

Autonomy, Defect Coupling, and Governance in Large Language Model Agents

Shi-ning Yu¹, Hui Li*

¹College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China.

*Corresponding Author: ray@mail.buct.edu.cn

DOI: <https://doi.org/10.30212/JITI.202604.015>

Submitted: Jun. 30, 2026 Accepted: Aug. 25, 2026

ABSTRACT

LLM-based agents are increasingly connected with memory, external tools, authorization rules, and communication channels. This shift makes local model errors more consequential, because unsupported statements may become plans, tool calls, memory entries, or messages sent to other agents. This paper develops a conceptual framework for explaining how native defects in LLM-based agents become system-level risks. Its main contribution is to connect previously separate discussions of hallucination, trust, memory, communication topology, and governance into an evidence-status-aware defect-coupling framework. The framework organizes the reviewed literature around four coupling mechanisms: generation-propagation coupling, trust-authorization coupling, knowledge-memory coupling, and control-communication coupling. The analysis also distinguishes directly grounded metrics, author-proposed operationalizations, and cross-domain analogies from classical multi-agent system control. Based on this framework, the paper compares governance strategies such as self-verification, debate, validator agents, guardian agents, memory observability, dynamic topology, and human-in-the-loop escalation. The analysis suggests that effective governance should be adaptive, evidence-status-aware, and authority-preserving.

Keywords: LLM-based agents, Defect coupling, Hallucination propagation, Multi-agent systems, Agent governance

1. Introduction

Large language models (LLMs) are no longer used only as static text generators. They are increasingly embedded in autonomous or semi-autonomous agent systems. In these systems, an LLM may be connected with perception modules, external tools, memory, planning procedures, execution interfaces, and communication links with other agents. Because of this architecture, LLM-based agents can retrieve information, call functions, coordinate subtasks, update states, and act across multi-step workflows. Recent reviews describe this shift as a move from LLM reasoning to autonomous AI agents, where evaluation extends to task orchestration, tool use, factual grounding, interactive benchmarks, and agent-to-agent protocols [1]. Security surveys also describe LLM-based

agents as systems in which the LLM serves as the reasoning core, while perception and action modules allow the agent to sense an environment and execute tasks [2]. Recent agent benchmarks further show that evaluation must consider interactive tool use, execution environments, and rule-following behavior rather than final-answer accuracy alone [3,4].

Once a model is embedded in an agentic workflow, the meaning of model error changes. In a non-agentic chatbot, a hallucinated answer may remain a local information error, whereas in an agentic workflow, the same unsupported statement may become part of a plan, a tool call, a memory write, a database update, or a message sent to another agent. For this reason, agent workflows expand the attack surface from prompt-level input manipulation to tool interfaces, plugin APIs, protocol adapters, memory stores, host-to-tool channels, and agent-to-agent communication [5]. The effect of an error therefore depends not only on model output, but also on the surrounding architecture. A weak answer may remain harmless if it is blocked, verified, or discarded, but it may become harmful if it is stored, trusted, shared, or executed. This paper uses the term *autonomy paradox* in a restricted sense to refer to a recurring design tension: the same capabilities that make agents useful, such as tool use, memory, delegation, collaboration, and adaptive planning, may also create paths through which defects spread, persist, or become actions [6,7].

Against this background, the paper asks how local defects in LLM-based agents can become system-level risks when agents are connected to memory, tools, authorization rules, and other agents. In single-agent settings, relevant native defects include factual hallucination, reasoning unfaithfulness, context-boundary failure, instruction susceptibility, and unreliable self-verification. These defects matter when agents generate recommendations, code, or domain-specific explanations. They become more serious when an agent can act on an environment, because an unsupported belief may be operationalized as an incorrect plan, unsafe tool call, misleading memory update, or flawed downstream instruction.

