From Failure Attribution to Reliability Enhancement in LLM-based Multi-Agent Systems: A Survey

Anonymous ACL submission

Abstract

Large Language Model-based Multi-Agent Systems (LLM-MAS) enable sophisticated collaborative reasoning, yet deployment is hindered by unreliable behaviors. Identifying the precise causes of failures, *Failure Attribution*, remains challenging due to the complexity of MAS interactions. This survey provides a unified perspective on Failure Attribution and its role in *Reliability Enhancement*. We formalize attribution as a structured 5Ws task (WHAT, WHEN, WHO, WHERE, WHY) over execution traces, summarize key challenges, and organize existing methods into four paradigms: in-context reasoning, fine-tuning critics, causal inference, and statistical analysis. We further review representative datasets and metrics, and introduce HOW as a practical lens for turning 5Ws findings into runtime intervention and post-hoc refinement. We conclude by identifying open challenges and outlining a roadmap toward more reliable MAS.

1 Introduction

Large Language Models (LLMs) are increasingly used as modular components of agentic workflows, supporting planning, tool use, code synthesis, and iterative refinement (Shinn et al., 2023; Madaan and Tandon, 2023; Jia et al., 2025). Broader surveys and software-engineering studies further show that LLM-based agents are becoming a general paradigm for building autonomous task-solving systems (Minaee et al., 2024; Ferrag et al., 2025; Vaidhyanathan and Muccini, 2025; Wang et al., 2026a). This shift has accelerated LLM-based multi-agent systems (LLM-MAS), where specialized agents collaborate through role-based communication, memory-augmented cooperation, or learned coordination policies (Jiang et al., 2026; Abdurahman et al., 2026; Nechepurenko and Shuvalov, 2026; Zhang, 2026a). By decomposing complex tasks across agents and tools, LLM-MAS can address long-horizon problems that are difficult for a single model to solve reliably (Lu et al., 2023; Li et al., 2024; Wan et al., 2025; Tran et al., 2025; Du et al., 2026; Zhu et al., 2026).

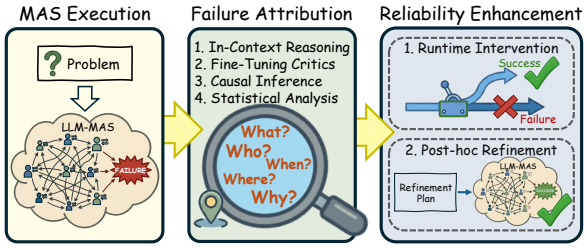

Figure 1: Overview of the survey pipeline from failure to attribution, leading to reliability enhancement.

As LLM-MAS are increasingly used for long-horizon coordination, reliability has emerged as a central obstacle to deployment (Pan et al., 2025; Deshpande et al., 2025; Zhu et al., 2025c; Pezeshkpour and Hruschka, 2026; Kirmayr et al., 2026; Roig, 2025; Lu et al., 2025; Vinay, 2025; Shah et al., 2026). Because execution involves many turns and agents, small mistakes can cascade through the workflow and cause system-level failures (Zhang et al., 2026a, 2025e; Tan et al., 2026; Liu et al., 2026b; Banerjee et al., 2025; Xie et al., 2026; Mazhar et al., 2026; Xia et al., 2026). Long-horizon traces exhibit complex dependencies, making *failure attribution* (FA), localizing and explaining the root failure cause within an execution trace (In et al., 2026; Shefin et al., 2026; Ou et al., 2025; Moshkovich and Zeltyn, 2025), a critical bottleneck for maintaining system reliability (Pan et al., 2025; Chen et al., 2026b; Li et al., 2026; Hu et al.). A growing body of work has proposed methods for FA in LLM-MAS (Zhang et al., 2025e; Kong et al., 2026; Chen et al., 2026b; Barke et al., 2026), as well as reliability enhancement strategies that leverage attribution signals (Lin et al., 2025; Xie et al., 2025).

Failure attribution in LLM-MAS is studied un-

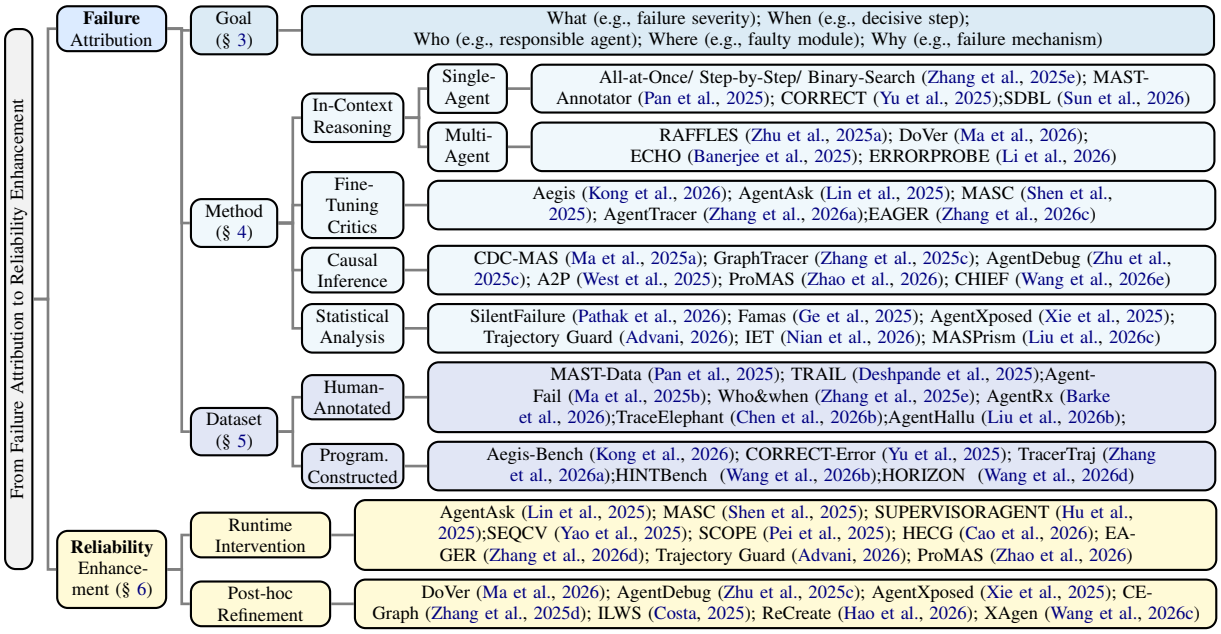

Figure 2: A comprehensive taxonomy of failure attribution and reliability enhancement in LLM-MAS.

der heterogeneous problem formulations (In et al., 2026), varying in attribution targets, granularity, and causal criteria, which makes results difficult to interpret and compare across works. It remains unclear how attribution paradigms relate to each other, what datasets and metrics capture across settings, and how diagnostic outputs can be translated into reliability improvements, motivating our taxonomy and organized review of both FA and attribution-driven reliability enhancement. To address this gap, as illustrated in Figure 1, we present a review of failure attribution in LLM-MAS and the reliability enhancement methods that build on attribution signals. Our main contributions are as follows. (i) **Unified Problem View(5Ws + H):** We conceptualize the lifecycle of failure in LLM-MAS by formalizing *Failure Attribution* as a 5Ws' task (WHAT, WHEN, WHO, WHERE, WHY) and introducing HOW as a complementary dimension that operationalizes diagnoses into reliability-improving actions. (ii) **Systematic Taxonomy of Attribution Methodologies:** We propose a taxonomy that organizes attribution methods into four paradigms: *In-Context Reasoning, Fine-Tuning Critics, Causal Inference* and *Statistical Analysis*. Then, we review the evaluation ecosystem, scrutinizing datasets and metrics. (iii) **Failure Attribution-Driven Reliability Enhancement:** We investigate how diagnostic signals catalyze system reliability. We categorize enhancement methods into *runtime intervention* and *post-hoc refinement* based on their temporal application, highlighting their roles in active error mitigation and long-term system evolution. (iv) **Challenges and Future Directions:** We analyze gaps in current methods and propose a roadmap for research, aiming to build more reliable LLM-MAS.

2 Notations

We model an LLM-MAS as a turn-based system $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, \phi \rangle$ (Li et al., 2023; Zhang et al., 2025e), where $\mathcal{N} = \{1, \ldots, N\}$ is the agent set, $\mathcal{S}$ the observable state space, $\mathcal{A}$ the action space, P the transition kernel, and $\phi(t)$ the active agent at step t. Each agent i acts in $\mathcal{A}_i \subseteq \mathcal{A}$ and emits $a_t \in \mathcal{A}_{\phi(t)}$; the system evolves by $P(s_{t+1} \mid s_t, a_t, \phi(t))$. An execution trace is $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$, where s_t includes dialogue, memory, retrieved evidence, tool observations, and intermediate artifacts. We define failure attribution over such traces.

