environments (P8 ✓). Agent-SafetyBench (Zhang et al., 2024) partially checks both grounding and compliance. *Adversarial resilience testing* is more common. AgentDojo (Debenedetti et al., 2024) and AgentDyn (Li et al., 2026) test whether instruction adherence survives prompt injection embedded in tool-returned data. AgentHarm (Andriushchenko et al., 2025) tests whether agents refuse explicitly malicious user requests and jailbreaks. CIBER (Ba et al., 2026) tests resilience to prompt injection, memory poisoning, and backdoor attacks ($\diamond$). These stress tests reveal whether a constraint *can*

be broken but do not audit whether the agent faithfully represents its instructions during ordinary operation. This mirrors Finding 3: adversarial compliance tests (P8 $\diamond$) outnumber communication-grounding audits (P7), and routine audits of role, scope limits, and instruction-derived authority remain rare.

User fulfillment (P9). User fulfillment asks whether the agent satisfied the user’s expressed needs, not merely the task’s database-state target. Approaches include LM-simulated users matched against annotated goal states (Yao et al., 2025; Barres et al., 2025), multi-turn need satisfaction scoring (Qian et al., 2025; Qin et al., 2026; He et al., 2025b; Shen et al., 2026), combined user-need and data-grounding audits (Hadad et al., 2026; Gou et al., 2025; He et al., 2025a), and phase-aware compliance in healthcare conversations (Kulkarni et al., 2026). Despite this breadth, user fulfillment remains coupled to task-level success rather than measured independently (Finding 2): most benchmarks lack a standalone user-need specification, so satisfying the database state can be mistaken for satisfying the user.

Quality dimensions and cross-session evaluation. Quality dimensions appear across several works but remain weakly integrated with integrity reporting. Efficiency (Q_1) is measured as step count or latency (Chen et al., 2026; He et al., 2025a; Xue et al., 2025). Resilience (Q_2) tests whether agents recover from misleading retrieval (Chen et al., 2026; Shao et al., 2025) or adversarial oversight protocols (Kenton et al., 2024). Interaction quality (Q_3) covers naturalness, coherence, pedagogic quality, and conversational adequacy (Mehri and Eskenazi, 2020; Pei et al., 2025; Chen et al., 2025b; Zhang et al., 2026). As Finding 2 implies, quality dimensions remain spot-checked in isolation. Cross-session issues such as memory leakage, commitment drift, and procedural inequity remain structurally absent because they require repeated runs, linked sessions, or matched user conditions.

3.4 Reporting Standard for Procedure-Aware Benchmarks

The coverage gaps above motivate a minimal reporting standard. A procedure-aware benchmark should state (i) its outcome metric, (ii) which integrity pairs and quality dimensions it audits, and (iii) which violations invalidate success versus mark lower-severity degradation. It should also

state (iv) which trace artifacts are available, (v) which exclusions remain because a policy, artifact, or cross-episode design is missing, and (vi) how automated judges were validated. This does not prescribe one scoring algorithm. It makes the scope and severity assumptions of any procedural evaluation claim explicit.

4 New Missions for NLP

Section 3 shows that the central gap is not a lack of agent benchmarks, but a lack of methods for evaluating agent procedure. Current evaluations usually check one view at a time, so the final state can be correct while the procedure is unsupported, unauthorized, misleading, or unevenly applied.

Mission 1: What evidence supports agent communication? Finding 3 shows two gaps. First, evaluations check action compliance more often than communication grounding: they ask whether agents follow policies or instructions more often than whether agents explain evidence, policy, or authority faithfully to users. Second, existing grounding checks (Chen et al., 2026; Acikgoz et al., 2025; Kamelhar, 2026; Gou et al., 2025; Diao et al., 2025; Chang et al., 2025; Qi et al., 2025) are mostly local: they compare claims to a retrieved passage, tool output, or database return, but rarely check whether a user-facing message is supported by the full procedure across turns, tool calls, action logs, policies, and instructions. These gaps matter because agent communication is user-facing: a wrong price, fabricated policy, or false completion notice can directly shape a user’s belief and decision. The missing object is an *evidence-backed communication check*: for each agent message, the evaluator asks which observations, actions, policies, instructions, or prior turns would make the message true. A message such as “the price is \$200” should be checked against the latest price returned by the booking tool, and “the policy requires this” should be checked against the policy text. This is broader than standard NLI because evidence may be distributed across time: a policy check may occur in one turn, a price lookup in a later tool call, and an instruction may restrict disclosure. In multi-step and multi-turn settings, each agent message may need checking against relevant prior and current artifacts, not only a single passage or final state. This check complements user fulfillment: an agent may satisfy a user’s goal while misleading the user about price, policy, or what action was taken. *Research di-*

rection: The core NLP task is multi-premise textual entailment where premises span tool returns, action logs, policies, and prior turns. Three decisions remain open. (i) *What to pair*: which user-facing claims require support and which evidence views are admissible for each claim type. (ii) *How to compare*: whether unsupported claims, contradictions, omissions, and misleading statements require separate labels or a severity scale. (iii) *How to report*: benchmarks should state which evidence views are present and which remain unauditable. A concrete deliverable is a cross-view entailment suite with time-aligned evidence and claim-level labels.

Mission 2: What constraints bind agent behavior? Two agents can both fail a policy compliance check, yet one leaked user data to an unauthorized party while the other escalated to a human agent unnecessarily. Without a taxonomy of constraint types, both count equally as policy violations. This problem extends beyond policy: short slot-filling rules (Ye et al., 2026b; Lyn and Graham, 2025), multi-step SOPs with branching logic (Li et al., 2025; Wang et al., 2026), and lengthy enterprise regulations requiring cross-clause interpretation (Anand et al., 2026; Zwerdling et al., 2025) are all scored under the same label, yet they demand different checking logic and carry different severity. Current approaches assume either a single correct path or a finite checklist; the former punishes valid alternatives, the latter invites loophole satisfaction. The core NLP difficulty is that real policies are written in natural language with ambiguity, implicit exceptions, and cross-clause dependencies that resist direct formalization. Without shared constraint semantics, benchmark authors cannot compare what two evaluations test, practitioners cannot tell which constraints their agent satisfies, and Mission 1’s grounding audits cannot determine which constraints a message should reflect. *Research direction*: (i) *Decomposition*: broad requirements, whether from policies, instructions, user specifications, or evidence standards, should be mapped into typed constraints with explicit priority, exceptions, and whether violation vetoes success or marks lower-severity degradation. This connects semantic parsing, legal and policy NLP, process modeling, and conformance checking (Van der Aalst, 2016). (ii) *Coverage reporting*: benchmark authors should report which constraint types their evaluation covers and which remain untested, and the concrete deliverable is a

typed constraint schema that makes the kind and severity of tested rules explicit across benchmarks.

Mission 3: When is reasoning faithful to evidence and action? No benchmark connects evidence grounding, causal faithfulness, and communication into a single reasoning audit. Existing checks verify that reasoning cites tool evidence (Kim et al., 2025), test whether chain-of-thought causally drives the answer (Khanzadeh, 2026), show that false premises can be silently incorporated (Waqas et al., 2026), and find that traces resist manipulation independently of outputs (Chen et al., 2025a). Procedural *evaluation* can work without reasoning: Missions 1 and 2 check communication against evidence, policies, and instructions without requiring traces. But procedural *diagnosis* cannot: reasoning traces let an auditor distinguish a retrieval error from a faulty inference, a misapplied policy, or sycophantic accommodation of a user’s false premise. The missing object is a *procedure-level reasoning-faithfulness audit* that connects evidence, reasoning, action, and communication into a single chain, closing the evaluation-to-improvement loop: evaluation detects that something went wrong; reasoning audits diagnose why. *Research direction:* Three levels should be separated. (i) *Evidence grounding:* each reasoning step should be checked against the observations available at that time, verifying that claims in the trace are supported by the agent’s actual observation history. (ii) *Reasoning-to-action faithfulness:* causal intervention methods (Khanzadeh, 2026) test whether the stated rationale supports the action taken; process reward models (Lightman et al., 2024; Xi et al., 2026) provide step-level training signals but require faithful traces as a precondition. (iii) *Diagnostic attribution:* when an earlier mission detects a failure, trace-based methods should localize it to retrieval, inference, action selection, or communication; single-step checks can be automated, but multi-step diagnosis across tool calls and reasoning requires LLM-as-judge (Zheng et al., 2023; Cao et al., 2025) or agent-as-judge (Jeong et al., 2025) protocols. This connects counterfactual and causal reasoning, process reward modeling, and automated procedural judging to a shared diagnostic target.

Mission 4: Who receives equitable procedure?

The cross-session failures noted in §3 (memory leakage, commitment drift, and procedural inequity) divide into two sub-problems. *Cross-run*

robustness asks whether the same agent, given the same input, satisfies the same constraints across stochastic runs and environmental perturbations; memory isolation failures that leak information across sessions and commitment drift that causes promised exceptions to disappear are examples. *Cross-user fairness* asks whether agents apply the same procedural standards to matched users who differ only in task-irrelevant attributes (Blodgett et al., 2020; Cao et al., 2026b). The goal is not literal procedural identity: communicative adaptation (adjusting tone, vocabulary, or framing) is legitimate as long as it does not change substantive actions such as verification, escalation, or writing. Outcome parity is insufficient: two users may receive the same refund while one is asked for extra verification or bears more conversational burden (Cao, 2025). This mission depends on the prior three: Mission 2’s constraint semantics define which constraints are substantive, Mission 1’s evidence-backed checks detect when communication diverges from action, and Mission 3’s reasoning audits reveal whether divergent rationales reflect genuine deliberation or post-hoc rationalization. *Research direction:* For cross-run robustness, the concrete task is detecting whether repeated runs on the same input produce procedurally inconsistent behavior, such as leaking prior-session data or renegeing on commitments. For cross-user fairness, detection requires three components. (i) *Matched conditions:* define what information is held fixed and which sociolinguistic variation is legitimate to compare, drawing on dialect modeling and style-controlled generation. (ii) *Divergence points:* score divergence across matched traces on every integrity pair and quality dimension in $\mathcal{F}^*$, using conformance checking (Van der Aalst, 2016) for structural differences (an action present for one user but absent for another) and cross-document entailment for semantic differences (the same action performed but with divergent content). (iii) *Adjudication:* for each divergence, decide whether it is harmful, justified, or neutral by extending causal-fairness admissible-variable criteria (Kilbertus et al., 2017) to path-dependent procedural traces; a minimum adequacy floor should block equality-by-degradation. A concrete deliverable is a communication-variation suite shipped alongside benchmark personas.

Shared measurement layer. These missions require a common reporting layer rather than another

single score, as outlined in the reporting norm (§3). Severity matters because an integrity violation such as an unauthorized transfer and a quality degradation such as a verbose explanation should not be collapsed into one score. Judge validation matters because many checks will rely on LLM auditors whose error rates must themselves be measured (Gu et al., 2026). The broader mission for NLP is to build methods that can read, compare, and validate procedures across language, action, policy, reasoning, and interaction.

5 Conclusion

Agent evaluation is moving from outcome-only scoring toward procedure-aware checks, but this transition is incomplete and uncoordinated. This survey organized fragmented procedure-aware work through $\mathcal{F}^*$, a map of nine integrity pairs and three quality dimensions derived by pairing the sub-components of agent procedure. Mapping the reviewed papers against this framework reveals three patterns: most reviewed benchmarks still score only terminal state, so corrupt success (procedurally flawed behavior that produces correct-looking outcomes) remains undetectable; the procedure-aware works that do exist are spot-checking, with no benchmark auditing more than two integrity pairs; and communication grounding lags far behind action compliance. The gaps motivate four new missions for NLP (§4): grounding agent communication in procedural evidence, building constraint semantics for agent behavior, verifying reasoning-to-evidence faithfulness, and ensuring consistent and equitable procedure across runs and communication styles. Realizing this shift requires shared infrastructure: procedure-complete trace schemas, severity-aware reporting formats, and calibrated procedural judges whose error rates are themselves measured.

Limitations

This survey focuses on unifying fragmented procedural checks rather than exhaustively cataloguing benchmark details. First, coverage is limited to English-language publications collected from ACL Anthology, OpenReview and arXiv; agent-evaluation work published in other languages or venues may be missed, and the AI-assisted screening stage may filter out papers whose procedural content is not signaled by title or abstract. Second, we map only what published papers report: de-

ployed systems may implement procedural checks not described in enough detail to annotate. Third, several proposed missions (constraint semantics, comparative procedural fairness) require normative decisions that no taxonomy can settle; the paper flags but does not resolve these choices. Finally, the nine-pair framework is an organizing map for the surveyed literature, not a ready-to-run benchmark, scoring pipeline, or complete domain-specific compliance standard. Not every pair applies to every system; the number of applicable pairs is risk-tiered and varies by deployment context, from dense coverage in policy-governed settings to sparser coverage in sandboxed or adversarial-only evaluations. Future work may refine it for richer agent architectures and modalities.

Disclosure of Generative AI. Generative AI tools were used to assist with grammar correction and proofreading. The authors take full responsibility for the content and integrity of the submitted work.