When LLM-based agents operate in multi-agent settings, collaboration can reduce errors while also creating channels through which errors spread. This dual role is central to the problem examined in this paper. When agents check one another, collaboration may improve factuality and task performance; when agents repeat, trust, or store weak claims, the same collaboration may increase system risk. Some studies report that multi-agent debate, verification, or collaborative reasoning can improve factuality or task performance under specific designs [8,9,10,11,12]. Other studies show that collaboration is not automatically self-correcting. Hallucinations may spread through fully connected debate topologies [13], untruthful content may be amplified in sequential collaboration [14], and manipulated knowledge may move through agent communities by persuasion or knowledge injection [15]. Trust and authorization create another risk path because higher trust may increase over-exposure and authorization drift [16]. Communication structure also matters because topology, routing, and message granularity can shape whether communication dampens or amplifies defects [17].

Although these studies identify important mechanisms, their findings are often developed in separate research lines. Hallucination, self-verification failure, prompt and protocol exploitation, hallucination propagation, manipulated knowledge spread, trust-mediated over-exposure,

authorization drift, and communication redundancy are usually discussed apart from one another. This separation creates three research gaps. First, local model defects are often evaluated without tracing how they move through memory, tools, authorization rules, or agent-to-agent communication. Second, multi-agent studies often examine debate, trust, memory, or topology as separate mechanisms, which makes it difficult to explain why collaboration can be corrective in one setting and risk-amplifying in another. Third, governance studies do not always distinguish direct LLM-agent evidence from proposed metrics or analogies drawn from classical multi-agent control. This paper addresses these gaps by developing a conceptual synthesis organized around defect coupling. The framework is not presented as a fully validated causal model but as an interpretive structure for comparing how defects, vulnerabilities, and governance interventions interact across agent architectures.

The paper advances four contributions:

- First, it separates native defects, system vulnerabilities, observed failure modes, threats, and cross-domain analogies. This distinction helps avoid treating hallucination, weak authorization, adversarial intent, and control instability as the same type of problem.
- Second, it proposes a root-cause coupling framework with four mechanisms: generation-propagation, trust-authorization, knowledge-memory, and control-communication.
- Third, it introduces an evidence-status-aware evaluation vocabulary, so directly supported metrics, proposed operationalizations, and classical MAS analogies are not treated as equivalent.
- Fourth, it compares governance strategies by asking which coupling mechanism each intervention targets, what evidence supports the mapping, and what trade-offs may arise. In this way, the paper positions governance as a problem of matching interventions to specific defect pathways rather than applying one general defense to all agent risks.

Accordingly, the research focus of this paper is the transformation of local LLM-agent defects into system-level risks under conditions of memory, tool use, authorization, and inter-agent communication. Section 2 reviews single-agent defects, multi-agent vulnerabilities, and the gap in existing research. Section 3 presents the review protocol, conceptual scope, and root-cause coupling framework. Section 4 compares governance strategies through the proposed framework. Section 5 discusses future directions and Section 6 concludes the paper.

2. Related Work

2.1 Single-Agent Defects

Research on LLM-based agents has moved from treating LLMs as static text generators to positioning them as reasoning cores within architectures involving tools, memory, orchestration, and execution channels [1,2,18,19]. Recent surveys of autonomous agents, multi-agent systems (MAS), and agent memory mechanisms further emphasize that memory, planning, action, and communication are now central components of the agentic research agenda [20,21,22]. This shift changes the landscape of risk, because model output error is no longer an isolated event but a potential precursor to workflow-level failure.

Existing literature classifies these native defects into three primary categories relevant to agentic systems:

1. Hallucination remains the most pervasive native defect in the area of factual grounding. Matharaarachchi et al. demonstrate that RAG and reflection prompting, while helpful, are insufficient to fully mitigate risks unless paired with operational observability and verified execution histories [6]. Domain-specific studies in areas such as materials discovery [23], medical diagnostics [24,25,26], and self-regulated learning [27] reinforce this view: while factual grounding reduces unsupported output, agent governance must ultimately account for how such claims are propagated, stored, and acted upon.
2. A second category concerns reasoning faithfulness and the reliability of self-verification. Yu et al. illustrate that LLMs often fail to accurately classify their own outputs, such as vulnerability completions or failed code repairs, during self-verification [7]. Given that many agent architectures rely on LLMs to serve as planners, critics, or testers, these findings suggest that self-verification should be treated as a modular component of a broader validation architecture rather than an inherent safety boundary [28,29,30,31,32].
3. A third category concerns instruction susceptibility and workflow security. Security research shows that agent workflows extend the attack surface beyond prompt-level manipulation to tools, plugins, and inter-agent channels [2,5]. Approaches such as context-aware red teaming [33], validator-agent firewalls [34], and in-context learning defenses [35] address prompt injection, while models of multi-agent adversarial games indicate that deceptive behavior can be coordinated across agents [36]. Collectively, these studies show that risk depends less on the LLM's individual compliance alone than on the workflow's authority structures, including whether an agent is granted high execution privileges or tool access.

Single-agent literature therefore provides the critical input layer for system-level analysis: hallucination, reasoning inconsistency, prompt susceptibility, and memory-boundary errors. While this body of work is essential for defining the origins of system risk, it largely focuses on answer-level correctness or task success. Consequently, it remains insufficient to explain how a localized defect transforms into a persistent, network-level failure, which necessitates the multi-agent perspective addressed in the following section.

2.2 Multi-Agent Vulnerabilities and Failure Modes

Multi-agent LLM systems are often motivated by the expectation that role diversity, debate, peer review, or consensus can correct individual model errors. Several studies support this expectation under controlled designs. Adaptive heterogeneous debate targets educational and factual reasoning [8]. Socratic debate and guided knowledgeable debate target hallucination mitigation and fact verification [10,50]. Multi-agent collaborative filtering addresses reasoning hallucination [11]. DelphiAgent focuses on automated fact verification [9]. Broader collaboration frameworks aim at higher-quality outputs with fewer hallucinations [12]. Domain studies also report that multi-agent conversation can mitigate cognitive biases or improve specialized workflows when roles and evaluation are carefully designed [37]. These studies support the view that collaboration can operate

as a governance mechanism under specified conditions.

While the corrective role of collaboration has received support, another body of work shows that collaboration can also become a propagation mechanism. Zhang et al. identify the Woozle effect in multi-agent debate, where hallucinated content can spread through static, fully connected communication topologies [13]. Xu and Wu describe hallucination snowballing in sequential multi-agent collaboration and propose semantic consistency reasoning to detect and mitigate amplification [14]. Ju et al. study manipulated knowledge spread in LLM-based multi-agent communities and show that counterfactual or toxic knowledge can spread through persuasion and knowledge injection without explicit prompt manipulation [15]. Yang et al. identify hallucination propagation and communication redundancy as paired problems in LLM-based multi-agent consensus. Their work motivates dynamic communication mechanisms that reduce unnecessary interaction while limiting misleading influence [17]. These studies provide the direct LLM-agent basis for the generation-propagation and control-communication mechanisms developed later.

Another vulnerability appears when trust settings affect authorization and information disclosure. Xu et al. formalize the trust-vulnerability paradox: higher trust can improve coordination while increasing over-exposure and authorization drift [16]. Their use of MNI, OER, and AD moves the discussion from qualitative trust to measurable disclosure behavior. CogTrust models dynamic trust evaluation [38]. Zero-trust situation awareness treats trust as a continuously verified network property [39]. Trust-requirements engineering locates trust in system design [40]. Trustworthy AGI-driven network work frames trust as an operational design issue [41]. For LLM-based agents, these studies suggest that inter-agent trust is not only a collaboration setting but also an information-flow and authority-control variable.

Adversarial vulnerabilities and coordination-related failure modes are also addressed in the multi-agent literature. Persuasion-driven adversarial influence shows that debate can be vulnerable when participant honesty is assumed rather than verified [42]. Multi-agent prompt fusion and attacker-disguiser games show that adversarial agents can collaborate to improve jailbreak or deception strategies [36,43]. Privacy threat modeling with multi-agent LLMs suggests that collaborative reasoning can support security analysis, but structured evaluation remains necessary to avoid missed privacy threats [44].