3 A Unified FA as 5Ws Task

Inspired by 5W-style structuring schemes widely used to organize explanations and event semantics in related domains (Zhang et al., 2013; Chen and Li, 2020; Chen et al., 2026a), and consistent with emerging LLM-MAS failure attribution studies (Ge et al., 2025; Zhang et al., 2025e; Kong et al., 2026), we unify failure attribution as a structured 5Ws task, as shown in Figure 2. Formally, let $\mathcal{T}$ be the space of all possible execution traces

and $\mathcal{D}$ be the space of structured diagnoses. We define the **Failure Attribution** task as learning a mapping function $\mathcal{F} : \mathcal{T} \rightarrow \mathcal{D}$. Given a trace $\tau \in \mathcal{T}$, the attribution function produces a structured attribution tuple:

$$\mathcal{F}(\tau) = \big(D_{\text{WHAT}}, D_{\text{WHEN}}, D_{\text{WHO}}, \quad D_{\text{WHERE}}, D_{\text{WHY}}\big). \tag{1}$$

Each component of this tuple addresses a dimension of the failure. (i) **What** (D_{WHAT}) characterizes the outcome of τ (Lin et al., 2025; Merrill et al., 2026; Kirmayr et al., 2026). A form is a scalar score $z(\tau) \in \mathbb{R}_{\geq 0}$ measuring failure severity (i.e., distance to completion), where $z(\tau) = 0$ indicates success and larger values indicate more severe failures. (ii) **When** (D_{WHEN}) localizes the error step t^* in the trace, operationalized as the earliest *decisive error step* (Zhang et al., 2025e; Kong et al., 2026; Islam et al., 2026; In et al., 2026). Given a counterfactual trace τ_t' obtained by replacing a_t with a corrected action and re-executing subsequent steps, step t is decisive if $z(\tau_t') = 0$, and the attribution target is $t^* = \min\{t \mid z(\tau_t') = 0\}$. (iii) **Who** ($D_{\text{WHO}}$) identifies the responsible agent(s) in $\mathcal{N}$. In LLM-MAS attribution settings (Zhang et al., 2025e, 2026a; Chen et al., 2026b), WHO is instantiated as the agent that executes the decisive error step t^*. (iv) **Where** (D_{WHERE}) identifies the faulty system module (Pan et al., 2025; Zhu et al., 2025c; Banerjee et al., 2025; Zhang et al., 2026e) (e.g., planning, reflection, memory, or tool/API interface). (v) **Why** (D_{WHY}) describes the *failure mechanism* (Ma et al., 2025a; Wang et al., 2026d,g; Russo, 2026; Fan et al., 2026; Shah et al., 2026; Vinay, 2025), what went wrong and how it led to outcome, supported by evidence in τ (e.g., a fabricated fact caused a decision, or a tool argument produced a misleading observation).

4 A Taxonomy of FA Methods

The workflows are illustrated in Figure 3, and characteristics are summarized in Table 1. After outlining the intrinsic difficulties of failure attribution in LLM-MAS, we organize prior work into four paradigms: (i) *In-Context Reasoning*, where an LLM-as-a-Judge (You et al., 2026) is prompted to evaluate an extracted trace without parameter updates; (ii) *Fine-Tuning Critics*, where a critic is trained on traces (optionally with labels), e.g., via supervised fine-tuning (SFT) or RL, to perform failure attribution; (iii) *Causal Inference*, where a causal dependency structure is built from traces and analyzed to identify failure; (iv) *Statistical Analysis*, where trace-derived features are scored statistically to indicate failure source.

4.1 Intrinsic Difficulties in Attribution

Cascading Error Propagation. A single early deviation (e.g., a subtle hallucination, contaminated message, or underspecified instruction) can be reused and amplified by downstream agents, turning a local defect into a system-level collapse (Zhang et al., 2025e; Xie et al., 2026; Mazhar et al., 2026; Li et al., 2025). This yields a pronounced root, symptom gap: the trigger is often far earlier than visible failure and obscured by intermediate steps (Wang et al., 2025f; Li et al., 2026).

Complex Causal Dependencies. Although traces are logged sequentially, decision logic depends on long-range semantics: actions may draw on evidence from earlier turns, cross-agent messages, or retrieved memory rather than immediately preceding step (Zhang et al., 2025c; Ma et al., 2025a; Wang et al., 2026e). This misalignment can distort propagation paths, making chronology a poor proxy for causality and obscuring which prior information truly shaped failure (Mao et al., 2026; Zhou et al., 2025).

Contextual Overload in Execution Traces. FA requires inspecting various signals (messages, tools, memory, and code), while decisive evidence is often sparse and dispersed across thousands of tokens (Zhu et al., 2025a; Wang et al., 2025b; Sun et al., 2026; In et al., 2026). Under bounded context, evaluators must select and justify evidence, making reliable localization and mechanism explanation brittle over long horizons(Merrill et al., 2026; Wang et al., 2026f; Zhan et al., 2026).

4.2 In-Context Reasoning

4.2.1 Single-Agent Diagnosis

Single-agent methods use an LLM-as-a-Judge to diagnose logs with three trace-processing strategies. Global processing feeds the full log once (Zhou et al., 2024): All-at-Once (Zhang et al., 2025e) is efficient but prone to "lost-in-the-middle" (Liu et al., 2024), while MAST-Annotator (Pan et al., 2025) uses few-shot structured prompts (Khattab et al., 2024; Yuksekgonul et al., 2025) for taxonomy-based labeling. Trajectory scanning trades cost for granularity: Step-by-Step uses $O(T)$ checks, Binary Search uses

Method	Cat.	Target	Key Mechanism	Dataset	Metric	L.
MAST-Annotator (Pan et al., 2025)	IC	WT,WR	Few-shot judge with failure taxonomy	MAST-Data	Mod-Acc/P/R/F1; κ	
All/Step/Binary (Zhang et al., 2025e)	IC	WT,WN,WO,WY	Full-log, sequential, and bisection attribution	Who&When	Agent-/Step-Acc; @n	
CORRECT (Yu et al., 2025)	IC	WT,WN	Distills recurrent errors into reusable schema templates	Who&When; CORRECT-Error	Step-Acc; @n	
SDBL (Sun et al., 2026)	IC	WT,WN,WO,WY	Two-stage scope localization	Who&When	Step-Acc; @n; Tok	
RAFFLES (Zhu et al., 2025a)	IC	WT,WN,WO	Collaborative agents with verification	Who&When	Step-Acc; @n	
DoVer (Ma et al., 2026)	IC	5Ws	Replay-validated intervention hypotheses	Who&When	Agent-/Step-Acc	
ECHO (Banerjee et al., 2025)	IC	WT,WN,WO,WR	Hierarchical consensus diagnosis	Who&When	Agent-/Step-Acc; @n; Tok	
ERRORPROBE (Li et al., 2026)	IC	5Ws	Verified-memory backward tracing	Who&When; TracerTraj	Agent-/Step-Acc	
Aegis (Kong et al., 2026)	FT	WT,WN,WO,WR	Fault-injected critic training via SFT/GRPO/DCL	Who&When; Aegis-Bench	Micro/Macro P/R/F1	
AgenTracer (Zhang et al., 2026a)	FT	WT,WN,WO	Counterfactual replay for trace critics	Who&When; TracerTraj	Agent-/Step-Acc	
AgentAsk (Lin et al., 2025)	FT	5Ws	Clarification policy at handoffs	–	–	
MASC (Shen et al., 2025)	FT	WT,WN,WO,WY	Next-step embedding anomaly scoring	Who&When	Agent-/Step-Acc	
EAGER (Zhang et al., 2026c)	FT	WT,WN,WO,WR	Reasoning-scope contrastive memory	AutoG/RCL/SWE	F1; Lat	
CDC-MAS (Ma et al., 2025a)	CI	WT,WN,WO,WY	Causal graph ranks agents and steps	Who&When; TRAIL	Agent-/Step-Acc	
GraphTracer (Zhang et al., 2025c)	CI	WT,WN,WO,WY	Information-dependency tracing	Who&When; TracerTraj	Agent-/Step-Acc	
AgentDebug (Zhu et al., 2025c)	CI	5Ws	Taxonomy-guided root-cause feedback	AgentErrorBench	Agent-/Step-Acc	
A2P (West et al., 2025)	CI	WT,WN,WO,WY	Abduct–Act–Predict intervention tests	Who&When	Step/Module-Acc	
ProMAS (Zhao et al., 2026)	CI	WT,WN,WO	Markov transitions for error forecasting	Who&When	Agent-/Step-Acc	
CHIEF (Wang et al., 2026e)	CI	WT,WN,WO,WY	Hierarchical causal graph backtracking	Who&When	Agent-/Step-Acc; Tok	
SilentFailure (Pathak et al., 2026)	SA	WT,WR	Trace fingerprints for silent anomalies	Stock; Writing	Acc; MacroF1; P/R	
Famas (Ge et al., 2025)	SA	WT,WN,WO	SBFL over replayed trajectories	Who&When	Agent-/Step-Acc	
AgentXposed (Xie et al., 2025)	SA	5Ws	Behavioral anomaly scoring	–	–	
Trajectory Guard (Advani, 2026)	SA	WT,WY	Hybrid loss for task–trajectory alignment	RAS; Who&When	P/R/F1; Lat	
IET (Nian et al., 2026)	SA	WT,WN,WO,WR	Keyed implicit execution tracing	MAMA; Who&When	TokenAcc; IoU; EdgeSim; Agent-/Step-Acc	
MASPrism (Liu et al., 2026c)	SA	WT,WN	Prefill signals from small LM	Who&When; TRAIL	Agent-Acc; Lat; Tok	

Table 1: Overview of FA methods. Abbreviations used: Category (Cat.), Link (L.), In-Context Reasoning (IC), Fine-Tuning Critics (FT), Causal Inference (CI), Statistical Analysis (SA), WHAT (WT), WHEN (WN), WHO (WO), WHERE (WR), WHY (WY), Precision (P), Recall (R), Step-Acc@n (@n), token (Tok), and latency (Lat).

$O(\log T)$ calls, and SDBL narrows failure scope before localization (Zhang et al., 2025e; Sun et al., 2026). Retrieval augmentation adds external error knowledge (Qin et al., 2024; Patil et al., 2024), as in CORRECT, which retrieves error schemata for analogical diagnosis (Yu et al., 2025).

