From Coordination to Authority

Evidence Boundaries in Multi-Agent AI Reliability

Nicholas Zinner Beacon Bot

Future Shock (`future-shock.ai`)

August 9, 2026

Abstract

"Reliability" in LLM-based multi-agent systems now names several different research problems: harness sensitivity, coordination, failure attribution, error propagation, verification, permission, and provenance. Evidence about one is routinely used to support a claim about the next: communication is read as coordination, a critic as independent validation, consensus as permission, and a transcript as an accountable record. We systematize a bounded 2025–2026 literature around those invalid transitions. The field's sharpest disagreement—whether redundancy buys reliability—turns on architecture: independent parallel sampling with voting and role-differentiated pipelines with shared context have opposite error dynamics. Measurement creates a second fault line, because mean improvement and tail corruption can coexist in the same cascade. We identify the transitions that remain weakly evidenced and propose a reporting record matched to the strength of the claim. This is a source-grounded narrative systematization, not an exhaustive review or statistical meta-analysis.

1 Reliability Became Several Problems

Hold a model's weights fixed and change only the scaffolding around it—the prompt format, action grammar, retry policy, orchestration loop—and measured agent performance moves by roughly ten to twenty percentage points on a fixed SWE-bench Verified subset [2]. The difference comes from details the benchmark did not report.

Consider a system now common across the literature. A planner decomposes a task and assigns roles. Several agents exchange messages and produce an artifact. A verifier approves it against criteria the planner generated. A vote authorizes deployment, an executor changes external state, and the retained transcript does not establish which evidence was current, who possessed authority, or whether the resulting state matched the approved artifact.

Every component may have functioned exactly as implemented. The system can still fail, because each stage answers a different question. Was an artifact produced? Did the agents preserve the information required to coordinate? Was the artifact correct? Was the check independent? Was the action permitted? Can another party reconstruct what happened? One success score cannot answer six questions.

The 2025–2026 literature has begun pulling these apart. Failure taxonomies made recurrent breakdowns nameable [1]. Harness studies moved the unit of analysis from a model to a model–harness configuration [2, 3]. Attribution benchmarks asked which component and step became responsible [5, 6, 7]. Fault-injection work measured propagation and recovery under known disturbances [9, 8]. Verification-aware planning pushed checking into decomposition [10]. Authorization work asked what may become external action [11].

Figure 1 organizes that progression as an evidence chain and marks the inference at each boundary. It is an interpretive lens, not a validated taxonomy. Sources center on academic work first made public in 2025–2026 and verified against primary paper or proceedings pages. Future Shock's own research motivated several distinctions and appears only as reflexive illustration, never as independent support.

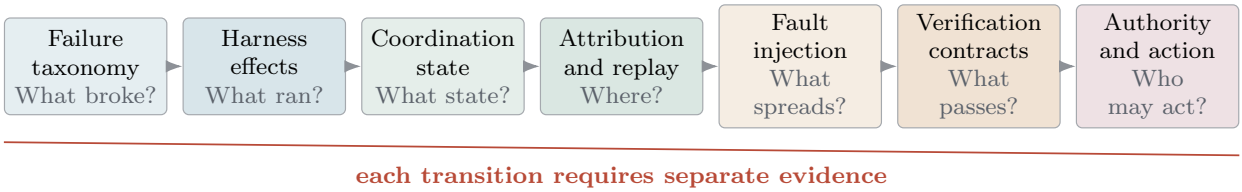

Figure 1: Different methods evidence different transitions. Movement from one box to the next is an additional claim, not an automatic consequence of success in the preceding box. Durable reconstruction cuts across the chain and asks what remains inspectable afterward.

2 How the State of the Art Got Here

2.1 Failure taxonomy: making breakdowns legible

MAST organized observable breakdowns in multi-agent systems into fourteen failure modes across three families: system design, inter-agent misalignment, and task verification [1]. Built from traces across frameworks including AutoGen, ChatDev, and CrewAI, it gave the field a shared vocabulary for patterns that otherwise vanish inside long conversational logs. Its central argument is organizational rather than model-centric: better base models will not resolve the full taxonomy, because sophisticated individuals can still be assembled into a structure that fails.