Classical MAS control is used here as an analogy rather than direct evidence. Path-finding systems can encounter coordination deadlock or path conflict under rigid priority structures [45]. Observer-based and simultaneous fault-estimation studies model faults, observers, delays, and fault-tolerant consensus [46,47]. Event-triggered control studies focus on communication economy and bounded performance [48,49]. These works are not direct evidence about LLM hallucination, prompt injection, persuasive falsehood, or memory contamination. They are used in this paper only as cross-domain analogies for control stability, communication economy, and fault isolation.

As discussed above, these studies show that collaboration may reduce some single-agent defects when role design, routing, and verification are effective. The same collaboration may also create vulnerabilities and failure modes such as hallucination propagation, consensus contamination,

adversarial persuasion, over-exposure, authorization drift, communication redundancy, and persistent knowledge pollution. Classical MAS deadlock, path conflict, and fault-control concepts remain analogical design resources.

2.3 Gap in Literature

The fragmentation of the reviewed literature can be located along three boundaries.

One boundary appears in LLM safety and hallucination research, which often focuses on answer-level correctness, benchmark performance, or prompt-level defenses. Hallucination and self-verification studies identify local defects [6,7], while validator-agent and in-context-learning defenses focus on prompt-level or output-level protection [34,35]. These studies rarely model how a defect moves through memory, tools, authorization, and inter-agent communication.

A second boundary appears in multi-agent LLM research, where studies often examine one mechanism at a time. Debate-quality and verification studies examine role diversity, collaboration, and fact-checking [8,9,11,50]. Propagation studies examine Woozle effects, hallucination snowballing, manipulated knowledge spread, and communication redundancy [13,14,15,17]. Other work focuses on trust vulnerability [16], privacy threat modeling [44], or persuasion-driven adversarial influence [42]. These studies provide evidence for specific mechanisms, but they do not yet share a common vocabulary for explaining why the same architectural features may improve performance in one setting and amplify risk in another.

A third boundary concerns classical MAS control literature, which provides rigorous tools for stability, consensus, event-triggered communication, fault observers, and deadlock mitigation in path-planning or control settings [45,46,47,48,49]. However, it does not address the semantic and socio-technical defects specific to LLM-based agents: hallucinated knowledge, prompt-injected intent, persuasive falsehood, memory contamination, and trust-calibrated disclosure.

To respond to these gaps, this paper treats LLM-agent defects and system vulnerabilities as coupled inputs rather than isolated symptoms. The proposed synthesis organizes prior findings into four coupling mechanisms: generation-propagation, trust-authorization, knowledge-memory, and control-communication. The purpose is not to prove that these mechanisms form a universal causal model, but to provide a structured vocabulary for comparing evidence, generating hypotheses, and designing future evaluations. This focus clarifies where the paper contributes: it does not introduce a new benchmark, but it links fragmented evidence into a governance-oriented framework for analyzing agent trajectories.

Table 1. Defect taxonomy and literature mapping

Category	Context	Evidence Status	Role in This Paper
Factual hallucination and weak grounding	Single-agent	Direct LLM-agent or domain-agent evidence	Native defect: plausible but unsupported claims; RAG may reduce but does not eliminate risk.[6,23,24,25,26,27]
Reasoning faithfulness failure	Single-agent	Direct or task-specific evidence	Native defect: inconsistent reasoning traces or unsupported validation