References

- Emre Can Acikgoz, Carl Guo, Suvodip Dey, Akul Datta, Takyoun Kim, Gokhan Tur, and Dilek Hakkani-Tur. 2025. [TD-EVAL: Revisiting task-oriented dialogue evaluation by combining turn-level precision with dialogue-level comparisons](#). In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 113–132, Avignon, France. Association for Computational Linguistics.
- Ashwani Anand, Ivi Chatzi, Ritam Raha, and Anne-Kathrin Schmuck. 2026. [Mantra: Synthesizing smt-validated compliance benchmarks for tool-using llm agents](#). *Preprint*, arXiv:2605.06334.
- Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. 2025. [Agentharm: A benchmark for measuring harmfulness of LLM agents](#). In *The Thirteenth International Conference on Learning Representations*.
- Lei Ba, Qinbin Li, and Songze Li. 2026. [Ciber: A comprehensive benchmark for security evaluation of code interpreter agents](#). *Preprint*, arXiv:2602.19547.
- Sumanth Balaji, Piyush Mishra, Aashraya Sachdeva, and Suraj Agrawal. 2026. [Beyond IVR: Benchmarking customer support LLM agents for business-adherence](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 193–208, Rabat, Morocco. Association for Computational Linguistics.

- Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. [\$\tau^2\$ -bench: Evaluating conversational agents in a dual-control environment](#). *Preprint*, arXiv:2506.07982.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. [Language \(technology\) is power: A critical survey of “bias” in NLP](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.
- Hongliu Cao. 2025. Writing style matters: An examination of bias and fairness in information retrieval systems. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 336–344.
- Hongliu Cao, Ilias Driouich, Robin Singh, and Eoin Thomas. 2025. [Multi-agent llm judge: automatic personalized llm judge design for evaluating natural language generation applications](#). *Preprint*, arXiv:2504.02867.
- Hongliu Cao, Ilias Driouich, and Eoin Thomas. 2026a. Beyond task completion: Revealing corrupt success in llm agents through procedure-aware evaluation. *arXiv preprint arXiv:2603.03116*.
- Hongliu Cao, Eoin Thomas, and Rodrigo Acuna Agost. 2026b. [When LLMs imagine people: A human-centered persona brainstorm audit for bias and fairness in creative applications](#). *Preprint*, arXiv:2602.00044.
- Hwan Chang, Yumin Kim, Yonghyun Jun, and Hwanhee Lee. 2025. [Keep security! benchmarking security policy preservation in large language model contexts against indirect attacks in question answering](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6780–6800, Suzhou, China. Association for Computational Linguistics.
- Yanyu Chen, Jiyue Jiang, Jiahong Liu, Yifei Zhang, Xiao Guo, and Irwin King. 2026. [Trace: Trajectory-aware comprehensive evaluation for deep research agents](#). In *Proceedings of the ACM Web Conference 2026*, WWW ’26, page 2524–2534, New York, NY, USA. Association for Computing Machinery.
- Yueh-Han Chen, Robert McCarthy, Bruce W. Lee, He He, Ian Kivlichan, Bowen Baker, Micah Carroll, and Tomek Korbak. 2025a. Reasoning models struggle to control their chains of thought. OpenAI Technical Report. Available at https://cdn.openai.com/pdf/a21c39c1-fa07-41db-9078-973a12620117/cot_controllability.pdf.
- Ze Chen, Chengcheng Wei, Jiewen Zheng, and Jiarong He. 2025b. [TCQA²: A tiered conversational Q&A agent in gaming](#). In *Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025)*, pages 289–297, Vienna, Austria. Association for Computational Linguistics.
- Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, and Feng Zhao. 2024. [T-eval: Evaluating the tool utilization capability of large language models step by step](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9510–9529, Bangkok, Thailand. Association for Computational Linguistics.
- Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. 2025c. [Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery](#). In *The Thirteenth International Conference on Learning Representations*.
- Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. [Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. [Mind2web: towards a generalist agent for the web](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA. Curran Associates Inc.
- Lingxiao Diao, Xinyue Xu, Wanxuan Sun, Cheng Yang, and Zhuosheng Zhang. 2025. [Guidebench: Benchmarking domain-oriented guideline following for LLM agents](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11361–11399, Vienna, Austria. Association for Computational Linguistics.
- Purna Sai Garigipati, Onur Ayan, Kishor Chandra Joshi, and Xueli An. 2026. [Beyond state machines: Executing network procedures with agentic tool-calling sequences](#). *Preprint*, arXiv:2605.02584.
- Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, Tianshu Zhang, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, and 7 others. 2025. [Mind2web 2: Evaluating agentic search with agent-as-a-judge](#). *Preprint*, arXiv:2506.21506.
- Milan Gritta, Debjit Paul, Xiaoguang Li, Lifeng Shang, Jun Wang, and Gerasimos Lampouras. 2026. [Process evaluation for agentic systems](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 2678–2692, Rabat, Morocco. Association for Computational Linguistics.

- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2026. [A survey on llm-as-a-judge](#). *The Innovation*, page 101253.
- Shengyue Guan, Jindong Wang, Jiang Bian, Bin Zhu, Jian-Guang Lou, and Haoyi Xiong. 2026. [Evaluating LLM-based agents for multi-turn conversations: A survey](#). *ACM Transactions on Intelligent Systems and Technology*, 17(4):1–40.
- Guy Hadad, Shadi Iskander, Oren Kalinsky, Sofia Tolmach, Ran Levy, and Haggai Roitman. 2026. [Hotelquest: Balancing quality and efficiency in agentic search](#). *Preprint*, arXiv:2602.23949.
- Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, Yue Xing, Jiliang Tang, and Benoit Dumoulin. 2025a. [TRAJECT-Bench: A trajectory-aware benchmark for evaluating agentic tool use](#). *Preprint*, arXiv:2510.04550.
- Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Chengcheng Han, Dengchang Zhao, Hui Su, Kefeng Zhang, Man Gao, Xi Su, Xiaodong Cai, Xunliang Cai, Yu Yang, and Yunke Zhao. 2025b. [Vitabench: Benchmarking llm agents with versatile interactive tasks in real-world applications](#). *Preprint*, arXiv:2509.26490.
- Siyuan Huang, Ziyu Wang, Chao Pan, and Han Zhao. 2026. [Fm so.p: A progressive task mixture framework with automatic evaluation for cross-domain sop understanding](#). *Preprint*, arXiv:2602.09336.
- Yeonseok Jeong, Minsoo Kim, Seung-won Hwang, and Byung-Hak Kim. 2025. [Agent-as-judge for factual summarization of long narratives](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23591–23608, Suzhou, China. Association for Computational Linguistics.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. [SWE-bench: Can language models resolve real-world GitHub issues?](#) In *The Twelfth International Conference on Learning Representations*.
- Federico A. Kamelhar. 2026. [Gsar: Typed grounding for hallucination detection and recovery in multi-agent llms](#). *Preprint*, arXiv:2604.23366.
- Zachary Kenton, Noah Yamamoto Siegel, Janos Kramar, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah Goodman, and Rohin Shah. 2024. [On scalable oversight with weak LLMs judging strong LLMs](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Sourena Khanzadeh. 2026. [Project ariadne: A structural causal framework for auditing faithfulness in LLM agents](#). *Preprint*, arXiv:2601.02314.
- Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. 2017. [Avoiding discrimination through causal reasoning](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 656–666, Red Hook, NY, USA. Curran Associates Inc.
- Wonjoong Kim, Sangwu Park, Yeonjun In, Sein Kim, Dongha Lee, and Chanyoung Park. 2025. [Beyond the final answer: Evaluating the reasoning trajectories of tool-augmented agents](#). *Preprint*, arXiv:2510.02837.
- Shubham Kulkarni, Alexander Lyzhov, Preetam Joshi, and Shiva Chaitanya. 2026. [INSURE-dial: A phase-aware conversational dataset benchmark for compliance verification and phase detection](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5085–5109, Rabat, Morocco. Association for Computational Linguistics.
- Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. 2024. [St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents](#). *Preprint*, arXiv:2410.06703.
- Hao Li, Ruoyao Wen, Shanghao Shi, Ning Zhang, Yevgeniy Vorobeychik, and Chaowei Xiao. 2026. [Agent-dyn: Are your agent security defenses deployable in real-world dynamic environments?](#) *Preprint*, arXiv:2602.03117.
- Zekun Li, Shinda Huang, Jiangtian Wang, Nathan Zhang, Antonis Antoniadis, Wenyue Hua, Kaijie Zhu, Sirui Zeng, Chi Wang, William Yang Wang, and Xifeng Yan. 2025. [Sopbench: Evaluating language agents at following standard operating procedures and constraints](#). *Preprint*, arXiv:2503.08669.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. [Holistic evaluation of language models](#). *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations*.

- Jiawen Lyn and Yvette Graham. 2025. [TransLaTeX: Exposing the last-mile execution gap in LLM-agent for scientific formatting](#). In *Proceedings of The First Workshop on Human-LLM Collaboration for Ethical and Responsible Science Production (SciProdLLM)*, pages 19–24, Mumbai, India (Hybrid). Association for Computational Linguistics.
- Fabrizio Maria Maggi, Michael Westergaard, Marco Montali, and Wil MP van der Aalst. 2011. [Runtime verification of LTL-based declarative process models](#). In *International Conference on Runtime Verification*, pages 131–146. Springer.
- Shikib Mehri and Maxine Eskenazi. 2020. [USR: An unsupervised and reference free evaluation metric for dialog generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.
- Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. [GAIA: a benchmark for general AI assistants](#). In *The Twelfth International Conference on Learning Representations*.
- Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. [Evaluation and benchmarking of llm agents: A survey](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD '25*, page 6129–6139, New York, NY, USA. Association for Computing Machinery.
- Itay Nakash, George Kour, Koren Lazar, Matan Vetzler, Guy Uziel, and Ateret Anaby Tavor. 2025. [Effective red-teaming of policy-adherent agents](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2250–2268, Suzhou, China. Association for Computational Linguistics.
- Subhrangshu Nandi, Arghya Datta, Rohith Nama, Udit Patel, Nikhil Vichare, Indranil Bhattacharya, Prince Grover, Shivam Asija, Giuseppe Carenini, Wei Zhang, Arushi Gupta, Sreyoshi Bhaduri, Jing Xu, Huzefa Raja, Shayan Ray, Aaron Chan, Esther Xu Fei, Gaoyuan Du, Zuhaib Akhtar, and 5 others. 2026. [Sop-bench: Complex industrial sops for evaluating llm agents](#). *Preprint*, arXiv:2506.08119.
- Frans A Oliehoek and Christopher Amato. 2016. [A Concise Introduction to Decentralized POMDPs](#). Springer.
- A. Parasuraman, Valarie A. Zeithaml, and Leonard L. Berry. 1988. SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality. *Journal of Retailing*, 64(1):12–40.
- Jiahuan Pei, Fanghua Ye, Xin Sun, Wentao Deng, Koen Hindriks, and Junxiao Wang. 2025. [Conversational education at scale: A multi-llm agent workflow for procedural learning and pedagogic quality assessment](#). *Preprint*, arXiv:2507.05528.
- Yunjia Qi, Hao Peng, Xiaozhi Wang, Amy Xin, Youfeng Liu, Bin Xu, Lei Hou, and Juanzi Li. 2025. [Agentif: Benchmarking instruction following of large language models in agentic scenarios](#). *Preprint*, arXiv:2505.16944.
- Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. 2025. [Userbench: An interactive gym environment for user-centric agents](#). *Preprint*, arXiv:2507.22034.
- Tian Qin, Felix Bai, Ting-Yao Hu, Raviteja Vemulapalli, Hema Swetha Koppula, Zhiyang Xu, Bowen Jin, Mert Cemri, Jiarui Lu, Zirui Wang, and Meng Cao. 2026. [Compass: Benchmarking constrained optimization in llm agents](#). *Preprint*, arXiv:2510.07043.
- Ella Rabinovich, David Boaz, Naama Zwerdling, and Ateret Anaby-Tavor. 2026. [Near-miss: Latent policy failure detection in agentic workflows](#). *Preprint*, arXiv:2603.29665.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.
- A. Rozinat and W. M. P. van der Aalst. 2008. [Conformance checking of processes based on monitoring real behavior](#). *Inf. Syst.*, 33(1):64–95.
- Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. 2024. [Identifying the risks of LM agents with an LM-emulated sandbox](#). In *The Twelfth International Conference on Learning Representations*.
- Jiaqi Shao, Yuxiang Lin, Munish Prasad Lohani, Yufeng Miao, and Bing Luo. 2025. [Do llm agents know how to ground, recover, and assess? a benchmark for epistemic competence in information-seeking agents](#). *Preprint*, arXiv:2509.22391.
- Yuanzhe Shen, Zisu Huang, Zhengyuan Wang, Muzhao Tian, Zhengkang Guo, Chenyang Zhang, Shuaiyu Zhou, Zengjie Hu, Dailin Li, Jingwen Xu, Kaimin Wang, Wenhao Liu, Tianlong Li, Fengpeng Yue, Fenglu Hong, Cao Liu, and Ke Zeng. 2026. [TRIP-Bench: A benchmark for long-horizon interactive agents in real-world scenarios](#). *Preprint*, arXiv:2602.01675.
- Tanya Shourya, Yingfan Wang, Zhaoyi Joey Hou, Shamik Roy, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiyah. 2026. [When users are happy but agents are wrong: Multi-dimensional evaluation of tool-augmented dialogue](#). *Preprint*, arXiv:2510.19186.
- Zachary S. Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebel, and Arvind Narayanan. 2024.

- Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *Preprint*, arXiv:2409.11363.
- Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, and 11 others. 2024. [Scicode: A research coding benchmark curated by scientists](#). *Preprint*, arXiv:2407.13168.
- Tom R Tyler. 2003. [Procedural justice, legitimacy, and the effective rule of law](#). *Crime and justice*, 30:283–357.
- Wil M. P. Van der Aalst. 2016. *Process Mining: Data Science in Action*, 2nd edition. Springer, Berlin.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. [Voyager: An open-ended embodied agent with large language models](#). *Preprint*, arXiv:2305.16291.
- Jiaming Wang, Zhe Tang, Zehao Jin, Hefei Chen, Yilin Jin, Peng Ding, Xiaoyu Li, and Xuezhi Cao. 2026. [Sop-maze: Evaluating large language models on complicated business standard operating procedures](#). *Preprint*, arXiv:2510.08942.
- Daud Waqas, Aaryamaan Golthi, Erika Hayashida, and Huanzhi Mao. 2026. [Assertion-conditioned compliance: A provenance-aware vulnerability in multi-turn tool-calling agents](#). *Preprint*, arXiv:2512.00332.
- Cailin Winston, Claris Winston, and René Just. 2026. [Solver-aided verification of policy compliance in tool-augmented llm agents](#). *Preprint*, arXiv:2603.20449.
- Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. 2026. [Agentprm: Process reward models for llm agents via step-wise promise and progress](#). In *Proceedings of the ACM Web Conference 2026*, WWW ’26, page 4184–4195, New York, NY, USA. Association for Computing Machinery.
- Sixiong Xie, Zhuofan Shi, Haiyang Shen, Gang Huang, Yun Ma, and Xiang Jing. 2026. [M3-bench: Process-aware evaluation of llm agents social behaviors in mixed-motive games](#). *Preprint*, arXiv:2601.08462.
- Frank F. 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 Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, and 2 others. 2025. [Theagentcompany: Benchmarking llm agents on consequential real world tasks](#). *Preprint*, arXiv:2412.14161.
- Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. 2025. [An illusion of progress? assessing the current state of web agents](#). In *Second Conference on Language Modeling*.
- Zhi Yang, Runguo Li, Qiqi Qiang, Jiashun Wang, Fangqi Lou, Mengping Li, Dongpo Cheng, Rui Xu, Heng Lian, Shuo Zhang, Xiaolong Liang, Xiaoming Huang, Zheng Wei, Zhaowei Liu, Xin Guo, Huacan Wang, Ronghao Chen, and Liwen Zhang. 2026. [Finvault: Benchmarking financial agent safety in execution-grounded environments](#). *Preprint*, arXiv:2601.07853.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. 2025. [{\\$\tau\\$}-bench: A benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser interaction in real-world domains](#). In *The Thirteenth International Conference on Learning Representations*.
- Bowen Ye, Rang Li, Qibin Yang, Yuanxin Liu, Linli Yao, Hanglong Lv, Zhihui Xie, Chenxin An, Lei Li, Lingpeng Kong, Qi Liu, Zhifang Sui, and Tong Yang. 2026a. [Claw-eval: Towards trustworthy evaluation of autonomous agents](#). *Preprint*, arXiv:2604.06132.
- Junjie Ye, Guoqiang Zhang, Wenjie Fu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2026b. [Cctu: A benchmark for tool use under complex constraints](#). *Preprint*, arXiv:2603.15309.
- Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. [Survey on evaluation of LLM-based agents](#). *Preprint*, arXiv:2503.16416.
- Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. 2024. [Assistantbench: Can web agents solve realistic and time-consuming tasks?](#) *Preprint*, arXiv:2407.15711.
- Erxin Yu, Jing Li, Ming Liao, Siqi Wang, Gao Zuchen, Fei Mi, and Lanqing Hong. 2024. [CoSafe: Evaluating large language model safety in multi-turn dialogue coreference](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17494–17508, Miami, Florida, USA. Association for Computational Linguistics.
- Binquan Zhang, Li Zhang, Lin Shi, Song Wang, Yuwei Qian, Linhui Zhao, Fang Liu, An Fu, and Yida Ye. 2026. [An empirical study of interaction smells in multi-turn human-llm collaborative code generation](#). *Preprint*, arXiv:2603.09701.
- Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. 2025. [Agent security bench \(ASB\): Formalizing and benchmarking attacks and defenses in LLM-based agents](#). In *The Thirteenth International Conference on Learning Representations*.
- Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. 2024. [Agent-SafetyBench: Evaluating the safety of LLM agents](#). *Preprint*, arXiv:2412.14470.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-bench and chatbot arena](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Ye Zheng, Yimin Chen, and Yidan Hu. 2026. [Audagent: Automated auditing of privacy policy compliance in ai agents](#). *Preprint*, arXiv:2511.07441.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*.

Naama Zwerdling, David Boaz, Ella Rabinovich, Guy Uziel, David Amid, and Ateret Anaby Tavor. 2025. [Towards enforcing company policy adherence in agentic workflows](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 595–606, Suzhou, China. Association for Computational Linguistics.

A Connections to Adjacent Disciplines

Procedure-aware evaluation is not a new problem outside NLP. Three adjacent disciplines have studied it for decades, and each offers methods that the agent evaluation community has not fully absorbed.

Business Process Management (BPM). BPM studies conformance between execution traces and normative process models (Van der Aalst, 2016). Conformance checking algorithms (Rozinat and van der Aalst, 2008) take a process model and an event log and return a fitness score quantifying how well the log reproduces the model. The analogy to agent evaluation is direct: the process model corresponds to a policy or SOP, the event log corresponds to the agent’s action trace, and fitness corresponds to policy compliance (P4). Constraint types derivable from Declare templates, such as existence, ordering, mutual exclusion, and response dependencies, further show how process rules can be made machine-checkable for agent settings (Maggi et al., 2011). What BPM adds is a mature algorithmic toolkit (token replay, alignment-based conformance, precision and generalization metrics) that NLP has not adopted. A key limitation is that BPM conformance operates on discrete action sequences and does not handle natural-language communication auditing (P3, P7, P9), which is where NLP methods are essential.

Service Science. Service Science studies quality beyond outcome correctness. The SERVQUAL model (Parasuraman et al., 1988) decomposes service quality into five dimensions (tangibles, reliability, responsiveness, assurance, and empathy) measured as gaps between customer expectations and perceived performance. For agent evaluation, these frameworks motivate measuring interaction burden (Q_3) and user fulfillment (P9) as first-class targets rather than afterthoughts. They also suggest that agent benchmarks should include user-expectation specifications alongside task specifications.

Procedural Justice. Procedural Justice research demonstrates that people evaluate fairness based on how a decision was reached, not only on what the decision was (Tyler, 2003). Factors such as voice (being heard), consistency (same rules for everyone), and transparency (understanding the rationale) determine perceived fairness independently of outcome favorability. For agents, this implies that procedural evaluation should eventually test whether agents apply the same verification standards across users, explain decisions transparently, and allow meaningful user input, concerns that map directly to our cross-session fairness mission.

These traditions converge on the same principle: outcomes are insufficient evidence for process quality. The agent era brings NLP to that same conclusion, but now in a form that requires unified evaluation methodology. Such a methodology should combine BPM’s trace-level auditing, Service Science’s interaction-burden measures, and Procedural Justice’s fairness criteria with NLP’s ability to audit natural-language communication. Together, these traditions also support the reporting standard above: benchmark papers should disclose not only scores, but the procedural evidence and coverage assumptions behind those scores.

B Pair Enumeration: Retained and Excluded Cross-Products

The $\mathcal{F}^*$ equation includes $\mathcal{E}_t$ (environment state), but $\mathcal{E}$ anchors *outcome* evaluation (§2): mismatches between $\mathcal{E}$ and any other component are either infrastructure failures (the environment returns incorrect observations) or outcome-evaluation targets ($\mathcal{E}_T$ does not match the goal), not agent procedural faults. The integrity pairs therefore enumerate over the eight *process* sub-components: Λ , A^R , A^W , A^C , O^{sys} , O^{ctx} , O^{usr} , and ϕ (28 unordered pairs). Table 1 retains nine integrity pairs covering

16 of these raw pairs; some aggregate related sub-components (P5 uses O for all observation types; P6 uses $\mathcal{A}$ for all action types) because the consistency question is the same regardless of source or action kind. The remaining 12 cross-products are excluded because they fall into four categories that do not yield independent, agent-attributable consistency checks.

- **Observation–observation pairs** ($O^{sys}-O^{ctx}$, $O^{sys}-O^{usr}$, $O^{ctx}-O^{usr}$): Consistency among the inputs the agent receives is a task-design or infrastructure property, not an agent-level fault.
- **Instruction–input pairs** ($\Lambda-O^{sys}$, $\Lambda-O^{ctx}$, $\Lambda-O^{usr}$, $\Lambda-\phi$): Instructions constrain what the agent *does*, not what it *receives* or *thinks*. $\Lambda-\phi$ is not independently checkable: any instruction-violating reasoning that matters will surface as an action that violates instructions (P7 or P8) or a reasoning trace that fails to drive its own conclusion (P6).
- **Generative pairs** ($O^{sys}-A^R$, $O^{sys}-A^W$): A^R produces O^{sys} (a read action returns an observation); this is a causal relationship, not a consistency check the agent can violate. $O^{sys}-A^W$ is similarly generative: a write action changes state, and subsequent observations reflect the new state.
- **Mediated pairs** ($O^{usr}-A^R$, $O^{usr}-A^W$, A^R-A^W): Whether the agent’s reads and writes fulfill the user’s request is mediated: user fulfillment is checked through communication (P9), and whether specific reads or writes were appropriate is checked through policy and instruction compliance (P4, P8). Cross-action sequencing (e.g., reading the right record before writing) is captured by Order and Conditional constraints on P4 and P8, not by an independent pair.

This enumeration does not claim that the nine pairs are the only possible decomposition; a finer-grained formalization could split aggregated pairs or introduce environment-level checks. The claim is that every process-level cross-product is accounted for: each is either retained as an integrity pair or excluded for a stated reason.

C Justification for the Prior-Survey Gap Table

We apply a conservative annotation rule to Table 3. A $\checkmark$ means the cited work treats the column as a central organizing goal; $\sim$ means it offers adjacent or partial coverage; $-$ means the feature is not a

Table 3: Gap table relative to nearby surveys and evaluation frameworks. Columns correspond to this paper’s claimed structural contributions; symbols mark emphasis rather than overall value. $\checkmark$ = central focus; $\sim$ = partial or adjacent coverage; $-$ = not a primary focus.

Survey	Agentic	Proc.-centric	Integ. tax.	Qual. tax.	Cov. matrix
KDD agent eval. (Mohammadi et al., 2025)	$\checkmark$	$\sim$	$-$	$\sim$	$-$
ACL agent eval. (Yehudai et al., 2025)	$\checkmark$	$\sim$	$-$	$\sim$	$-$
Multi-turn agents (Guan et al., 2026)	$\checkmark$	$\sim$	$-$	$\sim$	$-$
LLM-as-judge (Gu et al., 2026)	$\sim$	$-$	$-$	$-$	$-$
Holistic LM eval. (Liang et al., 2023)	$-$	$-$	$-$	$\sim$	$\sim$
This survey	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

Columns: Proc.-centric = procedure as the central unit of analysis; Integ. tax. = taxonomy of hard consistency pairs (P1–P9); Qual. tax. = taxonomy of quality dimensions (Q1–Q3), explicitly separated from integrity; Cov. matrix = systematic paper $\times$ dimension annotation.

primary focus. We verified each row against the corresponding survey paper.

KDD agent evaluation survey (Mohammadi et al., 2025). $\checkmark$ *Agentic*: the survey is explicitly about evaluation and benchmarking of LLM agents. $\sim$ *Proc.-centric*: its evaluation-process axis covers interaction modes, datasets, benchmarks, metric computation, and enterprise issues such as dynamic and long-horizon interactions, but it does not organize the field around procedure as the central unit of analysis. $-$ *Integ. tax.*: its two-dimensional taxonomy separates evaluation objectives from evaluation processes rather than isolating hard consistency pairs. $\sim$ *Qual. tax.*: it discusses metrics such as efficiency, cost, and robustness, but does not separate them as a distinct layer from correctness. $-$ *Cov. matrix*: it organizes agent-evaluation work into topical categories but does not produce a systematic paper $\times$ dimension annotation.

ACL agent evaluation survey (Yehudai et al., 2025). $\checkmark$ *Agentic*: the survey defines LLM agents as systems that plan, reason, use tools, and interact with dynamic environments. $\sim$ *Proc.-centric*: it discusses static versus dynamic environments, offline traces, trajectory-based assessment, and reference-based trajectory alignment, but treats these as one part of the broader benchmark landscape rather than as the central unit of analysis. $-$ *Integ. tax.*: its five perspectives; agent capabilities, applica-

tion benchmarks, generalist agents, benchmark dimensions, and developer frameworks/tools; do not distinguish integrity relations. $\sim$ *Qual. tax.*: it discusses cost-efficiency, safety, and robustness as evaluation dimensions but does not separate them from correctness as a distinct quality layer. – *Cov. matrix*: it analyzes benchmark dimensions and trends but does not map individual papers against a fixed set of procedural dimensions.

Multi-turn conversational-agent survey (Guan et al., 2026). $\checkmark$ *Agentic*: the survey studies LLM-based agents for multi-turn conversations, including tool use, memory, and planning. $\sim$ *Proc.-centric*: it emphasizes multi-turn dynamics, error propagation, context retention, and planning processes, but does not require full procedure-complete action–communication traces. – *Integ. tax.*: its categories include task completion, memory, and planning consistency, which are capability and performance dimensions rather than hard consistency relations between actions, communications, and instructions. $\sim$ *Qual. tax.*: it covers response quality, user experience, and evaluation methodology, which are adjacent to quality dimensions but not separated as a distinct layer from integrity. – *Cov. matrix*: it synthesizes evaluation goals, methods, datasets, and metrics across nearly 250 sources but does not produce a paper $\times$ dimension annotation.