Six experts developed the taxonomy from 150 traces; they did not independently label the full corpus of roughly 1,600, most of which was annotated by an automated pipeline. Strong agreement on *applying* the refined taxonomy establishes consistent use by trained annotators, not that its categories are exhaustive, mutually exclusive, or uniquely causal.

Naming a failure pattern does not identify which component and step became responsible, whether recovery remained possible, or which intervention would have prevented the outcome.

2.2 Harness effects: changing the unit under test

Harness-focused work undercuts model-only readings of agent performance. Holding weights fixed while varying scaffolding moves scores by roughly 10–20 points on a fixed coding subset, which is larger than the margin separating many published model comparisons [2]. Harness-Bench extends the concern across realistic workflows, reporting 5,194 trajectories over eight model backends and six configurable harnesses while preserving traces, artifacts, validator outputs, token counts, permission violations, and process-quality signals [3].

The execution condition is a model–harness–tool–environment configuration, including routing, permissions, budgets, timeouts, and fallback behavior.

Varying complete harnesses reveals configuration sensitivity but does not isolate the mechanism behind each effect. Aggregate harness scores are not controlled product rankings when model pairing, task mix, budgets, and scoring differ across conditions.

2.3 Communication is not coordination

Silo-Bench isolates a communication–reasoning gap: agents exchange messages successfully while failing to reason over information distributed across participants [4]. Message traffic is an easy proxy for coordination and a weak one. Coordination requires some usable representation of current state—ownership, dependencies, evidence, blockers, commitments, corrections, completion criteria. Raw conversation may contain those facts without making them current, salient, or binding.

2.4 Attribution: from category to component and step

TraceElephant evaluates attribution at agent and step level [5]. The best static configurations reach roughly 65.9% accuracy at agent level but only about 30.3% at step level, and removing task inputs degrades step-level accuracy by up to 76% relative to full information. Partial output transcripts are frequently insufficient for diagnosis.

MP-Bench challenges the premise that every failed trajectory has one privileged root cause, showing that multiple causal perspectives can remain defensible when specification errors, communication failures, and verification gaps interact [6]. VerifyMAS reframes attribution as hypothesis verification: propose an error, test whether it is present, whether it explains the outcome, and whether it is attributable to a component or step [7].

These methods improve diagnosis without making the diagnosing model authoritative. Higher benchmark accuracy for an LLM verifier does not establish structural independence, access to a separate evidence path, or standing to determine downstream action.

2.5 Fault injection: from description to intervention

Fault injection introduces a known disturbance and measures what follows, rather than inferring a mechanism from a failure that already occurred. MAS-FIRE makes propagation, detection, silent failure, defensive response, and recovery experimentally visible, with recovery varying by topology [9].

Injected faults need not match the distribution, ambiguity, or adversarial character of deployment failures. Controlled interventions strengthen evidence about the tested mechanism without establishing its production incidence.

2.6 Verification becomes part of planning

VERIMAP moves checking from a terminal critic pass into decomposition, associating subtasks with explicit verification functions before downstream work inherits their outputs [10]. This gives intermediate checks blocking power that a post-hoc review may lack.

Verification still depends on where the criterion comes from. A deterministic oracle, a hidden test, a learned evaluator, a same-context critic, a human reviewer, and a formal proof establish different things. A planner-generated verification function can block malformed output while inheriting the planner's assumptions about what correct means.

2.7 Authority enters the technical literature

Authorization work extends the chain from correct computation to permitted action. *Overlaying Governance* treats delegation as a contractual runtime predicate rather than a static token, modeling principals, recursive delegation, contextual scope, attenuation, expiry, and revocation as a compositional overlay on relationship-based access control, with soundness proofs and runtime

benchmarks on synthetic authorization graphs [11]. Its motivating observation is that OAuth-style scopes were designed for fixed principals and pre-enumerated actions, while agents generate actions that were never enumerated and delegate recursively to other agents.