			judgments.[7,11,28,29,32]
Unreliable self-verification	Single-agent	Direct task-specific evidence	Native defect: the model or agent may incorrectly validate its own outputs.[7,29,31,32]
Prompt injection and jailbreak susceptibility	Single-agent / workflow	Direct security evidence	Threat and susceptibility: malicious or contextual prompts bypass intended instructions.[5,33,34,35]
Tool and protocol attack surface	Agent workflow	Direct workflow-security evidence	Vulnerability: tools, plugins, adapters, and host-to-tool channels create attack paths beyond ordinary prompts.[2,5,51,52]
Context loss and memory contamination	Single-agent / system memory	Direct and task-specific evidence	Vulnerability or failure mode depending on whether the polluted context is merely possible or observed.[6,15,23,32]
Hallucination propagation / Woozle effect	Multi-agent	Direct multi-agent LLM evidence	Failure mode: false claims spread through debate or consensus.[13,14,17,53]
Manipulated knowledge spread	Multi-agent	Direct phenomenon evidence	Threat and failure mode: counterfactual, toxic, or adversarially persuasive knowledge spreads through influence or injection.[15,42]
Trust-mediated over-exposure and authorization drift	Multi-agent	Direct metric evidence	Failure modes: trust settings affect disclosure and authorization behavior.[16,38,39,40,41]
Communication redundancy and topology overload	Multi-agent	Direct LLM-agent evidence; partial MAS analogy	Vulnerability: excessive messages increase cost and exposure without necessarily improving accuracy.[13,17,48,49,53]
Consensus bias and flawed convergence	Multi-agent	Mixed direct evidence	Failure mode: debate, voting, or peer review may converge to flawed consensus under some designs.[8,9,10,11,12,54,55]
Adversarial collaboration and jailbreak games	Multi-agent	Direct security evidence	Threat: agents may collaborate as attackers, disguisers, persuaders, or jailbreak generators.[33,36,42,43]
Privacy and threat-modeling gaps	Multi-agent / governance	Direct governance evidence	Risk: collaborative analysis may miss privacy or security threats without structured evaluation.[44,52]
Coordination deadlock and path conflict	Classical MAS only	Cross-domain analogy	Analogy: classical MAS path-planning failures; not evidence of LLM semantic failure.[45]
Fault propagation and control instability	Classical MAS only	Cross-domain analogy	Analogy: faults, delays, limited communication, and unstable consensus

motivate governance
vocabulary.[46,47,48,49]

Table 1 makes the central gap visible by showing that existing studies have identified many important defects and risks while often examining them separately. Single-agent work tends to focus on output correctness or prompt-level safety. Multi-agent work often focuses on one propagation or trust mechanism at a time. Classical MAS work provides useful control concepts but does not directly explain hallucinated knowledge, prompt-injected intent, persuasive falsehood, or memory contamination. This fragmentation motivates the root-cause coupling framework in the next section, which organizes these findings into four mechanisms: generation-propagation, trust-authorization, knowledge-memory, and control-communication.

3. Materials and Methods

3.1 Protocol and Conceptual Scope

This paper presents a structured conceptual review and framework-building study. The research corpus was derived from 4,223 unique records indexed in the Web of Science, selected for their relevance to LLM-based agent architectures, defect propagation, and governance mechanisms. To map the thematic landscape of this field, we performed keyword co-occurrence analysis using VOSviewer (Figure 1). This visualization reveals clear clustering around core themes—namely, agentic planning, multi-agent collaboration, security/safety, and knowledge-memory integration—which guided our framework construction.