LLM-as-judge survey (Gu et al., 2026). $\sim$ *Agentic*: the survey defines LLM-as-a-judge broadly enough to evaluate data, models, actions, decisions, and agents, but its primary object is the judging methodology. – *Proc.-centric*: reliability, bias, prompt design, and benchmark construction for judges are central, whereas procedure-level agent analysis is not. – *Integ. tax.*: the organizing taxonomy concerns judge reliability and bias rather than procedural integrity pairs. – *Qual. tax.*: judge quality (consistency, correlation with humans) is discussed, but not as a quality taxonomy for agent procedures. – *Cov. matrix*: it does not map agent-evaluation papers against procedural dimensions.

HELM (Liang et al., 2023). – *Agentic* and *Proc.-centric*: HELM evaluates language models under standardized scenarios and metrics, not autonomous agents or procedure-level analysis. – *Integ. tax.*: it does not model procedural integrity pairs. $\sim$ *Qual. tax.*: HELM centrally argues for multi-metric evaluation across accuracy, calibra-

tion, robustness, fairness, bias, toxicity, and efficiency; this is adjacent to our quality layer but not separated from correctness as a distinct dimension. $\sim$ *Cov. matrix*: HELM taxonomizes scenarios and metrics and reports what is missing or underrepresented, but the coverage unit is model-scenario-metric rather than surveyed paper versus procedural dimension.

D Search Protocol

We provide the full details of each pipeline stage below. Table 4 summarizes the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)-aligned flow.

Table 4: PRISMA-aligned search and screening flow.

Stage	Description	n
Identification	3 databases (2020–2026)	3,392
Deduplication	Fuzzy title matching	3,342
Tier 2	Procedural keyword screening	$\approx$ 1,030
Tier 3	Strict eligibility filters	360
Tier 4	AI-assisted rubric screening	169
Manual review	Full-text reading of all 169	169

Tier 1: Identification and deduplication. Three databases were searched for papers published January 2020–April 2026. *ACL Anthology*: the full collection ($>50k$ papers) was scanned via title/abstract regex using the *acl-anthology* Python package. *arXiv*: keyword queries across categories *cs.CL*, *cs.AI*, *cs.SE*, *cs.CR*, *cs.LG*, *cs.MA*, *cs.HC*, and *cs.IR*, covering eight query axes: six aligned to the $\mathcal{F}^*$ procedural sub-groups used in Tier 2 below (execution, reasoning, instruction, policy, user fulfillment, quality) plus two breadth axes (core agent evaluation and adjacent procedural work) to ensure recall. *OpenReview*: venue-year pagination for ICLR 2022–2026, NeurIPS 2021–2024, ICML 2023–2024, and COLM 2024. The search used two keyword axes: $A = \{\text{agent terms} \wedge \text{evaluation terms}\}$. Agent terms include *agent(s/ic)*, *tool-use/call*, *function call**, *trajectory/ies*, *multi-step/turn*, *autonomous*, and *action space*. Evaluation terms include *benchmark*, *evaluat**, *metric(s)*, *assess**, *audit**, *scoring*, *leaderboard*, *test suite/bed/bench*, *diagnos**, *verif**, *monitor**, and *oversight*. Fuzzy title deduplication (Levenshtein token-sort ratio ≥ 82) reduced 3,392 records to 3,342 unique candidates.

Tier 2: Procedural keyword screening. Each candidate’s title and abstract were screened against

six procedural-signal sub-groups aligned to the $\mathcal{F}^*$ framework:

- **P_{exec}** (P1–P2): tool-call correctness, trajectory evaluation, execution trace, action grounding, step-level metrics.
- **P_{reason}** (P5–P6): reasoning compliance, chain-of-thought evaluation, rationale verification, process reward, agent hallucination detection.
- **P_{instr}** (P7–P8): instruction following, prompt injection, jailbreak, agent safety/security, red-teaming, deployment instructions.
- **P_{policy}** (P3–P4): policy compliance/adherence/violation, SOP, guideline following, rule following, constraint satisfaction, conformance, procedural integrity.
- **P_{user}** (P9): user satisfaction, multi-turn evaluation, dialogue/conversational evaluation, interaction quality.
- **P_{quality}** (Q1–Q3): efficiency, resilience, error recovery, robustness, scalable oversight, LLM-as-judge.

To control false positives from generic terms (*robustness*, *hallucination*, *efficiency*), we applied tiered confidence scoring. *Specific* groups (P_{exec}, P_{instr}, P_{policy}, P_{user}) qualified a paper at any match. *Generic* groups (P_{quality}, P_{reason}) required co-occurrence with an agent-procedural context signal, such as *multi-step*, *tool-call*, or *trajectory evaluation*. Negative exclusion lists removed non-procedural uses, such as *knowledge grounding*, *visual grounding*, *faithful summarization*, *distributionally robust*, *parameter efficient*, and *medical hallucination*. This yielded approximately 1,030 high-confidence procedurally relevant papers.

Tier 3: Strict eligibility filters. Three filters were applied sequentially.

(i) *Six-level procedure-aware classifier.* Each candidate was classified into one of six priority-ordered levels: L1 (trajectory/execution evaluation), L2 (multi-turn/dialogue evaluation), L3 (process/reasoning evaluation), L4 (safety/compliance evaluation), L5 (outcome-only evaluation), or L6 (system/framework paper). Each level used specific, moderate, and excluded keyword sets with confidence flags. Only L1–L4 papers were retained as procedure-aware.

(ii) *Evaluation-focus filter.* Papers were required to have evaluation-signal keywords in the title or ≥ 2 evaluation terms in the abstract, removing agent-system papers that incidentally mention evaluation.

(iii) *Targeted false-positive removal.* Five FP patterns were applied: (FP1) PRM-as-training: papers using process reward models for RL training without proposing an evaluation contribution. (FP2) pure method papers: system proposals without evaluation keywords in the title. (FP3) dialogue-system builders: papers proposing dialogue agents without evaluation. (FP4) math PRMs: process reward models for mathematical reasoning without agent context. (FP5) generic NLP: tabular/sentiment/NER/translation papers without agent signals. A second pass removed method papers without evaluation or safety/agent title signals, irrelevant-domain papers (personality, speech recognition, medical summarization), and pure attack papers without evaluation methodology. This reduced the pool to 360 strictly eligible candidates.

Tier 4: AI-assisted rubric screening. Each paper’s title and abstract were scored against a structured rubric with 10 inclusion and 8 exclusion criteria aligned to the $\mathcal{F}^*$ framework:

Inclusion criteria (INCLUDE if any are met): (I1) benchmark/dataset scoring agent trajectories or step correctness. (I2) tool-use correctness evaluation. (I3) reasoning compliance/integrity evaluation. (I4) policy/SOP compliance evaluation. (I5) instruction adherence evaluation. (I6) user fulfillment evaluation across turns. (I7) multi-turn dialogue quality with process-level metrics. (I8) process-level metric or judge for agent behavior. (I9) agent safety/security via procedural violation testing. (I10) efficiency, resilience, or robustness of agent procedures.

Exclusion criteria (EXCLUDE if any are met): (E1) proposes agent system without evaluation contribution. (E2) outcome-only evaluation. (E3) non-agentic LLM. (E4) training/optimization only. (E5) outcome-level safety judging. (E6) domain-irrelevant. (E7) pure attack paper without evaluation methodology. (E8) general survey without evaluation analysis.

Each paper received an INCLUDE, EXCLUDE, or BORDERLINE decision. This produced 169 included papers plus 15 borderline cases for manual review. All 169 papers (plus borderline cases and citation-chained additions) were then read in full text.

Notable exclusions. Several prominent agent benchmarks were reviewed in full text but excluded from Table 2 because their evaluation pipelines score only final outcomes without auditing any

procedural dimension. *Outcome-only benchmarks* such as SWE-bench (Jimenez et al., 2024), WebArena (Zhou et al., 2024), Mind2Web (Deng et al., 2023), GAIA (Mialon et al., 2024), AssistantBench (Yoran et al., 2024), TheAgentCompany (Xu et al., 2025), and ScienceAgentBench (Chen et al., 2025c) judge success by final-state correctness. Their criteria include patch acceptance, element match, or exact answer, without tracing how the agent arrived at that state. *Component-level benchmarks* such as SciCode (Tian et al., 2024) and CORE-Bench (Siegel et al., 2024) test isolated code-generation or reproduction steps but do not evaluate multi-step procedural integrity. *Safety benchmarks without procedural auditing* such as CoSafe (Yu et al., 2024) measure whether harmful content appears in the output but do not trace whether a specific policy document was consulted or followed step by step. Finally, *dialogue-level metrics* such as SGD (Rastogi et al., 2020) evaluate slot-filling accuracy across turns without checking whether the agent’s communicated commitments matched its executed actions. These exclusions illustrate that outcome-level success, even on complex agentic tasks, does not entail procedural coverage under our annotation rule.

Qualitative error analysis of excluded papers.

To verify that the 115 excluded papers do not reveal procedural signals outside $\mathcal{F}^*$, we examined representative papers from each exclusion pattern and grouped them into three recurring categories.

- *Terminal-state benchmarks* (e.g., SWE-bench, WebArena, GAIA, TheAgentCompany): procedure is discussed motivationally but never scored. The evaluation pipeline checks whether the final state matches a specification (patch passes tests, element matches target, answer equals gold), so any procedural violation that produces a correct-looking outcome is invisible to the metric.
- *Attack-outcome safety tests* (e.g., CoSafe, R-Judge): these score whether a multi-turn attack succeeded (harmful content produced or not), i.e., an outcome-level safety judgment rather than a diagnosis of which procedural relation (policy grounding, instruction compliance, reasoning faithfulness) failed during the attack.
- *Gold-label planning checks* (e.g., ToolBench action-matching, API-Bank parameter scoring): predicted tool calls or parameters are compared to a ground-truth label. This tests planning *capability* (can the model produce the right call?)

rather than runtime *consistency* (did the agent’s communication match what actually happened at execution time?), which is the focus of P1 and P2.

In each case, the exclusion reflects a *measurement* choice (procedure not scored) rather than a procedural signal that $\mathcal{F}^*$ cannot name. All 169 reviewed papers, including the 115 excluded from Table 2, were classifiable within $\mathcal{F}^*$ ’s categories; the framework could name the procedural property discussed even when the paper did not score it.

E Annotation Protocol for the Coverage Matrix

We apply a conservative paper-level annotation protocol to Table 2. The goal is not to judge benchmark quality, but to identify which procedural dimensions are clearly supported by each paper’s stated tasks, artifacts, metrics, and scoring pipeline. The search and screening pipeline that produced the 169 manually reviewed papers is detailed in Appendix D.

Annotator process. Three annotators performed the full-text review. The annotation rubric was iteratively refined over three rounds: an initial binary (procedural/non-procedural) scheme was replaced by the three-level direct/partial/adversarial scheme ($\checkmark/\sim/\diamond$) after the first calibration round revealed that binary labels conflated explicit audits with indirect evidence. Disagreements were resolved through discussion under the final protocol. The final protocol was then applied uniformly to all 169 papers, ensuring that labels reflect the converged criteria rather than earlier rubric versions.

Annotation decision rule. Assigning a symbol requires answering a precise question: does the paper’s evaluation pipeline contain a scored metric, judge, rubric, programmatic assertion, or reported qualitative analysis that directly audits the integrity pair or quality dimension in question? We code only what is reported in the paper. The evidence may come from systematic metrics, rubric-based judges, programmatic checks, manual error taxonomies, failure breakdowns, or case-study analyses. It cannot come from capabilities that the system may have implemented without describing them.

Cell labels. A $\checkmark$ means the paper explicitly audits the relevant integrity pair or quality dimension

with a dedicated, scored metric, judge, rule, or annotation procedure whose primary purpose is to audit that dimension. For example, a document-grounding score that checks whether a summary reflects retrieved documents counts for P1. A rule-level pass/fail judge that checks whether a response violates specified guidelines counts for P3. A $\sim$ means the paper provides partial or indirect evidence: it touches the pair through an incidental or adjacent mechanism rather than a dedicated audit. For example, outcome-based policy rewards touch P4 without tracing whether the policy was followed step by step. Reference-trajectory matching touches P2 without checking whether verbal claims matched actual actions. A $\diamond$ means the paper tests adversarial or outcome-based resilience related to the pair, such as whether instruction following or policy adherence survives attack. It does not directly audit the procedural mechanism of failure during ordinary successful execution. A blank cell means that we found no usable evidence that the paper evaluates that dimension under our annotation rule.

Exclusion principles. We exclude several common patterns that resemble procedural audits but do not meet the annotation rule. First, post-hoc analytical findings do not count unless an agent receives a score for the dimension. Observing that shorter trajectories correlate with success is not an efficiency metric. Second, incidental outcome entailment does not count as a direct audit. If an outcome metric would miss a procedural violation whenever the final state remains correct, the pair is not directly covered. Third, agent-side policies or constraints do not count unless the evaluator checks them. A system prompt that forbids fabrication is not a P3 audit if the scorer only checks final-answer correctness. Fourth, rubric-based outcome evaluation is not automatically user-fulfillment tracking. For P9, the evaluation must trace alignment between user utterances and subsequent agent behavior, not merely verify that requirements were eventually satisfied. Fifth, task-correctness checks are not runtime consistency audits. Comparing predicted tool calls or parameters against gold labels tests planning capability, whereas P1 and P2 require runtime signals comparing communicated claims to observed tool outputs or executed actions. Sixth, evaluation-protocol properties such as $\text{pass}@k$ or inter-annotator agreement describe the measurement instrument, not the agent’s procedu-

ral quality. Seventh, aggregate gaps between independently scored views, such as reasoning quality versus communication quality, do not count as integrity audits unless the evaluator cross-checks specific claims, commitments, or conclusions across views at the content level.