4.2.2 Multi-Agent Collaboration

Multi-agent diagnosis reduces single-judge bias by using multiple evaluators to propose, verify, and revise failure hypotheses. RAFFLES (Zhu et al., 2025a,b; Wu et al., 2026) adopts a collaborative propose-and-verify design, where a central judge generates hypotheses and specialized evaluators iteratively filter unreliable diagnoses. DoVer (Ma et al., 2026) further grounds verification through targeted intervention and replay, testing whether a suspected correction changes the outcome. ECHO (Banerjee et al., 2025) combines hierarchical trace summaries with consensus voting, while ERRORPROBE (Li et al., 2026) performs backward tracing with a Strategist–Investigator–Arbiter team. Reflective multi-agent systems also use inter-agent critique and revision for diagnosis or recovery (Mohamed et al., 2025).

4.3 Fine-Tuning Critics

Fine-tuning methods specialize LLMs into dedicated critics for failure attribution, aiming to overcome the instability of prompt-only diagnosis. Their main bottleneck is the scarcity of high-quality attribution data (Ma et al., 2024; Zhang et al., 2024), so recent work often constructs supervision through fault injection, counterfactual replay, or trace-level anomaly signals. Aegis (Kong et al., 2026) adopts a contrastive attribution mechanism: a manipulator injects specific faults, such as hallucinations, into successful trajectories to construct paired positive and negative samples with fine-grained labels. With SFT (Zeng et al., 2024), GRPO (Shao et al., 2024), and disentangled contrastive learning (Ilse et al., 2018), the critic learns to identify where failed traces diverge from successful execution logic. AgenTracer (Zhang et al., 2026a) further introduces responsibility-oriented supervision by using counterfactual replay to identify the decisive agent-step whose oracle replacement turns failure into success, and trains critics with multi-granular RL rewards to predict responsible agents and steps in long-horizon traces.

Beyond post-hoc diagnosis, some critics are designed for runtime or label-efficient attribution. AgentAsk (Lin et al., 2025) reframes attribution as real-time calibration by monitoring inter-agent handoffs and detecting missing information before it propagates into downstream failures. To reduce costly labels, recent work casts attribu-

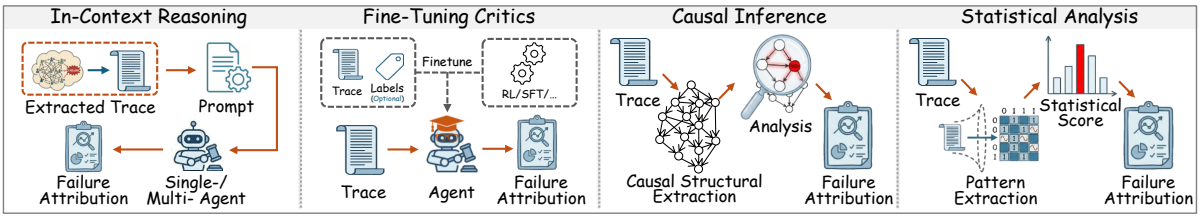

Figure 3: Illustrations of different failure attribution methods for LLM-MAS.

tion as unsupervised anomaly detection over execution traces (Yan et al., 2026; Chen et al., 2026c). MASC (Shen et al., 2025) scores each step by predicting next-action representation from history and using reconstruction error as anomaly signal, with prototype-guided enhancement for sparse early context. EAGER (Zhang et al., 2026c) learns reasoning-trace embeddings, retrieves historical failure, and supports step-wise detection together with reflexive mitigation during execution.

4.4 Causal Inference

Context-based reasoning leverages understanding but can confuse salient correlations with true causes in long, entangled traces. Causal-inference methods address this by estimating causal effects from logs or modeling dependency structures to recover mechanisms (Kiciman et al., 2023; Wu et al., 2024). One line estimates responsibility over traces. CDC-MAS (Ma et al., 2025a) transforms data-dependency graphs into performance causal graphs, then ranks agents or steps by contribution and causal effect. Another line traces dependencies: GraphTracer (Zhang et al., 2025c) builds a dependency graph to backtrack from failure-revealing outputs to upstream causes, while CHIEF (Wang et al., 2026e) constructs causal graphs to distinguish propagated symptoms from root causes. ProMAS (Zhao et al., 2026) models semantic transitions as Markovian dynamics and detects causal risk jumps for step localization. AgentDebug (Zhu et al., 2025c) combines module-level error profiling with counterfactual testing, attributing failure to the earliest candidate intervention that prevents the final error. A2P (West et al., 2025) formalizes this process as "Abduct, Act, Predict": infer latent error states, intervene on candidate steps, and predict rollout outcomes to verify attribution.

4.5 Statistical Analysis

Statistical analysis diagnoses failures by mapping execution traces to behavioral features and detecting abnormality via distributional shifts, anomaly scores, or failure correlations. It is especially suitable for subtle runtime deviations without explicit errors. SilentFailure (Pathak et al., 2026) frames this as trajectory anomaly detection using process-level signals such as token usage, latency, path structure, and prompt/context properties (Zheng et al., 2025; de Chezelles et al., 2025). Trajectory Guard (Advani, 2026) detects task–trajectory mismatches and structural incoherence with a hybrid-loss recurrent autoencoder, while AgentEval (Guo et al., 2026) models workflows as DAG-structured step evaluations to reveal error propagation.

Other methods adapt fault-localization or provenance signals to MAS traces. Famas (Ge et al., 2025) extends spectrum-based fault localization (de Souza et al., 2016) by replaying queries and ranking agent actions by suspiciousness over success/failure outcomes. MASPrism (Liu et al., 2026c) ranks failure candidates using prefill-stage negative log-likelihood and attention signals from a small language model (Merrill et al., 2026; Yuan et al., 2024). AgentXposed (Xie et al., 2025) detects malicious or intention-hiding agents via behavioral anomaly scores over inter-agent interactions, while IET (Nian et al., 2026) embeds conditioned statistical signals into generation to recover agent segments and transitions from final text.

5 Evaluation of Failure Attribution

5.1 Datasets

Figure 4 summarizes two construction pipelines for failure-attribution datasets: *human-annotated* datasets label naturally occurring failed traces via expert auditing, whereas *programmatically constructed* datasets derive attribution labels by design through controlled interventions. The consolidated overview of the representative benchmarks and their key properties is provided in Table 2.

Human-annotated datasets. These datasets collect naturally occurring failure traces and use

Dataset	Cat.	Scale	Target	Failure Module	Domain	L.
MAST-Data (Pan et al., 2025)	H.	1642	WT,WR	Specification Deviation; Inter-Agent Misalignment; Task Verification Failure	Code; Math; GAIAT; Sci.	
TRAIL (Deshpande et al., 2025)	H.	148	5Ws	Reasoning; System Execution; Planning & Coordination	Code; GAIAT	
AgentFail (Ma et al., 2025b)	H.	307	5Ws	Agent-level; Workflow-level; Platform-level	Code; GAIAT; RA	
Who&When (Zhang et al., 2025e)	H.	327	WT,WN,WO,WY	Open-ended (no fixed taxonomy)	GAIAT	
AgentRx (Barke et al., 2026)	H.	115	WT,WN,WO,WY	Plan Adherence; Invention; Invalid Invocation; Tool-output Misinterpretation; Intent Misalignment; Guardrail/System Failure	API; IR; Web/File	
TraceElephant (Chen et al., 2026b)	H.	220	WN,WO,WR	Open-ended (full execution observability)	GAIAT; SWE	
AgentHallu (Liu et al., 2026b)	H.	693	5Ws	Planning; Retrieval; Reasoning; Human-Interaction; Tool-Use	WK; Sci.; Math; GAIAT; Tool	
TracerTraj (Zhang et al., 2026a)	P.	2,476	WT,WN,WO,WY	Open-ended (no fixed taxonomy)	Code; GAIAT; Math	
Aegis-Bench (Kong et al., 2026)	P.	9,533	WT,WN,WO,WR	Specification Deviation; Inter-Agent Misalignment; Task Verification Failure	Math; Code; Sci.; GAIAT	
CORRECT-Error (Yu et al., 2025)	P.	2,264	WT,WR	Open-ended (no fixed taxonomy)	GAIAT; MHQA; Math	
HINTBench (Wang et al., 2026b)	P.	629	WT,WN,WR,WY	Goal Constraint; Factual Constraint; Capability Constraint; Procedural Constraint; State Constraint	Banking; Travel; Enterprise	
HORIZON (Wang et al., 2026d)	P.	3,132	WT,WR,WY	Environment; Instruction; False Assumption; Planning; History Accumulation; Memory Limitation	Web; OS; DB; Embodied	

Table 2: Overview of failure-attribution datasets. Abbreviations used: Category (Cat.), Human-annotated (H.), Programmatically constructed (P.), Multi-Hop QA (MHQA), Scientific Reasoning (Sci.), General AI Assistant Tasks (GAIAT) API/Tool-use Tasks (API), Incident Response (IR), Web Navigation (Web), File Manipulation (File), World Knowledge (WK), Database (DB), Operating System (OS) and Research Assistant (RA).