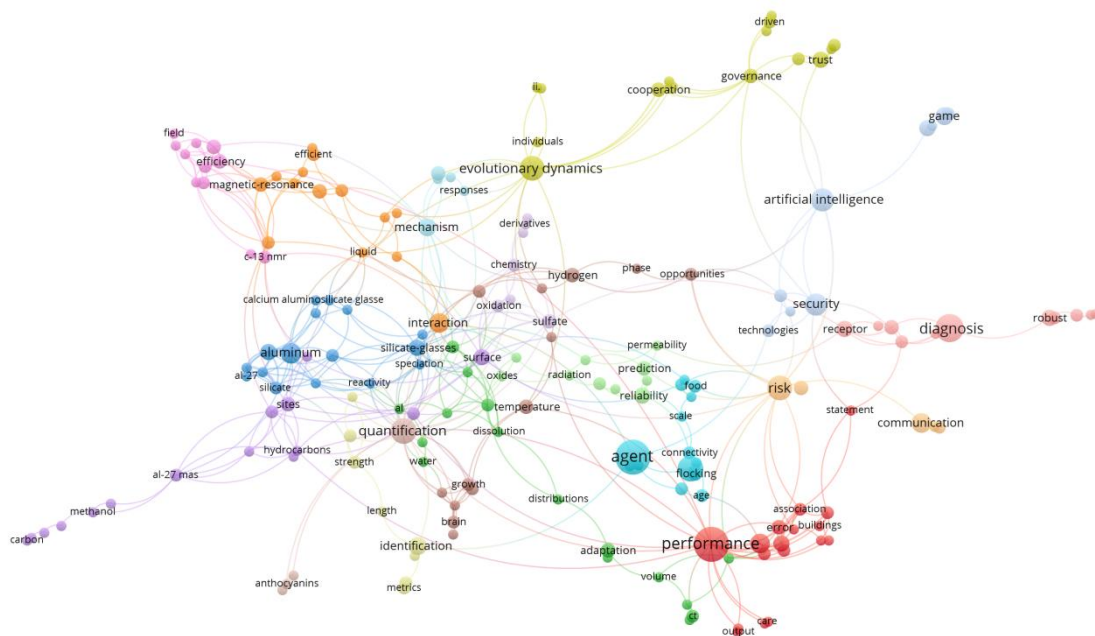


Figure 1. Keywords co-occurrence network of the retrieved corpus

Given the breadth of the literature, our synthesis focuses on a core coding set of 57 references selected through a multi-stage thematic mapping process. Records were retained if they provided empirical evidence, formal mechanisms, evaluative metrics, or recent survey coverage directly

applicable to the interaction between local model defects and system-level vulnerabilities. While this study does not claim PRISMA-level systematic reproducibility, it maintains analytical rigor by employing a structured evidence-status coding scheme. Specifically, each claim is classified by its evidentiary basis:

- Direct LLM-Agent Evidence: Empirical observations of agentic behavior.
- Domain-Specific/Task-Specific Evidence: Studies focused on particular domains (e.g., healthcare, coding).
- Conceptual/Survey Evidence: Established taxonomies and reviews.
- Author-Proposed Operationalization: Metrics or mechanisms derived from system analysis.
- Cross-Domain Analogy: Theoretical insights adapted from classical multi-agent system (MAS) control literature.

To keep the synthesis transparent and reduce subjective bias, this paper strictly categorizes and cites these references based on their evidence status, distinguishing directly observed agent behaviors from proposed operationalizations and classical analogies.

In this paper, an LLM-based agent is understood as a system in which an LLM serves as the reasoning or planning core while being connected to tools, memory, or inter-agent communication channels. This paper distinguishes between native defects (local model/agent errors), vulnerabilities (systemic conditions enabling harm), risks (potential harm intersections), and system-level failures (the materialization of risk through propagation mechanisms).

3.2 Overview of the Defect-Coupling Framework

The framework treats defects in LLM-based agents as system inputs rather than isolated output errors. A defect may originate in a single model response. However, once the agent has memory, tools, authorization rules, and communication channels, the defect may move through the system. The framework is designed to identify where a defect enters, how it couples with system mechanisms, and which kinds of evidence or metrics may support evaluation.