Evaluation paradigm. The **Eval** column records the scope of the evaluation setting. **S** denotes single-response benchmarks where the model receives one prompt and produces one response, without a multi-step procedure. **MS** denotes multi-step agentic trajectories triggered by a single user instruction, typically involving tool calls or environment interactions. **MT** denotes multi-turn user-agent conversation without tool-based actions. **MS+MT** denotes benchmarks that combine both multi-turn conversation and multi-step tool or environment actions. This distinction contextualizes the coverage claims. Procedural integrity is most meaningful, and hardest to audit, when actions unfold over time and must remain consistent across steps.

Counting rules. For the category counts in §3, a paper is counted once per category even if it covers multiple cells within that category. Direct coverage combines $\checkmark$ and $\sim$, while $\diamond$ is reported separately because adversarial success or failure does not necessarily imply direct procedure auditing. The statement that no benchmark explicitly audits more than two integrity pairs counts only $\checkmark$ cells; this maximum remains two even when $\checkmark$, $\sim$, and $\diamond$ labels are counted together.

Ambiguous cases. When a benchmark reports task success under a policy but does not inspect the trajectory, we code policy compliance as $\sim$ rather than $\checkmark$. When a benchmark attacks an agent with prompt injection or harmful requests and scores whether the final outcome is safe, we use $\diamond$ for the relevant instruction or policy dimension. We use a stronger label only if the scoring pipeline directly checks the action-instruction or action-policy relation. When a dialogue benchmark scores overall response quality without a user-need specification, we code interaction quality when appropriate. We do not code user fulfillment unless the user’s expressed request is independently checked. When a paper logs tool calls but scores only final state, we do not infer execution compliance unless the communicated claim or action trace is compared to the logged action or observation.

Per-paper audit principle. Each row in Table 2 was coded by matching the paper’s stated evaluation target to the closest integrity pair or quality dimension in Table 1. If multiple interpretations were possible, we assigned the weaker label. This conservative rule is why several agent benchmarks with rich trajectories still have sparse cells: trace availability alone is not counted as coverage unless the paper uses the trace in evaluation.

F Per-Paper Justifications

This appendix provides per-paper justifications for every non-empty cell in Table 2, explaining both why each checked pair is assigned and why plausible but ultimately unsupported pairs are excluded. Each subsection also lists excluded pairs; these follow a default pattern (most papers do not model all $\mathcal{F}^*$ sub-components), so we include them primarily to document non-obvious exclusion decisions where a reader might expect coverage.

F.1 TRACE (Deep Research)

TRACE (Chen et al., 2026) introduces a Hierarchical Trajectory Utility Function and diagnostic tools for Deep Research Agents, measuring Evidence Grounding, Process Efficiency, Reasoning Robustness, and cross-run Strategic Profiling.

P1 (Data Grounding): ✓. Evidence Grounding ($\mathcal{G}_E$) decomposes the agent’s final answer into atomic claims and checks NLI entailment against cited evidence, directly auditing whether communicated content is grounded in retrieved observations ($O^{sys}-A^C$).

$\mathcal{Q}_1$ (Efficiency): ✓. Process Efficiency ($\mathcal{E}$) quantifies exploration parsimony via a trajectory cost functional with a Redundant Exploration Penalty and Marginal Information Gain; a direct measure of trajectory efficiency.

$\mathcal{Q}_2$ (Resilience): ✓. Reasoning Robustness ($\mathcal{R}_R$) measures recovery latency from “information traps” (misleading but plausible information) as a dedicated, mathematically formalized scored metric for self-correction capability.

Excluded pairs. TRACE does not model domain policy documents (O^{ctx}), agent instructions (Λ), user intent (O^{usr}), or internal reasoning traces (ϕ_t), and does not check whether claimed actions were actually executed (P2). Latent Capability and Strategic Profiling are cross-procedure diagnostics rather than per-session integrity pairs.

F.2 TD-EVAL

TD-EVAL (Acikgoz et al., 2025) introduces three turn-level metrics for task-oriented dialogue (conversation cohesion, backend knowledge consistency, and policy compliance) scored on a 1–5 Likert scale by an LLM judge.

P1 (Data Grounding): ✓. “Backend knowledge consistency” checks each agent response against the *runtime* database results the agent actually received at that turn (e.g., flagging an agent that says “I found a restaurant” when the DB returned zero results). The LLM judge is provided the actual DB output, making this a runtime $O^{sys}-A^C$ audit rather than a gold-label comparison.

P3 (Policy Grounding): ∼. “Policy compliance” checks whether the agent’s responses adhere to domain-specific policy protocols (e.g., “request required slots before booking”). The agent receives the same policy instructions that the evaluator uses, making this a genuine cross-check of policy content against agent communication. However, the policies are lightweight conversational-flow rules rather than detailed multi-step procedural protocols with inter-step dependencies; we therefore assign ∼ rather than ✓.

Excluded pairs. “Conversation cohesion” is a general coherence rubric, not user-fulfillment tracking (Exclusion 4: P9 requires tracing user utterances to agent actions, not just on-topic coherence). No execution compliance (P2), reasoning traces (P5/P6), or system instructions (P7/P8) are modeled.

F.3 Process Evaluation for Agentic Systems

Process Evaluation (Gritta et al., 2026) is a position paper introducing a compliance-checklist methodology for auditing agent procedures on GAIA-style research tasks, scored by an LLM judge with full trajectory access.

P2 (Execution Compliance): ∼. The paper documents cases where agents “hallucinate tool use” (e.g., o3-mini simulating a code interpreter), which is conceptually a P2 violation. However, the compliance checklist measures this as a protocol-adherence check (“did the agent use a code interpreter?”) rather than a cross-check between the agent’s verbal claims and its actual tool-call log. We assign ∼ because the hallucination-detection concept is adjacent to P2 but the scoring mecha-

nism does not perform a dedicated communication-action cross-check.

P8 (Instruction Compliance): $\sim$. The compliance checklist specifies behavioural norms such as “use a code interpreter for calculations,” which resemble deployment-level instructions (Λ). However, Λ in $\mathcal{F}^*$ is a static, task-independent deployment configuration that the *agent* receives, whereas the paper’s “recommended protocol” is a per-task *evaluator-side* rubric that the agent never sees. We assign $\sim$ because the tool-use norm questions are adjacent to instruction compliance but the mechanism does not separate deployment instructions from task-specific procedure steps (Exclusion 5).

Excluded pairs. P1 is not assigned because checklist questions verify whether the agent obtained information from the correct source (a protocol-adherence check), not whether the agent’s communicated content faithfully reflects what the tool returned (the core $O^{sys}-A^C$ relationship). No policy documents (P3/P4), reasoning traces (P5/P6), instruction grounding (P7), user fulfillment (P9), or quality dimensions (Q_1-Q_3) are modeled.

F.4 GSAR

GSAR (Kamelhar, 2026) introduces a four-way claim partition (grounded, ungrounded, contradicted, complementary) and evidence-type-specific weights for grounding evaluation in multi-agent diagnostic systems.

P1 (Data Grounding): $\checkmark$. The groundedness score checks whether each claim in a specialist agent’s diagnostic report is supported by tool-observed evidence, directly measuring $O^{sys}-A^C$. Evidence-type weighting assigns higher epistemic strength to tool-verified observations than to model-inferred conclusions. The three-tier recovery loop (proceed/regenerate/replan) is agent-side infrastructure (Exclusion 3), but the groundedness score itself is evaluator-side.

Excluded pairs. The framework does not expose internal reasoning traces (P5/P6), does not model Λ (P7/P8), does not check policy documents (P3/P4), and does not audit user fulfillment (P9). No quality dimensions (Q_1-Q_3) are measured.

F.5 TRACE (Tool-Augmented Agents)

TRACE (Kim et al., 2025) evaluates reasoning trajectories of tool-augmented agents without requiring

a single gold trajectory, building an *evidence bank* from prior tool calls and scoring along three axes: efficiency, hallucination, and adaptivity.

P5 (Reasoning Grounding): $\checkmark$. The *hallucination* metric checks at each step whether the agent’s current thought is grounded in the accumulated evidence bank built from prior tool outputs. This directly measures reasoning grounding: the benchmark flags thoughts that introduce unsupported factual claims or contradictions not derivable from observations already collected.

Q_1 (Efficiency): $\checkmark$. TRACE explicitly measures trajectory efficiency by identifying the minimal evidence subset needed to derive the final answer and treating the remainder as unnecessary exploration. This is a direct measure of procedural parsimony.

Q_2 (Resilience): $\checkmark$. The *adaptivity* metric evaluates whether an agent recovers after an unavailable-tool failure by acknowledging the failure and choosing a sensible alternative strategy, rather than repeating the same call or stalling.

Excluded pairs. TRACE does not evaluate communicated-content grounding (P1), claimed-action execution (P2), policy conformance (P3/P4), causal control from reasoning to action (P6), instruction following (P7/P8), or user fulfillment (P9). Q_3 (Interaction Quality) is not addressed.

F.6 A-CC

Assertion-Conditioned Compliance (A-CC) (Waqas et al., 2026) evaluates sycophantic compliance in multi-turn function-calling dialogues. Built on the Berkeley Function-Calling Leaderboard v3, it injects misleading assertions from two sources: user-sourced assertions (USAs) and function-sourced assertions (FSAs). It then measures whether agents incorporate them into their tool-calling pipelines.

P6 (Reasoning Compliance): $\checkmark$. A-CC directly tests whether the agent’s reasoning process rejects false premises or blindly complies with them. USAs simulate sycophancy where the agent follows plausible but misinformed user beliefs rather than grounding its actions in tool documentation and prior execution history. FSAs inject contradictory “system policy notes” into tool response payloads, testing whether agents treat them as authoritative despite contradicting function documentation. The paper explicitly distinguishes this from

prompt injection: “A-CC demonstrates contextual trust: the agent maintains its helpful objective but fails to verify false premises against the tool history.” This is a P6 violation where reasoning should reject the false premise but actions comply. A key finding is that compliance and task success form *distinct axes*: “quiet” compliance (the agent complies but still succeeds) coexists with strong benchmark performance, representing latent procedural risk invisible to outcome metrics.

Excluded pairs. A-CC does not model domain policy documents (P3/P4), agent instructions (P7/P8), or internal reasoning traces directly (P5), and does not evaluate data grounding (P1), execution compliance (P2), or user fulfillment (P9). No quality dimensions are scored.

F.7 Ariadne

Project Ariadne (Khanzadeh, 2026) introduces a causal auditing framework for reasoning faithfulness in LLM agents. It models an agent’s chain-of-thought as a Structural Causal Model (SCM) and applies counterfactual interventions (do-calculus) to intermediate reasoning steps (inverting logic, negating premises, reversing factual claims) to measure whether the reasoning trace causally determines the final answer.

P5 (Reasoning Grounding): ✓. Interventions that negate factual premises in the reasoning trace test whether reasoning is grounded in stated observations. When an agent’s factual claim is replaced with its negation and the final answer remains unchanged, this reveals that the reasoning is not grounded in the evidence cited. The agent arrives at its conclusion through latent parametric priors rather than through the observations in its chain-of-thought.

P6 (Reasoning Compliance): ✓. This is the framework’s core contribution. Ariadne directly tests whether the chain-of-thought is a *causal driver* of the agent’s terminal action or *ungrounded reasoning* decoupled from the actual decision process. The Causal Decoupling phenomenon occurs when output similarity remains high despite interventions on intermediate reasoning. This constitutes a P6 violation: the reasoning trace is unfaithful to the actual decision process. This provides a stronger form of P6 evidence than textual coherence alone, because it uses *interventionist* methodology rather than surface-level consistency checks.

Scope caveat (S). Ariadne operates on single-step, single-turn prompts: one question → one chain-of-thought → one answer. Its causal interventions therefore audit whether *a single reasoning step* is faithful, which is a valid P5/P6 check at the step level. However, the framework does not extend to multi-step trajectories where reasoning errors may compound, cancel, or propagate across actions. Whether causal decoupling measured at the step level predicts trajectory-level reasoning failures remains an open question.

Excluded pairs. Ariadne does not evaluate data grounding in communication (P1/P2), policy compliance (P3/P4), instruction adherence (P7/P8), or user fulfillment (P9). No quality dimensions are scored.

F.8 CoT-Control

CoT-Control (Chen et al., 2025a) measures whether reasoning models can deliberately control the content of their chain-of-thought traces. It spans 14,076 instances with instructions that ask models to suppress keywords, add content, or apply stylistic modifications to their reasoning while solving problems.

P6 (Reasoning Compliance): ~. CoT-Control tests whether P6 auditing is *feasible* (whether models can fake their reasoning traces) rather than performing a P6 audit itself. The central finding that models exhibit dramatically lower CoT controllability than output controllability (e.g., Claude Sonnet 4.5: 2.7% CoT vs. 61.9% output) provides evidence that the reasoning trace is tightly coupled to the actual computation, making ϕ_t auditable. CoT controllability decreases with more RL training, higher reasoning effort, and increased task difficulty, suggesting the ϕ - $\mathcal{A}$ coupling strengthens where it matters most. We assign ~ because CoT-Control validates the *precondition* for reasoning compliance auditing (can models fake their CoT?) rather than directly auditing whether ϕ_t causally drove $\mathcal{A}_t$ in any specific case.