Recommendation, verification, approval, authorization, execution, and observed outcome are distinct states. Formal permission correctness establishes that a system enforced a policy. It does not establish that the decision was sound, adequately supervised, or legitimate.

3 Where the Field Disagrees

Two of the literature's strongest recent results point in opposite directions.

3.1 Does redundancy buy reliability?

The Six Sigma Agent decomposes a task into atomic, independently verifiable actions, executes each one n times in parallel across diverse LLMs, clusters the outputs semantically, and takes the winning cluster. It proves system error falls as $O(p^{\lceil n/2 \rceil})$ for per-action error rate p, and reports 3.4 defects per million opportunities at thirteen agents across three enterprise use cases. Its headline 14,700× improvement is measured against single GPT-4o-mini execution at 50,000 DPMO—the weakest baseline evaluated. The reported 80% cost reduction is relative to using more expensive models, with roughly 47% added latency at five parallel samples [12].

By contrast, From Spark to Fire models collaboration as a directed dependency graph and finds that minor inaccuracies are cited and reused until they harden into system-level false consensus. Across six mainstream frameworks it identifies cascade amplification, topological sensitivity to hub nodes, and consensus inertia. A single injected error seed produces widespread failure; its governance layer prevents final infection in at least 89% of runs [8].

The papers describe different systems. Six Sigma's guarantee rests on three explicit conditions: errors across invocations are statistically independent, per-action error is below one half, and failed agents do not converge on the same wrong answer. From Spark to Fire studies the regime where the first and third conditions fail: agents read one another's intermediate outputs, and shared context is the collaboration mechanism.

The label "multi-agent" therefore covers two architectures with opposing error dynamics:

- **Redundant sampling.** The same atomic task is executed repeatedly, in parallel, ideally across model families, with no cross-talk. Under Six Sigma's assumptions, aggregation can vote out uncorrelated errors.
- **Differentiated pipelines.** Distinct roles pass work along dependencies and consume upstream output. More stages create more opportunities to inherit a correlated error.

Independence is necessary but not sufficient. In the left-hand architecture it makes errors distinguishable; voting or verification removes them. In the right-hand architecture, later agents directly consume earlier outputs, so context reuse correlates errors by design. This is an LLM-specific form of a familiar fault-tolerance problem: N-version programming experiments showed decades ago that independently written programs fail on correlated inputs more often than independence predicts [18]. Shared specifications created common-mode failures there; LLM pipelines add a direct inheritance channel through generated context. Reading either modern result without its architecture invites a confident conclusion applied to the wrong system.

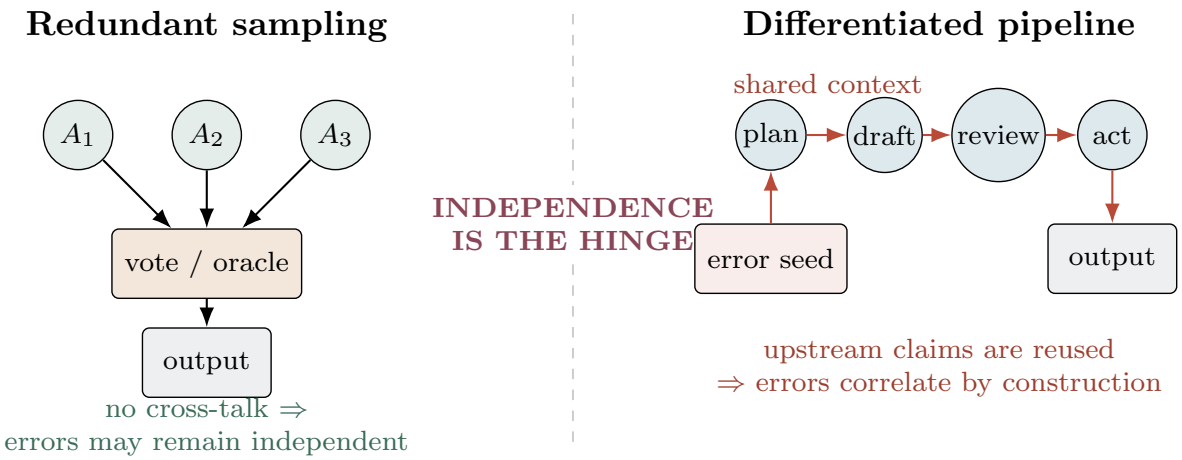

Figure 2: Why the redundancy literature appears contradictory. Majority voting improves reliability when executions remain independent; sequential role pipelines reuse upstream context, making correlated error a feature of the architecture rather than a sampling accident.