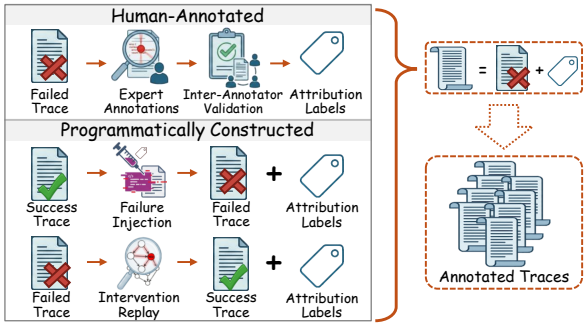

Figure 4: Illustrations of two construction pipelines for failure-attribution datasets.

expert auditors for labeling, including MAST-Data (Pan et al., 2025), TRAIL (Deshpande et al., 2025), AgentFail (Ma et al., 2025b), and Who&When (Zhang et al., 2025e). They provide high-fidelity, context-rich diagnoses, but are usually small due to costly annotation and validation. TRAIL reflects real-world observability by evaluating attribution over long structured traces that mix tool execution and model reasoning (Merrill et al., 2026; Deng et al., 2023; Xie et al., 2024). AgentFail targets platform-orchestrated agent workflows (Trivedi et al., 2024; Rawles et al., 2025; Kapoor et al., 2024), using counterfactual repair to strengthen label reliability. AgentRx (Barke et al., 2026) provides evidence-grounded critical-step and failure-category labels across API, incident-response, and web/file tasks (Kapoor et al., 2024; Koh et al., 2024; Zheng et al., 2024). TraceElephant (Chen et al., 2026b) focuses on full-observability executable traces, annotating responsible components and decisive steps. AgentHallu (Liu et al., 2026b) studies hallucination attribution with responsible-step labels, hallucination categories, and causal explanations.

Programmatically constructed datasets. They derive labels from controlled interventions (Kirmayr et al., 2026), providing 5Ws supervision without expert auditing. Examples include Aegis-Bench (Kong et al., 2026), TracerTraj (Zhang et al., 2026a), and CORRECT-Error (Yu et al., 2025). Aegis-Bench injects targeted faults into successful traces to form paired traces with known errors. TracerTraj combines counterfactual replay and fault injection to localize responsible agents and decisive steps. CORRECT-Error retrieves condensed error patterns to transfer knowledge to new traces. HINTBench (Wang et al., 2026b) builds long-horizon intrinsic-risk trajectories with typed risk-step annotations. HORIZON (Wang et al., 2026d) expands tasks across domains and attributes long-horizon failures with a diagnostic taxonomy (Jimenez et al., 2024; Yang et al., 2024).

5.2 Metrics

Existing evaluation metrics are summarized in Table 1. Prior work mainly evaluates attribution *localization*: WHO predicts the responsible *agent*, and WHEN predicts the decisive *step*, while recent studies further consider calibration and attribution reliability (Zhang et al., 2026b; Hu et al.). Because most benchmarks already provide failed traces, WHAT is often implicit as a binary outcome. Thus, Agent-/Step-Acc are the dominant metrics, with Step-Acc@n used for long-horizon cascades where nearby steps may also be plausible causes. For WHERE, studies report Module-

Method	Stg.	Key Mechanism	OH	Integration	Domain	L.
AgentAsk (Lin et al., 2025)	RT	Targeted clarification at message handoffs	Low	PnP; Lightweight module	Code; Math; QA	
MASC (Shen et al., 2025)	RT	History-conditioned anomaly detection and correction	Mod.	PnP; LLM agent	Code; Math; QA; GAIA; WebShop	
SUPERVISOR AGENT (Lin et al., 2026)	RT	History-conditioned runtime supervision and adaptive intervention	Low	PnP;LLM-free filter+ supervisor	Code; Math; QA; GAIA	
SEQCV (Yao et al., 2025)	RT	Segment-level verification and recursive subtask recovery	Mod.	GVS; Recursive re-exec.	Code; Math; QA; Multihop	
AgentXposed (Xie et al., 2025)	RT	Behavior profiling from interaction histories	Mod.	Analysis module	Security; Code; Math; QA	
SCOPE (Pei et al., 2025)	RT	Dual-stream prompt evolution for context optimization	Mod.	PnP	HLE; GAIA; DeepSearch	
HECG (Cao et al., 2026)	RT	Hierarchical error localization and corrective recovery	Mod.	Error-corrective graph	Embodied planning	
EAGER (Zhang et al., 2026d)	RT	Reasoning-trace-based failure detection and mitigation	Mod.	PnP; RTM	Code; software agents	
Trajectory Guard (Advani, 2026)	RT	Trajectory representation learning for temporal anomaly detection	Low	PnP; Lightweight modul	Security; MAS; synthetic	
ProMAS (Zhao et al., 2026)	RT	Proactive failure forecasting via Markov transition dynamics	Low	PnP; MT	Who&When	
AgentDebug (Zhu et al., 2025c)	Post	Attribution-based trajectory decomposition and feedback	High	Replay	ALFWorld; GAIA; WebShop	
DoVer (Ma et al., 2026)	Post	Intervention-based hypothesis validation	High	Replay	GAIA; AssistantBench; GSMPlus	
CE-Graph (Zhang et al., 2025d)	Post	Failure distribution modeling and graph-based workflow refinement	Mod.	WGE; Replay	Math; Code; QA	
ILWS (Costa, 2025)	Post	Reflection-driven instruction-space refinement	Mod.	Replay	Enterprise support; E-commerce	
ReCreate (Hao et al., 2026)	Post	Experience-driven scaffold refinement and hierarchical agent updates	Mod.	Replay; Hier. updates	Math; Data science; SWE; Digital tasks	
XAgen (Wang et al., 2026c)	Post	Explainability-driven failure localization & correction in multi-agent workflows	High	Replay	HCI; workflow; debugging	

Table 3: Overview of reliability-enhancement methods. Abbreviations used: intervention stage (Stg.), runtime (RT), post-hoc (Post), overhead (OH), moderate (Mod.), plug-and-play (PnP), generate–verify–split (GVS), re-execution (re-exec.), reasoning trace memory (RTM), Markov transition (MT), and workflow graph editing (WGE).

Acc or Precision/Recall/F1 when module or taxonomy labels exist, and agreement scores such as Cohens κ (McHugh, 2012) for human annotations. Micro-/macro-averaged Precision/Recall/F1 are also common for multi-class or multi-label failure types. Runtime-oriented methods additionally report efficiency metrics such as Token and Latency (Wang et al., 2026f; de Chezelles et al., 2025) to capture attribution cost.

6 FA-Driven Reliability Enhancement

Runtime intervention methods. Runtime intervention improves MAS reliability *during* execution by converting failure attribution online: after risky handoffs or steps are localized (Section 4), the system repairs context, reroutes tasks, adapts allocations, or rewrites outputs before downstream agents consume them, reducing cascading error amplification (Acharya, 2026; Bian et al., 2026; Yue et al., 2025; Wang et al., 2025d; Xu et al., 2025; Wang et al., 2025a; Yang et al., 2026).

AgentAsk (Lin et al., 2025) uses minimal clarification (Kirmayr et al., 2026; Pezeshkpour and Hruschka, 2026), asking handoff questions to fix ambiguous or missing information before propagation. MASC (Shen et al., 2025) scores execution anomalies and triggers a correction agent to rewrite abnormal outputs, forming a lightweight detect-then-rewrite loop (Agarwal, 2026). SUPERVISORAGENT (Lin et al., 2026) uses adaptive supervision to detect misinformation, inefficient reasoning, or unreliable observations and intervene via correction, guidance, and observation purification. SEQCV (Yao et al., 2025) applies segment-level cross-model verification; failed segments are recursively split and re-executed in a generate–verify–split loop. AgentXposed (Xie et al., 2025) detects intention-hiding malicious agents from interaction histories using psychology-inspired probing and behavioral monitoring. SCOPE (Pei et al., 2025) converts runtime failures and suboptimal successes into reusable prompt guidelines through dual-stream prompt evolution, enabling continual online adaptation without human intervention. HECG (Cao et al., 2026) models execution as an error-corrective graph, classifies failures by severity and recoverability, and applies local refinement, strategy switching, or replanning using causal-context graph retrieval. EAGER (Zhang et al., 2026d) encodes reasoning traces, detects abnormal patterns via contrastive similarity to historical failures, and performs reflexive mitigation with retrieved failure knowledge. Trajectory Guard (Advani, 2026) uses a Siamese recurrent autoencoder to check task–trajectory alignment and sequential validity for real-time anomaly detection. ProMAS (Zhao et al., 2026) forecasts failures by modeling reasoning as Markov transitions and detecting abnormal transitions or velocity shifts before

cascades.

Post-hoc Refinement Post-hoc refinement improves reliability after execution by using attribution results for targeted debugging, configuration repair, replay-based verification, and iterative recovery (Wang et al., 2026c; Sheng et al., 2026; Zhang, 2026b; Nanda et al., 2026; Abuzakuk et al., 2026). AgentDebug (Zhu et al., 2025c) performs attribution-guided debugging with a modular error taxonomy, localizes root causes, and generates corrective feedback for later runs. DoVer (Ma et al., 2026) converts failure hypotheses into targeted interventions and validates them through trajectory replay (Lim et al., 2026; Acharya, 2026), measuring whether interventions recover failed trials or improve progress. CE-Graph (Zhang et al., 2025d) models failures in a high-dimensional Failure Signature Space, clusters recurring failure modes, and applies verified workflow-graph edits for targeted refinement. Maestro (Wang et al., 2025c) uses reflective textual feedback from traces to identify structural failure modes and jointly optimize agent configurations and workflow graphs. ILWS (Costa, 2025) analyzes interaction traces with a reflection engine, producing version-controlled instruction-level updates over prompts, user preferences, and tool policies before optional distillation. ReCreate (Hao et al., 2026) evolves domain agents by retrieving past success/failure traces and translating reusable experience patterns into scaffold edits through a reasoning–creating pipeline. AGDebugger (Epperson et al., 2025a) supports human-in-the-loop debugging by visualizing traces, editing historical messages, and resetting execution states for interactive steering. XAgen (Wang et al., 2026c) combines workflow visualization, LLM-as-a-judge failure localization, and human feedback to support interpretable debugging and iterative refinement of multi-agent workflows.