Scope caveat (S). CoT-Control evaluates single-turn controllability; it does not test whether reasoning fidelity degrades across multi-step trajectories.

Excluded pairs. CoT-Control does not model tool outputs (P1/P2), domain policy (P3/P4), agent instructions (P7/P8), or user intent (P9). No quality dimensions are addressed.

F.9 SeekBench

SeekBench (Shao et al., 2025) evaluates the episodic competence of LLM search agents through step-level trace analysis, measuring reasoning quality, evidence recovery, and answer-decision calibration.

P5 (Reasoning Grounding): ✓. The Reasoning Quality Index (RQI) checks whether each reasoning step is grounded in retrieved evidence, cross-checking observations against reasoning; a direct $O_t \rightarrow \phi_t$ audit.

Q_2 (Resilience): ✓. The Evidence Recovery Function (ERF) measures whether agents adaptively reformulate queries after insufficient search results; a direct resilience metric testing whether the agent can recover productively from retrieval setbacks rather than stalling or repeating the same query.

Excluded pairs. SeekBench checks reasoning grounding in evidence (P5), not whether the agent’s *communicated output* faithfully reflects tool returns (P1). The evaluation does not compare internal reasoning against communicated output (P6). No policy documents (P3/P4), deployment instructions (P7/P8), or user fulfillment (P9) are modeled. Q_1 and Q_3 are not addressed.

F.10 ToolEmu

ToolEmu (Ruan et al., 2024) uses an LM-emulated tool-execution sandbox and an LM-based safety evaluator to assess whether agents take unsafe actions when user instructions are underspecified (e.g., ambiguous recipients, missing confirmation steps).

P8 (Instruction Compliance): ✓. The safety evaluator inspects the agent’s action trajectory across 144 test cases and 36 toolkits, scoring whether tool calls violate implicit safety constraints of the deployment context. Failures are defined by risky *actions* (destructive commands, unauthorized payments, unverified data sharing) rather than by what the agent communicates to the user. This is a direct audit of whether agent behavior complies with safe-execution norms inherent in its deployment instructions.

Excluded pairs. ToolEmu does not audit communication fidelity (P1/P7), reasoning traces (P5/P6), contextual policy documents (P3/P4), or user satisfaction (P9). No quality dimensions are scored.

F.11 AgentDojo

AgentDojo (Debenedetti et al., 2024) is a dynamic framework for evaluating prompt injection attacks and defenses across 97 tasks and 629 security test cases (email, banking, travel, calendar).

P8 (Instruction Compliance): ◇. The *security* metric verifies whether agent actions were hijacked by injected prompts embedded in tool-returned data (e.g., data exfiltration, unauthorized messages) by inspecting environment state changes. This constitutes an adversarial-resilience test: agents should act according to the user’s legitimate task, not the attacker’s injected goal.

Excluded pairs. AgentDojo does not evaluate communication content (P1/P7), reasoning traces (P5/P6), contextual policy documents (P3/P4), or user satisfaction (P9). No quality dimensions are scored.

F.12 AgentIF

AgentIF (Qi et al., 2025) benchmarks instruction following in realistic agentic scenarios with 707 human-annotated instructions from 50 real-world applications, averaging 1,723 words and 11.9 constraints per instruction.

P7 (Instruction Grounding): ✓. *Formatting* and *semantic* constraints directly audit whether the agent’s output/communication matches deployment-instruction specifications (e.g., response structure, style, content requirements). Each constraint is evaluated via code-based or LLM-based verification.

P8 (Instruction Compliance): ✓. *Tool* constraints verify whether the agent’s function-calling behavior adheres to tool-specification protocols in the instructions (e.g., required parameters, call sequencing).

Scope caveat (S). AgentIF evaluates single-turn, single-step instruction following. The constraint taxonomy is valid at the instance level, but does not test whether compliance degrades across multi-step trajectories or multi-turn interactions.

Excluded pairs. AgentIF does not model tool-output grounding (P1/P2), reasoning traces (P5/P6), contextual policy (P3/P4), or user satisfaction (P9). No quality dimensions are scored.

F.13 AgentDyn

AgentDyn (Li et al., 2026) extends prompt-injection evaluation with 60 open-ended tasks and 560 injection test cases across Shopping, GitHub, and Daily Life domains, introducing dynamic re-planning and helpful-vs-malicious instruction discrimination.

P8 (Instruction Compliance): ✖. The attack success rate (ASR) measures whether agent actions are hijacked to serve attacker goals under adversarial conditions. The dynamic-replanning and helpful-instruction aspects test whether agents can behaviorally distinguish legitimate from malicious action directives.

Excluded pairs. AgentDyn does not evaluate communication fidelity (P1/P7), reasoning traces (P5/P6), contextual policy (P3/P4), or user satisfaction (P9). No quality dimensions are scored.

F.14 Agent-SafetyBench

Agent-SafetyBench (Zhang et al., 2024) provides 349 interaction environments and 2,000 test cases across 8 risk categories and 10 failure modes, scored by a fine-tuned safety evaluation model.

P7 (Instruction Grounding): ~. Failure mode 1 (generating harmful content without tool calls) is a communication-level safety failure; the agent’s output violates deployment-level safety norms. However, this is only 1 of 10 modes and concerns content safety rather than systematic instruction grounding, warranting a partial mark.

P8 (Instruction Compliance): ~. Failure modes 2–10 concern action-level failures: wrong tool parameters, calling tools with incomplete information, ignoring constraints, trusting unverified tool outputs. These test whether agent behavior complies with safety norms, but the benchmark frames evaluation around safety outcomes rather than directly auditing instruction-text fidelity, warranting a partial mark.

Excluded pairs. Agent-SafetyBench does not audit tool-output grounding (P1/P2), reasoning traces (P5/P6), contextual policy documents (P3/P4), or user satisfaction (P9). No quality dimensions are scored.

F.15 CIBER

CIBER (Ba et al., 2026) is an automated security benchmark for code interpreter agents, testing 4

adversarial attack types (direct/indirect prompt injection, memory poisoning, prompt-based backdoor) in dockerized sandboxes with real system privileges.

P8 (Instruction Compliance): ✖. The state-aware evaluation captures actual behavioral consequences (file exfiltration, process bombing, resource exhaustion) measuring whether agent code-execution actions remain compliant with legitimate deployment instructions under adversarial pressure.

Excluded pairs. CIBER does not evaluate communication content (P1/P7), reasoning traces (P5/P6), contextual policy (P3/P4), or user satisfaction (P9). No quality dimensions are scored.

F.16 ST-WebAgentBench

ST-WebAgentBench (Levy et al., 2024) pairs 375 web tasks with 3,057 policy instances across 6 safety-and-trustworthiness dimensions, using a hierarchical policy structure (organization > user > task).

P4 (Policy Compliance): ✓. The CuP (Completion-under-Policy) metric credits only task completions that respect all applicable policies (e.g., GitLab protected-branch rules, GDPR export checks). Policies are contextual documents; agent web actions are explicitly audited for compliance.

Excluded pairs. ST-WebAgentBench does not audit communication fidelity (P1/P7), reasoning traces (P5/P6), policy communication (P3), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.17 SOPBench

SOPBench (Li et al., 2025) provides 903 test cases across 7 customer-service domains with SOPs expressed as natural-language descriptions and oracle Python code implementations as rule-based verifiers.

P4 (Policy Compliance): ✓. Evaluation checks database outcome correctness, permissibility of each tool call, and completeness of verification procedures; all against the SOP specification. SOPs serve as contextual policy documents; agent tool calls are explicitly checked for compliance.

Excluded pairs. SOPBench does not audit communication (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.18 SOP-Bench (Nandi)

SOP-Bench (Nandi et al., 2026) contains 2,000+ tasks from human-expert-authored SOPs across 12 business domains, evaluating via execution correctness rate (ECR) and constraint-aware task success rate (C-TSR).

P4 (Policy Compliance): ✓. Agent tool/API interactions are graded against ground-truth outputs derived from expert-authored SOPs. The constraint-aware success rate (C-TSR) explicitly penalizes constraint violations during task execution.

Excluded pairs. SOP-Bench does not audit communication (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.19 GuideBench

GuideBench (Diao et al., 2025) provides 1,272 instances across 7 categories where LLMs must produce correct answers based on domain-oriented guidelines that may conflict with commonsense.

P3 (Policy Grounding): ✓. Guidelines are contextual policy documents; the evaluated output is the LLM’s textual response. The benchmark directly tests whether communication faithfully reflects the guidelines, including robustness to rule updates and alignment with human preference.

Scope caveat (S). GuideBench evaluates single-turn, single-step guideline following. It does not test whether guideline adherence degrades across multi-step workflows.

Excluded pairs. GuideBench does not audit tool-output grounding (P1/P2), reasoning traces (P5/P6), behavioral compliance (P4/P8), or user satisfaction (P9). No quality dimensions are scored.

F.20 CoPriva

CoPriva (Chang et al., 2025) evaluates LLM adherence to contextual non-disclosure security policies in QA over meeting transcripts, with 4,184 instances including direct and indirect information-elicitation attacks.

P3 (Policy Grounding): ✓. Security policies specify what information may or may not be disclosed; the benchmark tests whether the agent’s textual answer withholds policy-forbidden information; a direct $O^{\text{ctx}}-A^C$ audit.

Scope caveat (S). CoPriva evaluates single-turn responses. Indirect attacks probe adversarial resilience but within single exchanges rather than across multi-turn sessions where information may leak gradually.

Excluded pairs. CoPriva does not audit tool-output grounding (P1/P2), reasoning traces (P5/P6), behavioral compliance (P4/P8), instruction adherence (P7), or user satisfaction (P9). No quality dimensions are scored.

F.21 Company Policy Adherence

Company Policy Adherence (Zwerdling et al., 2025) evaluates whether web-agent trajectories comply with organizational policies, training a PolicyGuard guardrail model on a 60k-scale benchmark for detecting policy violations at the full-trajectory and prefix levels.

P4 (Policy Compliance): ✓. Organizational policies constrain agent action trajectories, and the system explicitly evaluates compliance through trajectory-level and prefix-based violation detection.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.22 Policy Red-Teaming

Policy Red-Teaming (Nakash et al., 2025) introduces the CRAFT multi-agent red-teaming system and the τ -break benchmark extending τ -bench, where adversarial users strategically try to manipulate policy-adherent customer-service agents into violating refund/cancellation policies.

P4 (Policy Compliance): ✧. Adversarial users attempt to induce policy violations through strategic manipulation. The attack success rate measures whether agents’ actions deviate from policy under pressure; an adversarial-resilience test of policy compliance.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.23 AgentHarm

AgentHarm (Andriushchenko et al., 2025) tests 110 explicitly malicious agent tasks across 11 harm categories, measuring whether LLM agents refuse

harmful requests and whether jailbreaks enable completion of multi-step malicious tasks.

P8 (Instruction Compliance): $\diamond$. Jailbreak attacks attempt to override system-prompt safety instructions. The refusal rate measures whether agents maintain behavioral compliance with safety instructions under adversarial pressure.

Why P4 is not checked. Although AgentHarm’s harm categories (fraud, cybercrime, harassment) loosely correspond to platform safety policies, these norms are not provided to the agent as explicit contextual policy documents (O^{ctx}). They are implicit in the model’s alignment training and system-prompt instructions, making them part of Λ (deployment instructions) rather than O^{ctx} (contextual policy). If implicit training-time safety norms counted as O^{ctx} , every safety benchmark would receive a P4 mark, diluting the distinction between policy compliance (P4) and instruction compliance (P8). The P8 $\diamond$ mark already captures what AgentHarm measures.

Excluded pairs. Does not audit communication fidelity (P1/P7), reasoning traces (P5/P6), or user satisfaction (P9). No quality dimensions are scored.

F.24 ASB

ASB (Zhang et al., 2025) is a comprehensive security benchmark spanning 10 scenarios, 10 agents, 400+ tools, and 27 attack/defense methods, testing prompt injection (direct/indirect), memory poisoning, and backdoor attacks.

P8 (Instruction Compliance): $\diamond$. Attacks attempt to override system-prompt instructions; the refuse rate measures whether agents reject unsafe requests per their deployment instructions under multi-vector adversarial pressure.

P4 (Policy Compliance): $\diamond$. Domain scenarios (e-commerce, finance, autonomous driving) involve contextual operational policies. The ASR measures whether agent actions violate these domain-specific policies under attack.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), or user satisfaction (P9). No quality dimensions are scored.

F.25 FinVault

FinVault (Yang et al., 2026) provides 31 regulatory-case-driven sandbox scenarios with state-writable databases, 107 real-world vulnerabilities, and 963 test cases, testing prompt injection, jailbreaking, and financially-adapted attacks with execution-grounded verification.

P8 (Instruction Compliance): $\diamond$. Jailbreak and prompt-injection attacks test whether agents deviate from safety instructions under adversarial pressure.

P4 (Policy Compliance): $\diamond$. Financial regulatory compliance constraints serve as explicit contextual policies; the benchmark verifies whether agents’ state-changing actions violate these constraints under attack via actual database-state inspection; a strong execution-grounded form of adversarial compliance testing.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), or user satisfaction (P9). No quality dimensions are scored.

F.26 TransLaTeX

TransLaTeX (Lyn and Graham, 2025) evaluates LLM agents on scientific document format conversion, using SafeFormatBench (100 documents) to measure compilation rate, placeholder integrity, ACL policy compliance, and visual fidelity.