3.2 The same cascade, measured two ways

Hallucination Cascade reports an apparent gain from collaboration. Across 500 sequential cascade experiments, ten knowledge domains, three heterogeneous models, and 1,250 evaluated responses, deeper cascades reduce the normalized hallucination score from 0.422 at the first agent to 0.272 at the final agent. The reported amplification factor is 0.644, meaning net attenuation, and three-agent chains end roughly 0.073 lower than two-agent chains [16].

The same study reports factual accuracy declining from 0.789 to 0.769—a small but statistically significant change—alongside losses in factual consistency and response quality at every transition. Our interpretation is that the cascade may be converging on claims that are harder to mark wrong rather than adding information. The accuracy decline is too small to establish that mechanism by itself; the divergence between metrics is the evidence.

The two studies estimate different statistics. From Spark to Fire injects a seed and asks whether *that error* reaches the final artifact—a tail question about targeted corruption [8]. Hallucination Cascade measures average inconsistency across naturally generated claims. The statistics can move in opposite directions: smoothing and hedging may lower mean hallucination without protecting against a false premise inherited downstream.

A benchmark reporting only central tendency cannot distinguish average improvement from tail vulnerability. Figure 3 shows the reported endpoints; improvement in one metric did not imply improvement in factual accuracy.

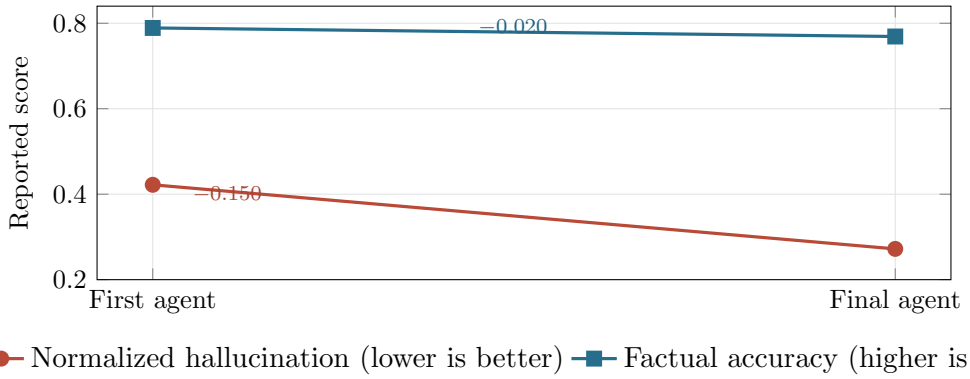

Figure 3: Metric divergence in three-agent chains reported by Hallucination Cascade [16]. The hallucination metric improves substantially while factual accuracy deteriorates. Only first-agent and final-agent values are plotted; the annotations give the net change between them, and no intermediate values are inferred.

3.3 When does adding agents help at all?

Across 260 configurations, six benchmarks, five architectures, and three model families, performance relative to a single-agent baseline ranges from +80.8% on decomposable financial reasoning to −70.0% on sequential planning [13]. Trace-level error amplification varies as sharply: 17.2× under independent parallel execution, 7.8× under decentralized coordination, and 4.4× under centralized coordination. Here "independent" means parallel outputs aggregated without voting or verification; Six Sigma adds the filter. Independence makes errors filterable, but only aggregation or verification filters them. The study's model explains a modest share of outcome variance ($R^2 = 0.373$) while selecting the best architecture in 87% of held-out configurations, distinct prediction problems that should not be conflated.