7 Challenges and Future Directions

Reliability Beyond More Agents. Recent work questions whether increasingly complex MAS architectures always improve reliability, showing that strong single-agent baselines, in-context prompting, dynamic topology adaptation, routing, and architecture search can sometimes outperform fixed multi-agent workflows (Xu et al., 2026b; Dennis et al., 2026; Liu et al., 2026a; Xu et al., 2026a; Zhang et al., 2025a; Yang et al., 2025; Yue et al., 2025; Wang et al., 2025e). Future failure-attribution research should therefore evaluate not only which agent or step failed, but also whether the chosen collaboration topology, routing policy, or role specialization was necessary for the task.

Single-Trace Attribution Is Not Enough. Most pipelines remain *single-instance-centric*: they diagnose one failed trace and apply an intervention, often post-hoc, to recover that run, e.g., DoVer (Ma et al., 2026). However, such fixes are tightly coupled with trace-specific contexts, limiting transfer and allowing the same failure mechanisms to recur across tasks. A direction is cross-trace learning (Dai et al., 2026; Zhang, 2026b; Zhang et al., 2025b): leverage attributed traces $\{\tau^{(k)}\}$ to build reusable failure memories, replay-validated repair operators, and proactive orchestration or guard policies that reduce recurrence.

Human-in-the-Loop Diagnosis and Repair. Purely automated attribution and reliability enhancement are costly and brittle, as high-confidence diagnosis often requires repeated judging, replay, or counterfactual re-execution. Human-in-the-loop debugging can serve as a compute-aware control knob (Epperson et al., 2025b; Wu et al., 2022; Wang et al., 2026c; Jacobson et al., 2026; Ikram et al., 2026; Yan et al., 2026): experts provide sparse guidance, and the system validates and applies it via targeted re-execution to avoid exhaustive search (Natarajan et al., 2025); evaluation should therefore shift from attribution accuracy to end-to-end debuggability, i.e., the time and cost needed to localize and correct failures under bounded expert effort.

8 Conclusion

In this survey, we reviewed the lifecycle of failure management in LLM-MAS, connecting failure attribution with reliability enhancement. We formalized attribution through a structured 5Ws framework (WHAT, WHEN, WHO, WHERE, WHY) and organized existing methods into four paradigms: in-context reasoning, fine-tuning critics, causal inference, and statistical analysis. Beyond diagnosis, we identified HOW as the bridge from attribution insights to actionable reliability enhancement, covering runtime interventions and post-hoc refinements.

Limitations

This survey has several limitations. First, the literature on LLM-based multi-agent systems is developing rapidly, and new methods, benchmarks, and evaluation protocols continue to appear. Although we try to cover representative work, the survey may not include every recent preprint or system. Second, existing studies use different tasks, trace formats, and annotation standards, which makes direct comparison across methods difficult. Our taxonomy therefore focuses more on organizing common ideas than ranking methods quantitatively. Finally, failure attribution and reliability enhancement are closely related to broader topics such as agent architecture, human feedback, and safety, which are only discussed when directly relevant to our scope.

References

Suhaib Abdurahman, Etsuko Ishii, Katerina Margatina, Divya Bhargavi, Monica Sunkara, and Yi Zhang. 2026. Explicit trait inference for multi-agent coordination. *arXiv preprint arXiv:2604.19278*.

Sami Abuzakuk, Lucas Crijns, Anne-Marie Kermarrec, Rafael Pires, and Martijn de Vos. 2026. RIVA: Leveraging LLM agents for reliable configuration drift detection. In *Proceedings of the Sixth European Workshop on Machine Learning and Systems*, pages 499–509.

Vivek Acharya. 2026. Semantic consensus: Process-aware conflict detection and resolution for enterprise multi-agent LLM systems. *arXiv preprint arXiv:2604.16339*.

Laksh Advani. 2026. Trajectory guard–a lightweight, sequence-aware model for real-time anomaly detection in agentic ai. In *Proceedings of the AAAI Conference on Artificial Intelligence TrustAgent Workshop*.

Abhinav Agarwal. 2026. Refute-or-promote: An adversarial stage-gated multi-agent review methodology for high-precision LLM-assisted defect discovery. *arXiv preprint arXiv:2604.19049*.

Adi Banerjee, Anirudh Nair, and Tarik Borogovac. 2025. Where did it all go wrong? a hierarchical look into multi-agent error attribution. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. 2026. AgentRx: Diagnosing AI agent failures from execution trajectories. *arXiv preprint arXiv:2602.02475*.

Haoyu Bian, Chaoning Zhang, Jiaquan Zhang, Xingyao Li, Yuanfang Guo, Wei Dong, and Yang Yang. 2026. Weak-link optimization for multi-agent reasoning and collaboration. *arXiv preprint arXiv:2604.15972*.

Cong Cao, Jingyao Zhang, and Kun Tong. 2026. A hierarchical error-corrective graph framework for autonomous agents with LLM-based action generation. *arXiv preprint arXiv:2603.08388*.

Jingdi Chen, Hanqing Yang, Zongjun Liu, and Carlee Joe-Wong. 2026a. The five ws of multi-agent communication: Who talks to whom, when, what, and why - a survey from MARL to emergent language and LLMs. *Transactions on Machine Learning Research*. Survey Certification.

Mengzhuo Chen, Junjie Wang, Fangwen Mu, Yawen Wang, Zhe Liu, Huanxiang Feng, and Qing Wang. 2026b. Seeing the whole elephant: A benchmark for failure attribution in LLM-based multi-agent systems. *arXiv preprint arXiv:2604.22708*.

Tiejin Chen, Huaiyuan Yao, Jia Chen, Evangelos E Papalexakis, and Hua Wei. 2026c. Every response counts: Quantifying uncertainty of LLM-based multi-agent systems through tensor decomposition. *arXiv preprint arXiv:2604.08708*.

Xinhong Chen and Qing Li. 2020. Event modeling and mining: a long journey toward explainable events. *The International Journal on Very Large Data Bases*, 29(1):459–482.

Rimom Costa. 2025. Instruction-level weight shaping: A framework for self-improving AI agents. *arXiv preprint arXiv:2509.00251*.

Pengyu Dai, Weihao Xuan, Junjue Wang, Hongruixuan Chen, Jian Song, Yafei Ou, and Naoto Yokoya. 2026. Experience-driven multi-agent systems are training-free context-aware earth observers. *arXiv preprint arXiv:2602.02559*.

Thibault Le Sellier de Chezelles, Maxime Gasse, Alexandre Lacoste, Massimo Caccia, Alexandre Drouin, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Graham Neubig, Quentin Cappart, Russ Salakhutdinov, and Nicolas Chapados. 2025. The BrowserGym ecosystem for web agent research. *Transactions on Machine Learning Research*. Expert Certification.

Higor A de Souza, Marcos L Chaim, and Fabio Kon. 2016. Spectrum-based software fault localization: A survey of techniques, advances, and challenges. *arXiv preprint arXiv:1607.04347*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Simon Dennis, Michael Diamond, Rivaan Patil, Kevin Shabahang, and Hao Guo. 2026. In-context prompting obsoletes agent orchestration for procedural tasks. *arXiv preprint arXiv:2604.27891*.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. 2025. TRAIL: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*.

Shangheng Du, Jiabao Zhao, Jinxin Shi, Zhentao Xie, Xin Jiang, Yanhong Bai, and Liang He. 2026. A survey on the optimization of large language model-based agents. *ACM Computing Surveys*, 58(9):1–37.

Will Epperson, Gagan Bansal, Victor Dibia, Adam Fourney, Jack Gerrits, Erkang Zhu, and Saleema Amershi. 2025a. Interactive debugging and steering of multi-agent AI systems. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–15. Association for Computing Machinery.

Will Epperson, Gagan Bansal, Victor C Dibia, Adam Fourney, Jack Gerrits, Erkang (Eric) Zhu, and Saleema Amershi. 2025b. Interactive debugging and steering of multi-agent AI systems. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery.

Jingru Fan, Dewen Liu, Yufan Dang, Huatao Li, Yuheng Wang, Wei Liu, Feiyu Duan, Xuanwen Ding, Shu Yao, Lin Wu, and 1 others. 2026. Towards a science of collective AI: LLM-based multi-agent systems need a transition from blind trial-and-error to rigorous science. *arXiv preprint arXiv:2602.05289*.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. 2025. From LLM reasoning to autonomous AI agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*.

Yu Ge, Linna Xie, Zhong Li, Yu Pei, and Tian Zhang. 2025. Who is introducing the failure? automatically attributing failures of multi-agent systems via spectrum analysis. *arXiv preprint arXiv:2509.13782*.

Dongxin Guo, Jikun Wu, and Siu Ming Yiu. 2026. AgentEval: DAG-structured step-level evaluation for agentic workflows with error propagation tracking. *arXiv preprint arXiv:2604.23581*.

Zhezheng Hao, Hong Wang, Jian Luo, Jianqing Zhang, Yuyan Zhou, Qiang Lin, Can Wang, Hande Dong, and Jiawei Chen. 2026. ReCreate: Reasoning and creating domain agents driven by experience. *arXiv preprint arXiv:2601.11100*.

Jesse Hu, Pratyush Shukla, and Ke Huang. Verifying the verifiers: Failure attribution for agentic benchmark diagnostics and training data curation. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*.

Jinwei Hu, Yi Dong, Shuang Ao, Zhuoyun Li, Boxuan Wang, Lokesh Singh, Guangliang Cheng, Sarvapali D Ramchurn, and Xiaowei Huang. 2025. Stop reducing responsibility in LLM-powered multi-agent systems to local alignment. *arXiv preprint arXiv:2510.14008*.