P4 (Policy Compliance): $\sim$. The aclpubcheck evaluation verifies adherence to ACL formatting rules (margins, fonts, references, section spacing). These are a narrow, stylistic form of contextual policy compliance rather than behavioral/procedural policy in the typical agent sense, warranting a partial mark.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.27 MANTRA

MANTRA (Anand et al., 2026) automatically synthesizes compliance benchmarks from natural-language procedural manuals and tool schemas, using an SMT solver to validate consistency of a symbolic world model and trace-level compliance checks across 285 tasks in 6 domains.

P4 (Policy Compliance): ✓. Procedural manuals serve as contextual policy documents; agent tool-call traces are explicitly checked via machine-verifiable compliance constraints derived from the manuals. The SMT-backed deterministic grading provides formal compliance guarantees.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.28 CCTU

CCTU (Ye et al., 2026b) provides 200 test cases across 12 constraint categories in 4 dimensions (resource, behavior, toolset, response), with an executable constraint validation module performing step-level verification during multi-turn tool-use interactions.

P4 (Policy Compliance): ✓. The constraint categories (latency limits, tool-access frequency restrictions, resource budgets) function as contextual operational policies. The executable validation module performs explicit step-level compliance checking of agent actions against these constraints.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.29 AudAgent

AudAgent (Zheng et al., 2026) monitors AI agents at runtime against their stated privacy policies, formalizing policies into ontology models via cross-LLM voting and using automata-based compliance checking to detect violations.

P4 (Policy Compliance): ✓. Privacy policies are contextual policy documents; runtime data practices (tool calls, data disclosures) are continuously audited for compliance via automata-based checking. This is an explicit, real-time policy compliance audit.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.30 Near-Miss

Near-Miss (Rabinovich et al., 2026) introduces a “latent failure” metric for detecting cases where

agents reach correct final states but bypass required policy checks, using ToolGuard to convert natural-language policies into executable guard code.

P4 (Policy Compliance): ✓. Policies are formalized into executable guards; agent tool-call trajectories are checked against those guards. This goes beyond outcome-matching to audit whether the *process* complied with policy; detecting “near-miss” successes where the correct outcome was reached through non-compliant means.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.31 Solver-Aided

Solver-Aided Compliance (Winston et al., 2026) translates natural-language tool-use policies into SMT-LIB 2.0 formal constraints, intercepting planned tool calls and verifying them against a Z3 solver before execution.

P4 (Policy Compliance): ✓. Policies are formalized into SMT constraints; agent tool calls are verified with formal guarantees. Violating calls are blocked at runtime. This is the strongest form of explicit compliance auditing in the coded corpus.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.32 SOP-Maze

SOP-Maze (Wang et al., 2026) benchmarks 397 instances across 23 business SOP scenarios with complex nested conditional branches, evaluating LLM ability to follow SOPs via reference-answer matching.

P4 (Policy Compliance): ~. SOPs function as contextual policy documents, and model outputs are compared to SOP-derived reference answers. The partial mark reflects that evaluation is outcome-based (reference matching) rather than a step-by-step compliance audit, and SOPs are delivered inline in the prompt.

Scope caveat (S). SOP-Maze evaluates single-turn responses to SOP-guided queries; it does not test multi-step workflow compliance.

Excluded pairs. Does not audit communication fidelity (P1/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.33 Beyond IVR

Beyond IVR (Balaji et al., 2026) introduces JourneyBench, which models business workflows as directed graphs with nodes representing SOP steps, evaluating agents via Tool Trace Alignment and Tool Call Accuracy.

P4 (Policy Compliance): $\sim$. The Tool Trace Alignment metric compares predicted vs. expected tool-call sequences step-by-step, which superficially resembles compliance checking. However, trace alignment is *golden-path matching*, not *policy compliance auditing*: the agent’s actions are compared against a single pre-computed reference trajectory rather than against the policy constraints themselves. These are fundamentally different; a true compliance audit checks whether each action satisfies the constraints in the policy document, admitting multiple valid action sequences, whereas golden-path matching treats one particular interpretation of the SOP as the sole correct path. An agent that finds a valid alternative route through the SOP is penalized, and a reference trajectory that was incorrectly derived from the SOP (due to ambiguity or annotation error) would silently reward agents that reproduce the mistake. The partial mark further reflects that SOPs are embedded in the agent’s prompt context rather than existing as separate auditable policy documents.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.34 FM SO.P

FM SO.P (Huang et al., 2026) is a progressive training framework for SOP understanding across 7 domains, with three-stage training (concept disambiguation, action sequence understanding, scenario-aware graph reasoning) and a multi-agent evaluation system.

P4 (Policy Compliance): $\sim$. The benchmark assesses SOP *comprehension* (whether the model can reason about SOPs) rather than directly auditing whether an agent’s actions comply with policy in

a live workflow. This is an indirect measure of compliance capability, warranting a partial mark.

Scope caveat (S). Evaluation uses single-turn QA on SOP-related questions rather than multi-step task execution.

Excluded pairs. Does not audit communication fidelity (P1/P3/P7), reasoning traces (P5/P6), instruction compliance (P8), or user satisfaction (P9). No quality dimensions are scored.

F.35 τ -bench

τ -bench (Yao et al., 2025) evaluates customer-service agents in airline and retail domains through multi-turn interactions with simulated users, where agents must resolve issues using API tools while following domain-specific policy documents delivered in the system prompt.

P4 (Policy Compliance): $\sim$. The agent must follow domain policies (e.g., “basic economy cannot be modified”) when performing write actions. The reward r_{action} checks whether the final DB state matches ground truth; indirectly auditing whether actions complied with policy constraints. The partial mark reflects that compliance is inferred from outcome correctness rather than via an explicit per-step policy-violation detector.

P9 (User Fulfillment): $\checkmark$. The reward r_{output} explicitly checks whether the agent’s responses contain all required information for the user; the overall reward directly measures whether the user’s intent was resolved correctly (DB outcome matches user’s desired state).

Excluded pairs. No explicit audit of communication–action consistency (P1/P2), reasoning traces (P5/P6), or deployment instructions (P7/P8). No quality dimensions are scored.

F.36 τ^2 -bench

τ^2 -bench (Barres et al., 2025) extends τ -bench to a dual-control Dec-POMDP where both agent and user have tools, adding a telecom domain and categorized evaluation (DB checks, status assertions, NL assertions, communication info checks, action matching).

P4 (Policy Compliance): $\sim$. Same rationale as τ -bench: domain policy constrains agent actions, and task success implicitly requires compliance.

The paper ablates policy removal (showing performance impact), but still provides no explicit per-step violation scoring.

P9 (User Fulfillment): ✓. Task success requires that the user’s issue is resolved (verified by DB state and communication info checks). The categorized evaluation explicitly verifies whether the agent conveyed correct information.

Excluded pairs. No explicit audit of P1/P2, P5/P6, P7/P8, or quality dimensions.

F.37 USR

USR (Mehri and Eskenazi, 2020) is a reference-free dialogue evaluation metric with unsupervised sub-metrics: Understandable, Natural, Maintains Context, Interesting, Uses Knowledge, and Overall Quality, trained on masked-language-model and dialogue-retrieval tasks.

Q₃ (Interaction Quality): ✓. USR is explicitly a multi-dimensional dialogue quality metric measuring naturalness, coherence, interestingness, and context maintenance; precisely interaction quality dimensions.

Scope caveat (S). USR evaluates single-turn response quality; it does not test whether interaction quality degrades across multi-turn or multi-step sessions.

Excluded pairs. USR is a *metric*, not a benchmark with task completion. It measures no procedural compliance, policy adherence, data grounding, or user fulfillment. No other quality dimensions are scored.

F.38 Userbench

Userbench (Qian et al., 2025) is a user-centric benchmark for multi-turn travel planning where simulated users have underspecified, incremental, and indirect preferences that the agent must elicit and satisfy.

P9 (User Fulfillment): ✓. The primary metric is a normalized reward score measuring alignment of the agent’s final recommendation with the user’s true (hidden) preferences. Preference Elicited % directly measures how well the agent uncovers user intent.

Excluded pairs. No audit of P1–P8 or quality dimensions Q1–Q3.

F.39 INSURE-Dial

INSURE-Dial (Kulkarni et al., 2026) is a phase-aware benchmark for insurance benefit verification calls, evaluating Information Compliance (was required information explicitly stated?) and Procedural Compliance (was the interaction carried out in the correct phase order?).

P9 (User Fulfillment): ~. The benchmark’s core contribution is procedural and informational compliance verification against an audit checklist; checking whether each phase was executed correctly and in order. We assign P9~ because successful completion of the verification call ultimately serves the healthcare provider’s information needs, but the benchmark frames evaluation around audit compliance rather than user satisfaction.

Excluded pairs. INSURE-Dial does not audit tool-output grounding (P1), reasoning traces (P5/P6), deployment instructions (P7/P8), or quality dimensions. Note: the benchmark’s procedural-compliance focus borders on P2/P4, but we map conservatively because the “policy” is an implicit call structure rather than an explicit document.

F.40 HotelQuEST

HotelQuEST (Hadam et al., 2026) evaluates hotel search agents on 214 queries spanning simple to complex, jointly measuring quality (Accuracy, Factuality) and efficiency (total tokens, latency, cost).

P1 (Data Grounding): ✓. The Factuality evaluator explicitly checks whether the agent’s answer is grounded in hotel descriptions and reviews; the prompt instructs: “Evaluate ONLY using the provided hotel descriptions and reviews.” This is a direct $O^{\text{sys}}\text{--}A^C$ audit.

P9 (User Fulfillment): ✓. The Accuracy evaluator measures whether the recommended hotel matches the user’s query and preferences (including clarified underspecified ones).

Q₁ (Efficiency): ✓. The paper’s central thesis is joint quality–efficiency evaluation. It reports total tokens, latency, cost, and analyzes cost-inefficiency from redundant tool calls, with quality and efficiency oracles providing bounds.

Excluded pairs. No audit of P2, P5/P6, P7/P8, P3/P4, Q2, or Q3.

F.41 Online-Mind2Web

Online-Mind2Web (Xue et al., 2025) benchmarks 300 web tasks across 136 real websites with human judgment and WebJudge (LLM-as-a-Judge), reporting binary task success and an explicit efficiency metric.

Q₁ (Efficiency): ✓. The efficiency metric $E = \frac{1}{|T_{\text{succ}}|} \sum_{i \in T_{\text{succ}}} S_i / \hat{S}_i$ explicitly compares agent steps to human reference trajectory length, providing a dedicated per-agent efficiency score.

Excluded pairs. No audit of P1–P9, Q2, or Q3.

F.42 Mind2Web 2

Mind2Web 2 (Gou et al., 2025) benchmarks agentic search (Deep Research) systems on 130 long-horizon web tasks, using an Agent-as-a-Judge framework with tree-structured rubrics that evaluate answer correctness and source attribution.

P1 (Data Grounding): ✓. The source-attribution evaluation explicitly audits whether agent-generated statements are grounded in retrieved web pages. The `verify_by_url` verifier checks each cited source, directly measuring $O^{\text{sys}} - A^C$ alignment.

P9 (User Fulfillment): ✓. The correctness evaluation checks whether the answer satisfies all user-specified requirements (budget, item constraints, etc.) via the rubric tree. Partial Completion and Success Rate metrics directly measure task-goal fulfillment.

Excluded pairs. No audit of P2, P5–P8, P3/P4, or quality dimensions.

F.43 TRAJECT-Bench

TRAJECT-Bench (He et al., 2025a) is a trajectory-level preference benchmark for reward models in tool-using scenarios, with four task families: Safety Refusal, Tool-Irrelevance/Unavailability, Complex Planning, and Robust Error Recovery.

P1 (Data Grounding): ~. Tool-grounded fabrication (claiming tool output that was not returned) is scored as a key failure mode. The partial mark reflects that TRAJECT-Bench tests the reward model’s ability to *detect* data-grounding violations rather than directly auditing agent data grounding.

P9 (User Fulfillment): ✓. Preferred trajectories are those that correctly fulfill the user’s intent; gold labels encode this distinction.

Q₁ (Efficiency): ✓. The scoring rubric distinguishes inefficient trajectories from incorrect ones, and performance analysis separates trajectory-length effects from hard failures.

Q₂ (Resilience): ~. “Robust Error Recovery” is one of four task families, testing whether reward models can judge recovery quality. The partial mark reflects that this tests RM recognition of resilience rather than adversarial probing of agents.

Excluded pairs. No audit of P2, P5–P8, P3/P4, or Q3.

F.44 Vitabench

Vitabench (He et al., 2025b) simulates food delivery, in-store consumption, and travel services with 66 tools and 400 tasks requiring multi-turn interaction, ambiguity clarification, and shifting-intent tracking, scored by a rubric-based sliding-window evaluator.

P9 (User Fulfillment): ✓. Task success rate is the primary metric; the rubric checklist measures whether user goals are met, including handling of underspecified preferences and intent shifts.

Excluded pairs. No audit of P1–P8 or quality dimensions Q1–Q3.

F.45 COMPASS

COMPASS (Qin et al., 2026) evaluates constrained preference optimization for travel planning with hard constraints (dates, budget) and soft preferences (cheapest, wishlist) across multi-turn interactions.

P9 (User Fulfillment): ✓. Measures both hard-constraint satisfaction and soft-preference optimization; directly auditing whether the agent fulfills user requirements.

Excluded pairs. No audit of P1–P8 or Q1–Q3. Note: the paper includes a dedicated “Conversation Efficiency” analysis (§5.2) measuring convergence speed, but this is framed as a secondary finding rather than a primary benchmark axis.