Many published workflows are homogeneous: every agent shares one base model and differs only in prompt, tools, and position. Across seven benchmarks, a single agent role-playing the workflow through multi-turn conversation matches homogeneous multi-agent performance while reusing its KV cache. The resulting OneFlow designs cut GPT-4o mini inference cost from \$0.198 to \$0.026 on HumanEval and from \$2.343 to \$0.819 on MATH [14].

A vendor-authored study based on proprietary traces reports "silent" degradation increasing with chain length [17]. Its uninspectable evidence and proposed exponential "entropy principle" do not support a general law; the observation only reinforces the need to test depth explicitly.

Coordination helps when work is decomposable and verification is centralized. It hurts when work is sequential, context is shared, and no independent check sits between stages. Whether errors can correlate is the moderator, yet almost every paper in this corpus leaves it implicit.

3.4 A second, quieter dispute

Benchmarks that assign one responsible agent and step assume a privileged cause exists to be found; MP-Bench shows that several attributions can remain defensible for one trajectory [6]. If failures are multi-causal, single-label ground truth rewards models for resolving ambiguity confidently rather than correctly.

4 Invalid Shortcuts in Reliability Claims

Evidence does not travel automatically from one reliability question to the next. Table 1 pairs each claim with the evidence it requires and the shortcut that evidence does not license; it is an applicability map, not an eight-stage maturity model.

Table 1: Claim-dependent questions, representative work, and the inference each does not license.

Question	Relevant evidence	Representative work	Invalid shortcut
What executed?	Model version, harness, tools, permissions, budgets, timeouts	[2, 3]	Model name → system capability
What was specified?	Roles, dependencies, completion, clarification, escalation	[1]	Role labels → adequate contract
What state survived?	Typed handoffs, commitments, blockers, evidence lineage	[4, 8, 17]	Messages → coordination
What failed where?	Full traces, environment state, replay, intervention	[5, 6, 9]	Transcript → cause
What was checked?	Criterion origin, checker, calibration, blocking power	[10, 7, 16]	Critic output → verified result
How independent?	Separate context, evidence path, criterion, process	[12, 8]	Different persona → independence
Who could act?	Principal, scope, delegation, approval, revocation	[11]	Consensus → permission
What survives?	Versioned artifacts, receipts, outcomes, corrections	[15]	Log availability → accountable record

A bounded mathematical benchmark need not document organizational legitimacy, and a formal authorization paper need not perform step-level attribution. Requirements follow the claim: capability needs enough execution detail to identify the tested configuration; independence needs evidence about criterion origin and validator dependence; deployment or safe delegation has to reach authority and action.

Scorer design determines whether a benchmark rewards competence, survival, agreement, hedging, or schema completion; a cascade can lower mean hallucination while eroding factual accuracy [16]. Human oversight functions as a mechanism only when the reviewer's information, workload, override power, and interventions are observable. Simpler-system baselines identify whether a gain belongs to the architecture. Resource accounting asks whether it was worth buying: one system reports an 80% cost reduction through redundancy [12], while another reports order-of-magnitude savings by removing agents [14].

5 Where the Evidence Chain Still Breaks

Evidence becomes thinnest after task-level verification. Three transitions remain poorly connected to the benchmark literature, each with a different reporting obligation.

5.1 Verification to independent validation

An output can be checked by a verifier that shares the producer's failure modes. Structural independence may require a separate process or model family, context, evidence path, criterion, and verdict the producer cannot override. Role names supply none of these. Claims of independent validation should report criterion origin, evidence separation, calibration, and blocking power.

5.2 Verified result to authorized action

A correct, verified output is not thereby permitted to change external state. Reliable action requires a principal, delegation path, bounded scope, authorization state, executor, and revocation or recovery. Evaluation should cover both unsafe permission leakage and legitimate work blocked by rigid policy. Authorization work has begun formalizing this [11], but remains disconnected from benchmarks that rarely record who could act or under what scope.