Fareya Ikram, Nischal Ashok Kumar, Junyang Lu, Hunter McNichols, Candace Walkington, Neil Heffernan, and Andrew S Lan. 2026. A multi-agent approach to validate and refine llm-generated personalized math problems. *arXiv preprint arXiv:2604.05160*.

Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, pages 2127–2136. PMLR.

Yeonjun In, Mehrab Tanjim, Jayakumar Subramanian, Sungchul Kim, Uttaran Bhattacharya, Wonjoong Kim, Sangwu Park, Somdeb Sarkhel, and Chanyoung Park. 2026. Rethinking failure attribution in multi-agent systems: A multi-perspective benchmark and evaluation. *arXiv preprint arXiv:2603.25001*.

Niful Islam, Ragib Shahriar Ayon, Deepak George Thomas, Shibbir Ahmed, and Mohammad Wardat. 2026. When agents fail: A comprehensive study of bugs in LLM agents with automated labeling. *arXiv preprint arXiv:2601.15232*.

Maxwell J Jacobson, Daniel Xie, Jackson Shen, Adil Wazeer, Haiyan Wang, Xinghang Zhang, and Yexiang Xue. 2026. A multi-agent human-LLM collaborative framework for closed-loop scientific literature summarization. *arXiv preprint arXiv:2604.01452*.

Yiping Jia, Zhen Ming Jiang, Shayan Noei, and Ying Zou. 2025. MemoCoder: Automated function synthesis using LLM-supported agents. *arXiv preprint arXiv:2507.18812*.

Eric Hanchen Jiang, Levina Li, Rui Sun, Xiao Liang, Yubei Li, Yuchen Wu, Haozheng Luo, Hengli Li, Zhi Zhang, Zhaolu Kang, and 1 others. 2026. Agent Q-Mix: Selecting the right action for LLM multi-agent systems through reinforcement learning. *arXiv preprint arXiv:2604.00344*.

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 *International Conference on Learning Representations*, volume 2024, pages 54107–54157.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. 2024. OmniACT: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *European Conference on Computer Vision*, pages 161–178. Springer.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, volume 2024, pages 54928–54958.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*.

Johannes Kirmayr, Lukas Stappen, and Elisabeth André. 2026. CAR-bench: Evaluating the consistency and limit-awareness of LLM agents under real-world uncertainty. *arXiv preprint arXiv:2601.22027*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905.

Fanqi Kong, Ruijie Zhang, Huaxiao Yin, Guibin Zhang, Xiaofei Zhang, Ziang Chen, Zhaowei Zhang, Xiaoyuan Zhang, Song-Chun Zhu, and Xue Feng. 2026. Aegis: Automated error generation and attribution for multi-agent systems. In *Proceedings of the International Conference on Learning Representations*.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Jiazheng Li, Emine Yilmaz, Bei Chen, and Dieu-Thu Le. 2026. Towards self-improving error diagnosis in multi-agent systems. *arXiv preprint arXiv:2604.17658*.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Zherui Li, Yan Mi, Zhenhong Zhou, Houcheng Jiang, Guibin Zhang, Kun Wang, and Junfeng Fang. 2025. Goal-aware identification and rectification of misinformation in multi-agent systems. In *Proceedings of the International Conference on Learning Representations*.

Jonghan Lim, Mostafa Tavakkoli Anbarani, Rômulo Meira-Góes, and Ilya Kovalenko. 2026. Logic-based verification of task allocation for LLM-enabled multi-agent manufacturing systems. *arXiv preprint arXiv:2604.17142*.

Bohan Lin, Kuo Yang, Yingchuan Lai, Yudong Zhang, Chen Zhang, Guibin Zhang, Xinlei Yu, Miao Yu, Xu Wang, and Yang Wang. 2025. AgentAsk: Multi-agent systems need to ask. *arXiv preprint arXiv:2510.07593*.

Fulin Lin, Shaowen Chen, Ruishan Fang, Hongwei Wang, and Tao Lin. 2026. Stop wasting your tokens: Towards efficient runtime multi-agent systems. In *Proceedings of the International Conference on Learning Representations*.

Guangyi Liu, Haojun Lin, Huan Zeng, Heng Wang, and Quanming Yao. 2026a. Mas-on-the-fly: Dynamic adaptation of LLM-based multi-agent systems at test time. *arXiv preprint arXiv:2602.13671*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xuannan Liu, Xiao Yang, Zekun Li, Peipei Li, and Ran He. 2026b. AgentHallu: Benchmarking automated hallucination attribution of LLM-based agents. *arXiv preprint arXiv:2601.06818*.

Yang Liu, Hongjiang Feng, Junsong Pu, and Zhuangbin Chen. 2026c. MASPrism: Lightweight failure attribution for multi-agent systems using prefill-stage signals. *Preprint*, arXiv:2605.07509.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478.

Ruofan Lu, Yichen Li, and Yintong Huo. 2025. Exploring autonomous agents: A closer look at why they fail when completing tasks. *arXiv preprint arXiv:2508.13143*.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An analytical evaluation board of multi-turn LLM agents. *Advances in Neural Information Processing Systems*, 37:74325–74362.

Guoqing Ma, Jia Zhu, Hanghui Guo, Weijie Shi, Jiawei Shen, Jingjiang Liu, and Yidan Liang. 2025a. Automatic failure attribution and critical step prediction method for multi-agent systems based on causal inference. *arXiv preprint arXiv:2509.08682*.

Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2026. DoVer: Intervention-driven auto debugging for LLM multi-agent systems. In *Proceedings of the International Conference on Learning Representations*.

Xuyan Ma, Xiaofei Xie, Yawen Wang, Junjie Wang, Boyu Wu, Mingyang Li, and Qing Wang. 2025b. Diagnosing failure root causes in platform-orchestrated agentic systems: Dataset, taxonomy, and benchmark. *arXiv preprint arXiv:2509.23735*.

Aman Madaan and Tandon. 2023. Self-refine: iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, pages 46534–46594, Red Hook, NY, USA. Curran Associates Inc.

Xutao Mao, Jinman Zhao, Gerald Penn, and Cong Wang. 2026. What happens inside agent memory? circuit analysis from emergence to diagnosis. *arXiv preprint arXiv:2605.03354*.

Anna Mazhar, Huzaifa Suri, and Sainyam Galhotra. 2026. Trace-level analysis of information contamination in multi-agent systems. *arXiv preprint arXiv:2604.27586*.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Mike A Merrill, Alexander G Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E Kelly Buchanan, and 1 others. 2026. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Aissa Hadj Mohamed, Frances A Santos, and Julio Cesar dos Reis. 2025. Enhancing LLM agent effectiveness via reflective multi-agent system. In *Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações (WESAAC)*, pages 206–217. SBC.

Dany Moshkovich and Sergey Zeltyn. 2025. Taming uncertainty via automation: Observing, analyzing, and optimizing agentic AI systems. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, page 38403844.

Rahul Nanda, Chandra Maddila, Smriti Jha, Euna Mehnaz Khan, Matteo Paltenghi, and Satish Chandra. 2026. Wink: Recovering from misbehaviors in coding agents. *arXiv preprint arXiv:2602.17037*.

Sriraam Natarajan, Saurabh Mathur, Sahil Sidheekh, Wolfgang Stammer, and Kristian Kersting. 2025. Human-in-the-loop or AI-in-the-loop? automate or collaborate? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28594–28600.

Maksym Nechepurenko and Pavel Shuvalov. 2026. Coordination as an architectural layer for LLM-based multi-agent systems. *arXiv preprint arXiv:2605.03310*.

Yi Nian, Haosen Cao, Shenzhe Zhu, Henry Peng Zou, Qingqing Luan, and Yue Zhao. 2026. When only the final text survives: Implicit execution tracing for multi-agent attribution. *arXiv preprint arXiv:2603.17445*.

Tianyue Ou, Wanyao Guo, Apurva Gandhi, Graham Neubig, and Xiang Yue. 2025. Agentdiagnose: An open toolkit for diagnosing LLM agent trajectories. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 207–215.

Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, and 1 others. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Divya Pathak, Felix George, Harshit Kumar, Anuska Roy, Kaustabha Ray, Mudit Verma, and Pratibha Moogi. 2026. Detecting silent failures in multi-agentic AI trajectories (work in progress paper). In *Companion of the 17th ACM/SPEC International Conference on Performance Engineering*, pages 33–39.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Zehua Pei, Hui-Ling Zhen, Shixiong Kai, Sinno Jialin Pan, Yunhe Wang, Mingxuan Yuan, and Bei Yu. 2025. Scope: Prompt evolution for enhancing agent effectiveness. *arXiv preprint arXiv:2512.15374*.

Pouya Pezeshkpour and Estevam Hruschka. 2026. From task solving to robust real-world adaptation in LLM agents. *arXiv preprint arXiv:2602.02760*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2025. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, volume 2025, pages 406–441.

JV Roig. 2025. How do LLMs fail in agentic scenarios? a qualitative analysis of success and failure scenarios of various LLMs in agentic simulations. *arXiv preprint arXiv:2512.07497*.

Daniel Russo. 2026. More is different: Toward a theory of emergence in AI-native software ecosystems. *arXiv preprint arXiv:2604.19827*.

Mehil B Shah, Mohammad Mehdi Morovati, Mohammad Masudur Rahman, and Foutse Khomh. 2026. Characterizing faults in agentic AI: A taxonomy of types, symptoms, and root causes. *arXiv preprint arXiv:2603.06847*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Risal Shahriar Shefin, Debashis Gupta, Thai Le, and Sarra Alqahtani. 2026. Interpretable failure analysis in multi-agent reinforcement learning systems. *arXiv preprint arXiv:2602.08104*.