F.46 TRIP-Bench

TRIP-Bench (Shen et al., 2026) evaluates long-horizon travel planning with up to 15 user turns, 150+ tool calls, and 200k+ tokens, using hierarchical rubric-to-constraint generation and rule-based + turn-level metrics.

P9 (User Fulfillment): ✓. Success rate on rubric-defined constraints directly measures user goal fulfillment across extended interactions.

Excluded pairs. No audit of P1–P8 or Q1–Q3.

F.47 M3-Bench

M3-Bench (Xie et al., 2026) evaluates LLM agents in 24 mixed-motive games via three process-aware modules: Behavioral Trajectory Analysis (BTA, what agents *do*), Reasoning Process Analysis (RPA, what agents *think*), and Communication Content Analysis (CCA, what agents *say*).

P2 (Execution Compliance): ✓. The core diagnostic is the mismatch between what agents communicate/reason (via RPA/CCA) and what they actually execute (via BTA). The speech–action consistency metric and the “overthink–undercommunicate” finding directly audit whether communicated intentions align with executed actions; a direct A^C-A^W audit.

P5 (Reasoning Grounding): ∼. RPA evaluates whether internal reasoning is coherent and grounded in game state (beliefs about others, interaction history, payoff structure) via LLM-judge scoring. The partial mark reflects that grounding is assessed by judge rather than formal trace-to-state verification.

Q3 (Interaction Quality): ∼. CCA analyzes communicative strategy, pragmatic speech acts, and style distribution; touching interaction quality. The partial mark reflects that the focus is strategic social communication in games rather than general agent–user interaction quality.

Excluded pairs. No audit of P1, P6–P8, P3/P4, P9, Q1, or Q2.

F.48 Scalable Oversight

Scalable Oversight (Kenton et al., 2024) studies whether weak LLM judges can effectively oversee strong LLM outputs via debate (two AIs argue opposing sides) and consultancy (one AI argues) protocols across 9 QA tasks.

P1 (Data Grounding): ∼. In extractive QA tasks, debaters must ground arguments in source articles via a verified-quote tool; judge accuracy partially reflects whether agent claims are data-grounded. The partial mark reflects that data grounding is a mechanism enabling the oversight

protocol rather than the object of systematic evaluation.

Q2 (Resilience): ◇. The core setup is adversarial: one debater argues for the *incorrect* answer, stress-testing whether the judge can resist persuasive but wrong arguments. The finding that debate resists incorrect-answer persuasion better than consultancy is a direct adversarial resilience result.

Excluded pairs. No audit of P2, P5–P8, P3/P4, P9, Q1, or Q3.

F.49 Claw-Eval

Claw-Eval (Ye et al., 2026a) is an end-to-end evaluation suite with 300 tasks, three evidence channels (execution traces, audit logs, environment snapshots), and 2,159 rubric items scoring Completion, Safety, and Robustness.

P4 (Policy Compliance): ∼. Safety evaluation catches violations (44% missed by trajectory-opaque methods); safety constraints function as implicit policies. The partial mark reflects that constraints are embedded in rubric items rather than delivered as separate contextual policy documents.

Q2 (Resilience): ✓. Robustness is a first-class scoring axis with dedicated error injection that drops Pass³ by up to 24 pp; an explicit resilience audit.

Excluded pairs. No audit of P1–P3, P5–P8, P9, Q1, or Q3.

F.50 Multi-Faceted

Multi-Faceted (Shourya et al., 2026) introduces the TRACE benchmark (516 conversations, 26 error cases) and the SCOPE framework for auto-discovering error patterns in tool-augmented dialogues.

P1 (Data Grounding): ∼. Error categories include tool-result misinterpretation (agent says X, tool returned Y); touching $O^{\text{sys}}-A^C$. The partial mark reflects that this is incidental to the broader error taxonomy, not a dedicated grounding metric.

Q3 (Interaction Quality): ∼. The paper evaluates overall conversation quality beyond user satisfaction, but the primary contribution is error detection rather than a standalone interaction-quality metric.

Excluded pairs. No audit of P2, P5–P8, P3/P4, P9, Q1, or Q2.

F.51 Interaction Smells

Interaction Smells (Zhang et al., 2026) provides an empirical taxonomy of interaction-quality problems in multi-turn human–LLM code generation, categorized into User Intent Quality, Historical Instruction Compliance, and Historical Response Violation (9 subcategories).

Q₃ (Interaction Quality): $\sim$. The taxonomy directly characterizes interaction-quality degradation patterns. The partial mark reflects that this is an empirical classification and mitigation framework rather than a dedicated scored quality metric applied per-agent.

Excluded pairs. “Historical Instruction Compliance” refers to user-issued instructions within the conversation, not deployment instructions (Λ); not P8. No audit of P1–P9, Q1, or Q2.

F.52 Beyond State Machines

Beyond State Machines (Garigipati et al., 2026) evaluates LLM agents executing network procedures (UE IP allocation) as tool-call sequences, comparing 4 approaches across procedure lengths and error conditions.

P2 (Execution Compliance): $\sim$. Execution correctness checks whether the agent’s tool-call sequence matches the required procedure. The partial mark reflects that this compares against a reference sequence (golden-path matching) rather than verifying communicated-vs-executed consistency.

P4 (Policy Compliance): $\sim$. The network procedure functions as a domain SOP, and step-sequence correctness is checked. The partial mark reflects that the “procedure” is a task specification delivered inline rather than a separable contextual policy document.

Excluded pairs. No audit of P1, P5–P8, P3, P9, or quality dimensions.

F.53 TCQA²

TCQA² (Chen et al., 2025b) evaluates tiered conversational Q&A agents for gaming NPCs, integrating knowledge graphs, FAQ pairs, and real-time web content, with multi-dimensional quality assessment including hallucination detection, relevance, safety, and personalized dialogue quality.

Q₃ (Interaction Quality): $\checkmark$. The paper evaluates personalized dialogue, multi-turn Q&A quality, emotional companionship, and risk mitigation as dedicated, scored evaluation targets; interaction quality is a primary evaluation axis.

Excluded pairs. No audit of P1–P9, Q1, or Q2.

F.54 WikiHowAgent

WikiHowAgent (Pei et al., 2025) evaluates a multi-LLM agent workflow for procedural learning (Teacher + Learner + Interaction Manager + Evaluator) across 114,296 conversations and 14,287 tutorials.

Q₃ (Interaction Quality): $\checkmark$. A dedicated Evaluator component assesses conversation quality using rubric-based pedagogic metrics aligned with human judgment; directly measuring interaction quality as a first-class scored dimension.

Excluded pairs. No audit of P1–P9, Q1, or Q2.

G Policy Comparison

Table 5 provides a structured comparison of all benchmarks that evaluate policy grounding (P3) or policy compliance (P4). The comparison shows large differences in what “policy” means across the literature and how compliance is measured.

G.1 Key Findings

Policy complexity spectrum. The benchmarks span a wide realism range. At the low end, Co-Priva uses short declarative non-disclosure rules and TransLaTeX checks a fixed set of formatting constraints. At the high end, MANTRA scales to 50+ page procedural manuals and τ -bench/ τ^2 -bench use multi-page prose policies with complex conditional rules requiring multi-hop reasoning (e.g., tier-specific cancellation rules that depend on cabin class, booking date, and membership status). SOP-Bench (Nandi) occupies the middle with 3–7 page expert-authored SOPs containing ambiguous language and implicit domain knowledge.

The verification–complexity tradeoff. A striking pattern emerges: benchmarks with the longest and most realistic policies tend to use the weakest verification methods, and vice versa. τ -bench and τ^2 -bench use realistic multi-page policies but rely on LLM judges for compliance scoring (outcome-level). Conversely, Solver-Aided and MANTRA achieve formal SMT-backed verification guarantees

Table 5: Comparison of policy characteristics across P3/P4 benchmarks. **Complexity:** S = short (a few rules), M = medium (1–5 pages), L = long (5–50+ pages). **Structure:** F = flat rules, H = hierarchical, G = graph/workflow, N = nested conditionals, P = prose. **Delivery:** Pr = in-prompt, Sep = separate document, FC = pre-compiled formal constraints. **Verification:** GP = golden-path matching, RB = rule-based oracle, GC = executable guard code, AM = automata, LJ = LLM judge, FM = formal methods (SMT), OB = outcome-based. **Granularity:** A = per-action, T = trajectory-level, O = outcome-only. **Multi-path:** Y = allows multiple valid paths, N = single golden path.

Paper	Policy Type	Comp.	#Rules	Struct.	Deliv.	Verif.	Gran.	Multi	Adv.
<i>P3 (Policy Grounding : communication side)</i>									
GuideBench (Diao et al., 2025)	Domain guidelines	S–M	7 categories	H,N	Pr	RB+LJ	O	Y	–
CoPriva (Chang et al., 2025)	Non-disclosure rules	S	2,763 policies	F	Pr	LJ	O	Y	✓
<i>P4✓ (Policy Compliance : explicit audit)</i>									
ST-WebAgentBench (Levy et al., 2024)	Enterprise ST policies	M	3,057 instances	H	Sep	RB	A	Y	–
SOPBench (Li et al., 2025)	Customer-svc SOPs	M	165 constraints	G	Sep	RB	A	Y	–
SOP-Bench (Nandi et al., 2026)	Industrial SOPs	M–L	2,000+ tasks	N,G	Sep	RB	A	Y	–
Co. Policy Adh. (Zwerdling et al., 2025)	Web-agent policies	S	60K pairs	F	Sep	RB	T	Y	–
MANTRA (Anand et al., 2026)	Procedural manuals	L	285 tasks	G	FC	FM	A	Y	–
CCTU (Ye et al., 2026b)	Operational constraints	M	12 categories	H	Pr	RB	A	Y	–
AudAgent (Zheng et al., 2026)	Privacy policies	M	Real policies	H	Sep	AM	A	Y	–
Near-Miss (Rabinovich et al., 2026)	Airline policies	M	~4 guards	N	Pr	GC	A	Y	–
Solver-Aided (Winston et al., 2026)	Tool-use policies	M	Airline subset	N	FC	FM	A	Y	–
<i>P4◊ (Policy Compliance : adversarial resilience)</i>									
Policy Red-Team. (Nakash et al., 2025)	Cust-svc policies	M	Airline+retail	N,P	Pr	LJ	T	Y	✓
ASB (Zhang et al., 2025)	Domain system prompts	S–M	10 scenarios	F	Pr	OB	T	N	✓
FinVault (Yang et al., 2026)	Regulatory constraints	M	107 vulns	N	FC	OB+GC	A	Y	✓
<i>P4~ (Policy Compliance : partial/indirect)</i>									
τ -bench (Yao et al., 2025)	Cust-svc domain policy	M	Airline+retail	N,P	Pr	LJ	O	Y	–
τ^2 -bench (Barres et al., 2025)	Cust-svc domain policy	M	3 domains	N,P	Pr	LJ	O	Y	–
TransLaTeX (Lyn and Graham, 2025)	Formatting rules	S	ACL checks	F	FC	RB	O	N	–
SOP-Maze (Wang et al., 2026)	Business SOPs	M	23 SOPs	N	Pr	GP	O	N	–
Beyond IVR (Balaji et al., 2026)	Workflow SOPs	M	3 domains	G	Pr	GP	T	N	–
FM SO.P (Huang et al., 2026)	SOPBench SOPs	M	7 domains	G	Pr	GP+LJ	O	N	–

but require policies to be translated into logical constraints; a process that may lose nuance from the original natural-language document. Near-Miss’s executable guard code occupies a middle ground: it formalizes specific safety-critical policy rules while leaving non-critical rules unchecked.

Golden-path matching vs. constraint-based verification. A basic method divide separates the P4~ from the P4✓ papers. SOP-Maze, Beyond IVR, and FM SO.P compare agent traces against pre-computed reference trajectories (golden-path matching). This conflates policy compliance with behavioral cloning: an agent that finds a valid alternative path through the SOP is penalized, and errors in the reference trajectory are silently propagated. In contrast, the P4✓ papers (ST-WebAgentBench, SOPBench, MANTRA, Near-Miss, Solver-Aided) verify each action against policy *constraints*, admitting any action sequence that satisfies the rules. This distinction is the primary driver of the ✓ vs. ~ distinction.

Policy delivery and the P4/P8 boundary. Many benchmarks deliver policies in-prompt (Pr), blurring the line between contextual policy (O^{ctx} , P3/P4) and deployment instructions (Λ , P7/P8). We assign P4 when the policy is clearly a domain document that varies across tasks and could be

swapped (e.g., different airline rules, different company SOPs), distinguishing it from fixed deployment instructions. CCTU is a borderline case: its constraints (resource budgets, tool-access limits) resemble deployment-level instructions but are coded as P4 because they vary per task and function as operational policies.

Adversarial testing is rare. Only 4 of 20 benchmarks test policy compliance under adversarial pressure: CoPriva (indirect information-elicitation attacks), Policy Red-Teaming (strategic user manipulation), ASB (multi-vector prompt injection), and FinVault (financially-adapted attacks). Most P4✓ benchmarks assume cooperative users, leaving open whether their agents would maintain compliance under adversarial conditions.

Communication vs. behavior. Only GuideBench and CoPriva evaluate the *communication* side of policy (P3: does the agent’s textual output reflect policy?). All other benchmarks focus on the behavioral side (P4: do the agent’s actions comply?). This gap suggests that policy *grounding* (whether agents accurately communicate policy constraints to users) remains largely unexamined, even though customer-facing agents routinely need to explain why they cannot fulfill a request.