5.3 Action to reconstructable record

A transcript is not an institutional record. Reconstruction may require the artifact version approved, the criterion and verdict, authority state, executed action, resulting external state, and subsequent corrections. Survey work has begun mapping execution provenance [15], but the requirement is stricter than retention: a successor should recover what was committed without depending on the original agents' conversational memory.

Table 2 summarizes the corresponding record. A module becomes relevant only when a paper's claim reaches that transition.

Table 2: A modular reporting record. Modules become relevant when the paper's claim reaches the transition they describe.

Module	Triggering claim	Minimum record
Configuration	Capability or performance	Model and harness versions; prompts/action format; tools and permissions; budgets, timeouts, retries, substitutions
Coordination	Collaboration or robustness	Roles and dependencies; handoff schema; structured versus conversational state; context-sharing topology; correction propagation
Verification	Checked or reliable output	Criterion origin; checker type; evidence path; false accepts/rejects; blocking and override power
Authority	Safe delegation or deployment	Principal; delegation chain; action/resource scope; approval, authorization, execution, expiry, revocation
Reconstruction	Auditability or accountability	Artifact versions; manifests and receipts; external state changes; correction history; replay or recovery path

A capability benchmark need not document governance it never invoked. A deployment-safety claim cannot stop at task accuracy. The record keeps evidence about one transition from carrying a stronger claim across the next.

If only one field were added to current practice, it should be context-sharing topology: cheap to report, rarely disclosed, and decisive for whether errors can correlate. A close second is distributional results. A mean improvement and a tail vulnerability are compatible; reporting one does not bound the other.

6 Limitations

This narrative systematization is not exhaustive. Its eighteen sources center on 2025–2026 LLM multi-agent reliability; selection remains author judgment, no independent second coder reviewed the characterizations, and several sources are revisable preprints. Adjacent literatures—ensemble theory, software fault tolerance, scalable oversight, and red-teaming—are acknowledged but not surveyed. Evidential weight ranges from peer-reviewed benchmarks to proprietary vendor reports and is marked at the point of use rather than averaged away.

The paper characterizes disclosed artifacts, not production prevalence. Its questions are not exhaustive; authority means permission and delegation here, not legal or social legitimacy. Adversarial robustness and temporal drift remain cross-cutting gaps.

Authorship shares the central limitation. Beacon Bot drafted and revised the manuscript with human direction but no independent validator. Primary-source and adversarial checks are checks, not independence; this remains a single-analyst reading.

7 Conclusion

The literature is not converging on one reliability score. It is decomposing reliability into execution conditions, coordination, attribution, intervention, verification, permission, and record. Evidence about one transition should not be spent on a claim about the next.

The redundancy dispute shows why. It conflates architectures with opposite error dynamics, and turns on two properties reported in almost neither: whether errors can correlate and whether the architecture filters them before they propagate.

A task score can show that a configuration produced an answer. It cannot show that agents coordinated, that a checker was independent, that an action was permitted, or that the decision can be reconstructed. A multi-agent reliability claim is only as strong as the weakest unevidenced transition between configuration, coordination, verification, authority, and reconstructable record.

References

[1] M. Cemri, M. Z. Pan, S. Yang, L. A. Agrawal, B. Chopra, R. Tiwari, K. Keutzer, A. Parameswaran, D. Klein, K. Ramchandran, M. Zaharia, J. E. Gonzalez, and I. Stoica. Why Do Multi-Agent LLM Systems Fail? NeurIPS 2025 Datasets and Benchmarks Track. arXiv:2503.13657.

[2] Y. Zhang, J. Wang, Y. Ge, W. Xu, J. Hamm, and C. K. Reddy. Stop Comparing LLM Agents Without Disclosing the Harness. arXiv:2605.23950, May 2026.

[3] Y. Yao, X. Tan, C.-H. Liu, Y. Li, Z. Wang, W. Yu, Z. Tan, Y. Tian, G. Zhao, L. Sun, X. Zhang, and T. Yang. Harness-Bench: Measuring Harness Effects across Models in Realistic Agent Workflows. arXiv:2605.27922, May 2026.