Xu Shen, Qi Zhang, Song Wang, Zhen Tan, Xinyu Zhao, Laura Yao, Vaishnav Tadiparthi, Hossein Nourkhiz Mahjoub, Ehsan Moradi Pari, Kwonjoon Lee, and 1 others. 2025. Metacognitive self-correction for multi-agent system via prototype-guided next-execution reconstruction. *arXiv preprint arXiv:2510.14319*.

Rui Sheng, Yukun Yang, Chuhan Shi, Yanna Lin, Zixin Chen, Huamin Qu, and Furui Cheng. 2026. Dills: Interactive diagnosis of LLM-based multi-agent systems via layered summary of agent behaviors. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–17.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Kai Sun, Wenqiang Li, Bo Dong, Yuxin Lin, Jingyao Zhang, and Bin Shi. 2026. Scope delineation before localization: A two-stage framework for enhancing failure attribution in multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33108–33116.

Gou Tan, Zilong He, Min Li, Pengfei Chen, Jieke Shi, Zhensu Sun, Ting Zhang, Danwen Chen, Lwin Khin Shar, Chuanfu Zhang, and 1 others. 2026. LIDL: LLM integration defect localization via knowledge graph-enhanced multi-agent analysis. *arXiv preprint arXiv:2601.05539*.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of LLMs. *arXiv preprint arXiv:2501.06322*.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076.

Karthik Vaidhyanathan and Henry Muccini. 2025. Software architecture in the age of agentic ai. In *European Conference on Software Architecture*, pages 41–49. Springer.

Vaishali Vinay. 2025. Failure modes in LLM systems: A system-level taxonomy for reliable AI applications. *arXiv preprint arXiv:2511.19933*.

Ziyu Wan, Yunxiang LI, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and Ying Wen. 2025. ReMA: Learning to meta-think for LLMs with multi-agent reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hanyu Wang, Yuanpu Cao, Lu Lin, and Jinghui Chen. 2026a. PreFlect: From retrospective to prospective reflection in large language model agents. *arXiv preprint arXiv:2602.07187*.

Jiacheng Wang, Jinchang Hou, Fabian Wang, Ping Jian, Chenfu Bao, and Zhonghou Lv. 2026b. HINT-Bench: Horizon-agent intrinsic non-attack trajectory benchmark. *arXiv preprint arXiv:2604.13954*.

Naibo Wang, Yifan Zhang, Sai Liu, Xinkui Zhao, Guanjie Cheng, and Yueshen Xu. 2025a. DRAMA: A dynamic and robust allocation-based multi-agent system for changing environments. *arXiv preprint arXiv:2508.04332*.

Qian Wang, Tianyu Wang, Zhenheng Tang, Qinbin Li, Nuo Chen, Jingsheng Liang, and Bingsheng He. 2025b. MegaAgent: A large-scale autonomous LLM-based multi-agent system without predefined SOPs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4998–5036, Vienna, Austria. Association for Computational Linguistics.

Wenxiao Wang, Priyatham Kattakinda, and Soheil Feizi. 2025c. Maestro: Joint graph & config optimization for reliable AI agents. *arXiv preprint arXiv:2509.04642*.

Xinru Wang, Ming Yin, Eunyee Koh, and Mustafa Doga Dogan. 2026c. XAgen: An explainability tool for identifying and correcting failures in multi-agent workflows. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–11.

Xinyu Jessica Wang, Haoyue Bai, Yiyou Sun, Haorui Wang, Shuibai Zhang, Wenjie Hu, Mya Schroder, Bilge Mutlu, Dawn Song, and Robert D Nowak. 2026d. The long-horizon task mirage? diagnosing where and why agentic systems break. *arXiv preprint arXiv:2604.11978*.

Yanbo Wang, Zixiang Xu, Yue Huang, Xiangqi Wang, Zirui Song, Lang Gao, Chenxi Wang, Xiangru Tang, Yue Zhao, Arman Cohan, and 1 others. 2025d. Dyflow: Dynamic workflow framework for agentic

reasoning. In *Advances in Neural Information Processing Systems*, pages 174148–174181.

Yawen Wang, Wenjie Wu, Junjie Wang, and Qing Wang. 2026e. From flat logs to causal graphs: Hierarchical failure attribution for LLM-based multi-agent systems. *arXiv preprint arXiv:2602.23701*.

Yingxu Wang, Siwei Liu, Jinyuan Fang, and Zaiqiao Meng. 2025e. Evoagentx: An automated framework for evolving agentic workflows. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 643–655.

Yuhang Wang, Yuling Shi, Mo Yang, Rongrui Zhang, Shilin He, Heng Lian, Yuting Chen, Siyu Ye, Kai Cai, and Xiaodong Gu. 2026f. SWE-Pruner: Self-adaptive context pruning for coding agents. *arXiv preprint arXiv:2601.16746*.

Zexin Wang, Jingjing Li, Quan Zhou, Haotian Si, Yuanhao Liu, Jianhui Li, Gaogang Xie, Fei Sun, Dan Pei, and Changhua Pei. 2025f. A survey on AgentOps: Categorization, challenges, and future directions. *arXiv preprint arXiv:2508.02121*.

Ziyi Wang, Chen Zhang, Wenjun Peng, Qi Wu, and Xinyu Wang. 2026g. TriEx: A game-based tri-view framework for explaining internal reasoning in multi-agent LLMs. *arXiv preprint arXiv:2604.20043*.

Alva West, Yixuan Weng, Minjun Zhu, Zhen Lin, Zhiyuan Ning, and Yue Zhang. 2025. Abduct, act, predict: Scaffolding causal inference for automated failure attribution in multi-agent systems. *arXiv preprint arXiv:2509.10401*.

Anpeng Wu, Kun Kuang, Minqin Zhu, Yingrong Wang, Yujia Zheng, Kairong Han, Baohong Li, Guangyi Chen, Fei Wu, and Kun Zhang. 2024. Causality for large language models. *arXiv preprint arXiv:2410.15319*.

Shuai Wu, Xue Li, Yanna Feng, Yufang Li, and Zhijun Wang. 2026. Council mode: Mitigating hallucination and bias in LLMs via multi-agent consensus. *arXiv preprint arXiv:2604.02923*.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381.

Boming Xia, Liming Zhu, Erdun Gao, Qinghua Lu, Minhui Xue, and Dino Sejdinovic. 2026. Uncertainty propagation in LLM-based systems. *arXiv preprint arXiv:2604.23505*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. 2026. From spark to fire: Modeling and mitigating error cascades in LLM-based multi-agent collaboration. *arXiv preprint arXiv:2603.04474*.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minghao Wang, and Chi Liu. 2025. Who's the mole? modeling and detecting intention-hiding malicious agents in LLM-based multi-agent systems. *arXiv preprint arXiv:2507.04724*.

Chen Xu, Yicheng Hu, Ruizi Wang, Xinyu Lin, Wenjie Wang, Dongrui Liu, and Fuli Feng. 2026a. Tacomas: Test-time co-evolution of topology and capability in LLM-based multi-agent systems. *arXiv preprint arXiv:2605.09539*.

Jiawei Xu, Arief Koesdwiady, Sisong Bei, Yan Han, Baixiang Huang, Dakuo Wang, Yutong Chen, Zheshen Wang, Peihao Wang, Pan Li, and 1 others. 2026b. Rethinking the value of multi-agent workflow: A strong single agent baseline. *arXiv preprint arXiv:2601.12307*.

Shengxiang Xu, Jiayi Zhang, Shimin Di, Yuyu Luo, Liang Yao, Hanmo Liu, Jia Zhu, Fan Liu, and Min-Ling Zhang. 2025. Robustflow: Towards robust agentic workflow generation. *arXiv preprint arXiv:2509.21834*.

Shuo Yan, Xiaolin Wen, Shaolun Ruan, Yanjie Zhang, Jiaming Mi, Yushi Sun, Huamin Qu, and Rui Sheng. 2026. InconLens: Interactive visual diagnosis of behavioral inconsistencies in LLM-based agentic systems. *arXiv preprint arXiv:2603.28106*.

Jiaxi Yang, Mengqi Zhang, Yiqiao Jin, Hao Chen, Qingsong Wen, Lu Lin, Yi He, Srijan Kumar, Weijie Xu, James Evans, and 1 others. 2025. Topological structure learning should be a research priority for LLM-based multi-agent systems. *arXiv preprint arXiv:2505.22467*.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652.

Qingwen Yang, Feiyu Qu, Tiezheng Guo, Yanyi Liu, and Yingyou Wen. 2026. Augmented runtime collaboration for self-organizing multi-agent systems: A hybrid bi-criteria routing approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29811–29819.

Yu Yao, Yiliao Song, Yian Xie, Mengdan Fan, Mingyu Guo, and Tongliang Liu. 2025. Can Dependencies Induced by LLM-Agent Workflows Be Trusted? In *Advances in Neural Information Processing Systems*, pages 17431–17469.

Runyang You, Hongru Cai, Caiqi Zhang, Qiancheng Xu, Meng Liu, Tiezheng Yu, Yongqi Li, and Wenjie Li. 2026. Agent-as-a-judge. *Preprint*, arXiv:2601.05111.

Yifan Yu, Moyan Li, Shaoyuan Xu, Jinmiao Fu, Xinhai Hou, Fan Lai, and Bryan Wang. 2025. CORRECT: Condensed error recognition via knowledge transfer in multi-agent systems. *arXiv preprint arXiv:2509.24088*.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, and 1 others. 2024. R-Judge: Benchmarking safety risk awareness for LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1467–1490.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. 2025. Masrouter: Learning to route LLMs for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15549–15572.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2024. Agenttuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077.

Yuhao Zhan, Tianyu Fan, Linxuan Huang, Zirui Guo, and Chao Huang. 2026. Why your deep research agent fails? on hallucination evaluation in full research trajectory. *arXiv preprint arXiv:2601.22984*.