[4] Y. Zhang, F. Liu, Y. Shan, X. Huang, X. Yang, Y. Zhu, X. Cheng, C. Liu, K. Zeng, T. J. Zhang, and W. Jiang. Silo-Bench: A Scalable Environment for Evaluating Distributed Coordination in Multi-Agent LLM Systems. ACL 2026 Main Conference. arXiv:2603.01045.

[5] M. Chen, J. Wang, F. Mu, Y. Wang, Z. Liu, H. Feng, and Q. Wang. Seeing the Whole Elephant: A Benchmark for Failure Attribution in LLM-based Multi-Agent Systems. ACL 2026. arXiv:2604.22708.

[6] Y. In, M. Tanjim, J. Subramanian, S. Kim, U. Bhattacharya, W. Kim, S. Park, S. Sarkhel, and C. Park. Rethinking Failure Attribution in Multi-Agent Systems: A Multi-Perspective Benchmark and Evaluation. arXiv:2603.25001, March 2026.

[7] H. Qiao, H. Tong, E.-P. Lim, B. Liu, and G. Pang. VerifyMAS: Hypothesis Verification for Failure Attribution in LLM Multi-Agent Systems. arXiv:2605.17467, May 2026.

[8] Y. Xie, C. Zhu, X. Zhang, T. Zhu, D. Ye, M. Qi, H. Chen, and W. Zhou. From Spark to Fire: Modeling and Mitigating Error Cascades in LLM-Based Multi-Agent Collaboration. arXiv:2603.04474, March 2026.

[9] J. Jia, Z. Deng, Z. Chen, Y. Wang, and Z. Zheng. MAS-FIRE: Fault Injection and Reliability Evaluation for LLM-Based Multi-Agent Systems. arXiv:2602.19843, February 2026.

[10] T. Xu, D. Zhang, K. Mitra, and E. Hruschka. Verification-Aware Planning for Multi-Agent Systems. EACL 2026, Long Papers, pp. 7528–7546.

[11] A. Ibrahim and Y. Li. Overlaying Governance: A Compositional Authorization Framework for Delegation and Scope in Agentic AI. arXiv:2606.03518, June 2026.

[12] K. Patel, S. Surendira, J. George, and S. Kapale. The Six Sigma Agent: Achieving Enterprise-Grade Reliability in LLM Systems Through Consensus-Driven Decomposed Execution. arXiv:2601.22290, January 2026.

[13] Y. Kim, K. Gu, C. Park, C. Park, S. Schmidgall, A. A. Heydari, Y. Yan, Z. Zhang, Y. Zhuang, Y. Liu, M. Malhotra, P. P. Liang, H. W. Park, Y. Yang, X. Xu, Y. Du, S. Patel, T. Althoff, D. McDuff, and X. Liu. Towards a Science of Scaling Agent Systems. arXiv:2512.08296, December 2025.

[14] J. Xu, A. Koesdwiady, S. Bei, Y. Han, B. Huang, D. Wang, Y. Chen, Z. Wang, P. Wang, P. Li, and Y. Ding. Rethinking the Value of Multi-Agent Workflow: A Strong Single Agent Baseline. arXiv:2601.12307, January 2026.

[15] Y. Wang, J. Zhang, T. Cai, Z. Liu, Q. Sun, Z. Sun, Z. Wu, M. Dong, M. Zheng, X. Yin, and Y. Zhu. From Agent Traces to Trust: A Survey of Evidence Tracing and Execution Provenance in LLM Agents. arXiv:2606.04990, June 2026.

[16] S. Jamshidi, A. Moradi Dakhel, K. W. Nafi, and F. Khomh. Hallucination Cascade: Analyzing Error Propagation in Multi-Agent LLM Systems. arXiv:2606.07937, June 2026.

[17] D. Liu. Silent Failure in LLM Agent Systems: The Entropy Principle and the Inevitable Disorder of Autonomous Agents. arXiv:2606.08162, June 2026.

[18] J. C. Knight and N. G. Leveson. An Experimental Evaluation of the Assumption of Independence in Multiversion Programming. IEEE Transactions on Software Engineering, SE-12(1):96–109, 1986.