Chenchen Zhang. 2026a. Reinforcement learning for LLM-based multi-agent systems through orchestration traces. *arXiv preprint arXiv:2605.02801*.

Di Zhang. 2026b. AgentDevel: Reframing self-evolving LLM agents as release engineering. *arXiv preprint arXiv:2601.04620*.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. 2025a. Multi-agent architecture search via agentic supernet. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 75834–75852.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. 2025b. Memevolve: Meta-evolution of agent memory systems. *arXiv preprint arXiv:2512.18746*.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng YAN. 2026a. AgenTracer: Who is inducing failure in the LLM agentic systems? In *The Fourteenth International Conference on Learning Representations*.

Heng Zhang, Yuling Shi, Xiaodong Gu, Haochen You, Zijian Zhang, Lubin Gan, Yilei Yuan, and Jin Huang. 2025c. GraphTracer: Graph-guided failure tracing in LLM agents for robust multi-turn deep search. *arXiv preprint arXiv:2510.10581*.

Jiaxin Zhang, Caiming Xiong, and Chien-Sheng Wu. 2026b. Agentic confidence calibration. *arXiv preprint arXiv:2601.15778*.

Jusheng Zhang, Kaitong Cai, Qinglin Zeng, Ningyuan Liu, Stephen Fan, Ziliang Chen, and Keze Wang. 2025d. Failure-driven workflow refinement. *arXiv preprint arXiv:2510.10035*.

Lingzhe Zhang, Tong Jia, Mingyu Wang, Weijie Hong, Chiming Duan, Minghua He, Rongqian Wang, Xi Peng, Meiling Wang, Gong Zhang, and 1 others. 2026c. EAGER: Efficient failure management for multi-agent systems with reasoning trace representation. *arXiv preprint arXiv:2603.21522*.

Lingzhe Zhang, Tong Jia, Mingyu Wang, Weijie Hong, Chiming Duan, Minghua He, Rongqian Wang, Xi Peng, Meiling Wang, Gong Zhang, and 1 others. 2026d. Efficient failure management for multi-agent systems with reasoning trace representation. *arXiv preprint arXiv:2603.21522*.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. 2025e. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *Proceedings of the International Conference on Machine Learning*, pages 76583–76599.

Weihao Zhang, Yitong Zhou, Huanyu Qu, and Hongyi Li. 2026e. Loosely-structured software: Engineering context, structure, and evolution entropy in runtime-rewired multi-agent systems. *arXiv preprint arXiv:2603.15690*.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. 2024. Agent-SafetyBench: Evaluating the safety of LLM agents. *arXiv preprint arXiv:2412.14470*.

Zhiyuan Zhang, Bing Wang, Faisal Ahmed, IV Ramakrishnan, Rong Zhao, Asa Viccellio, and Klaus Mueller. 2013. The five Ws for information visualization with application to healthcare informatics. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1895–1910.

Xinkui Zhao, Sai Liu, Yifan Zhang, Qingyu Ma, Guanjie Cheng, Naibo Wang, and Chang Liu. 2026. ProMAS: Proactive error forecasting for multi-agent systems using markov transition dynamics. *arXiv preprint arXiv:2603.20260*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. GPT-4V (ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning*, pages 61349–61385. PMLR.

Yusheng Zheng, Yanpeng Hu, Tong Yu, and Andi Quinn. 2025. AgentSight: System-level observability for AI agents using ebpf. In *Proceedings of the 4th Workshop on Practical Adoption Challenges of ML for Systems*, pages 110–115.

Pengcheng Zhou, Yinglun Feng, Halimulati Julaiti, and Zhongliang Yang. 2025. Why do AI agents communicate in human language? *arXiv preprint arXiv:2506.02739*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.

Chenyang Zhu, Spencer Hong, Jingyu Wu, Kushal Chawla, Charlotte Tang, Youbing Yin, Nathan Wolfe, Erin Babinsky, and Daben Liu. 2025a. RAFFLES: Reasoning-based attribution of faults for LLM systems. *arXiv preprint arXiv:2509.06822*.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Daisy Zhe Wang, Zhenhailong Wang, Cheng Qian, Robert Tang, Heng Ji, and 1 others. 2025b. Multiagentbench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, and 1 others. 2025c. Where LLM agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*.

Mingjie Zhu, Mingcheng Li, Mingzhang Cui, Ziyun Qian, Yue Jiang, Minghao Han, Mingyang Sun, Zizhi Chen, Yizhou Liu, Yang Liu, and 1 others. 2026. A survey on large language model-based multi-agent systems: Paradigms, applications, and challenges.

A Detailed Summary of Failure-Attribution Datasets

To complement the compact dataset overview in the main text, we provide a full comparison of existing failure-attribution datasets in Table 4. The appendix table preserves additional details that are useful for reproducing and interpreting prior benchmarks, including the specific MAS frameworks covered by each dataset and the way attribution labels are obtained. In particular, the "Covered MAS" column indicates the agent frameworks or systems from which failure trajectories are collected or constructed, while the "Annotation" column describes whether the labels are produced by human experts, counterfactual replay, fault injection, synthesis, LLM-based judging, or human verification. These details help distinguish benchmarks that study naturally occurring human-annotated failures from those that are programmatically constructed for scalable evaluation.

Dataset	Cat.	Scale	Attrition Label	Failure Module	Covered MAS	Domain	Annotation	Link
MAST-Data (Pan et al., 2025)	H.	1642	D_{WHAT}; D_{WHERE}	Specification Deviation; Inter-Agent Misalignment; Task Verification Failure	HyperAgent; AppWorld; AG2; ChatDev; MetaGPT; Magentic-One; OpenManus	Code; Math; GAIAT; Sci.	Human Expert	
TRAIL (Deshpande et al., 2025)	H.	148	D_{WHAT}; D_{WHEN}; D_{WHO}; D_{WHERE}; D_{WHY}	Reasoning; System Execution; Planning & Coordination	OpenDeepResearch; CodeAct	Code; GAIAT	Human Expert	
AgentFail (Ma et al., 2025b)	H.	307	D_{WHAT}; D_{WHEN}; D_{WHO}; D_{WHERE}; D_{WHY}	Agent-level; Workflow-level; Platform-level	Dify; Coze	Code; GAIAT; RA	Human Expert& Counterfactual	
Who&When (Zhang et al., 2025e)	H.	327	D_{WHAT}; D_{WHEN}; D_{WHO} D_{WHY}	Open-ended (no fixed taxonomy)	AG2; MagneticOne	GAIAT	Human Expert	
AgentRx (Barke et al., 2026)	H.	115	D_{WHAT}; D_{WHEN}; D_{WHO}; D_{WHY}	Plan Adherence; Invention; Invalid Invocation; Tool-output Misinterpretation; Intent Misalignment; Guardrail/System Failure	τ-bench; Flash; Magentic-One	API; IR; Web/File	Human Expert	
TraceElephant (Chen et al., 2026b)	H.	220	D_{WHEN}; D_{WHO}; D_{WHERE}	Open-ended (full execution observability)	Captain-Agent; Magentic-One; SWE-Agent	GAIAT; SWE	Human Expert	
AgentHallu (Liu et al., 2026b)	H.	693	D_{WHAT}; D_{WHEN}; D_{WHO}; D_{WHERE}; D_{WHY}	Planning; Retrieval; Reasoning; Human-Interaction; Tool-Use	SmolAgents; OpenDeepSearch; OpenManus; OctoTools; Magentic-One; OWL; Function-Calling Agent	WK; Sci.; Math; GAIAT; Tool	Human Expert	
TracerTraj (Zhang et al., 2026a)	P.	2,476	D_{WHAT};D_{WHEN}; D_{WHO}; D_{WHY}	Open-ended (no fixed taxonomy)	MetaGPT; AutoGen; AgentPrune; AFlow; Smolagents; OWL-Workforce	Code; GAIAT; Math	Counterfactual& Injection	
Aegis-Bench (Kong et al., 2026)	P.	9,533	D_{WHAT}; D_{WHEN}; D_{WHO}; D_{WHERE}	Specification Deviation; Inter-Agent Misalignment; Task Verification Failure	MacNet; DyLAN; Debate; AgentVerse; MagneticOne; SmolAgents	Math; Code; Sci.; GAIAT	Injection	
CORRECT-Error (Yu et al., 2025)	P.	2,264	D_{WHAT}; D_{WHERE}	Open-ended (no fixed taxonomy)	Magic-One; Autogen	GAIAT; MHQA; Math	Injection	
HINTBench (Wang et al., 2026b)	P.	629	D_{WHAT}; D_{WHEN}; D_{WHERE}; D_{WHY}	Goal Constraint; Factual Constraint; Capability Constraint; Procedural Constraint; State Constraint	Synthetic long-horizon agents	Banking; Travel; Enterprise	Synthesis& Human Verify	
HORIZON (Wang et al., 2026d)	P.	3,132	D_{WHAT}; D_{WHERE}; D_{WHY}	Environment; Instruction; False Assumption; Planning; History Accumulation; Memory Limitation	GPT-5; Claude-4 variants	Web; OS; DB; Embodied	LLM Judge& Human Verify	

Table 4: Overview of failure-attribution datasets. Abbreviations used: Category (Cat.), Human-annotated (H.), Programmatically constructed (P.), Multi-Hop QA (MHQA), Scientific Reasoning (Sci.), General AI Assistant Tasks (GAIAT) API/Tool-use Tasks (API), Incident Response (IR), Web Navigation (Web), File Manipulation (File), World Knowledge (WK), Database (DB), Operating System (OS) and Research Assistant (RA).