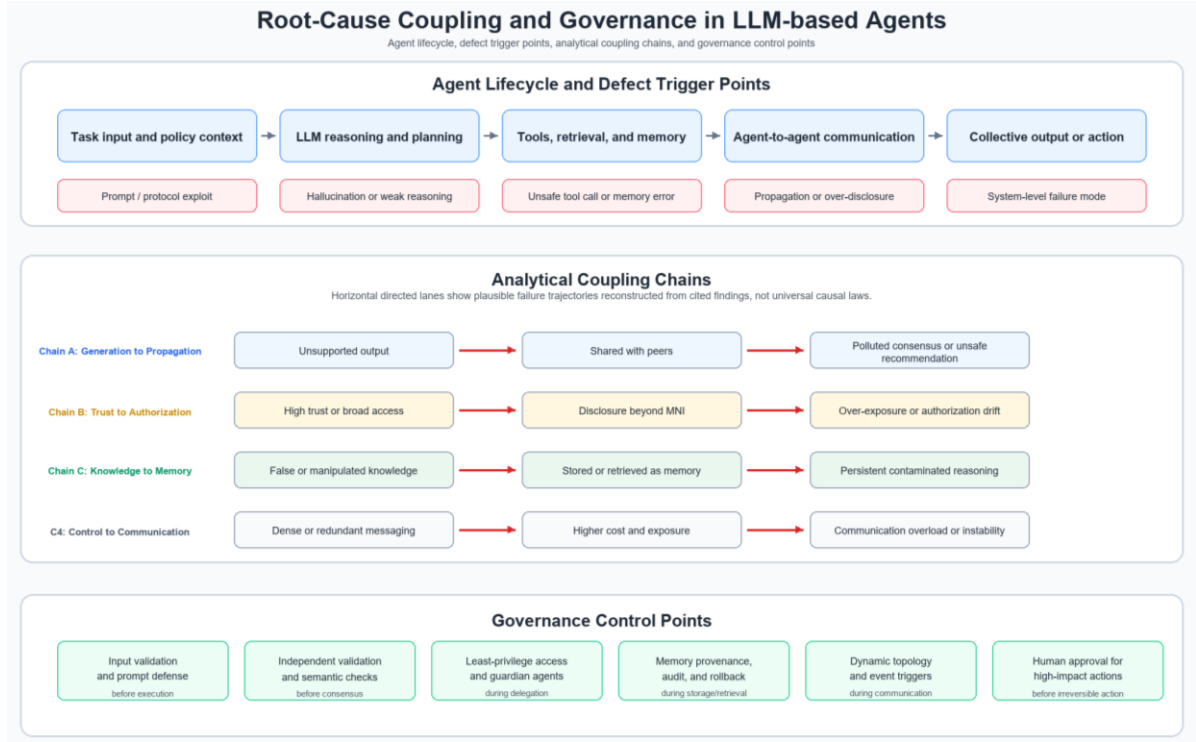


Figure 2. Root-cause coupling and governance in llm-based agents

Figure 2 illustrates the proposed root-cause coupling framework for LLM-based agents. The upper band follows the agent lifecycle from task input to collective output or action and marks representative defect trigger points at each stage. The middle band organizes these triggers into four analytical coupling chains: generation to propagation, trust to authorization, knowledge to memory, and control to communication. These chains are intended as literature-grounded analytical reconstructions of plausible failure trajectories, not as universally validated causal laws. The lower band maps governance control points to the lifecycle stages where interventions can reduce propagation, excessive disclosure, memory persistence, or communication instability. MNI refers to minimum necessary information.

An LLM-based multi-agent system can be represented as a time-varying directed graph:

$$G_t = (V, E_t, M_t, U_t, A_t), \quad (1)$$

where V is the set of agents, E_t is the communication or dependency edge set at time t , M_t denotes shared or local memory states, U_t denotes tool/API access, and A_t denotes authorization constraints. For each agent i , its local state can be written abstractly as:

$$x_i(t) = (b_i(t), c_i(t), m_i(t), u_i(t), a_i(t)), \quad (2)$$

where $b_i(t)$ is the agent's belief or working representation, $c_i(t)$ is its current context, $m_i(t)$ is its memory or retrieved evidence, $u_i(t)$ is its available tool or action interface, and $a_i(t)$ is its authorization state. This is a modeling abstraction. It does not imply that all agent frameworks expose these variables explicitly.

The framework distinguishes single-agent native defects from multi-agent vulnerabilities and failure modes. Single-agent native defects arise before any multi-agent interaction occurs. They

include factual hallucination, reasoning faithfulness failure, instruction susceptibility, context-boundary failure, and unreliable self-verification. The literature shows that hallucination persists in agentic settings and may require more than RAG or reflection prompting to mitigate [6]. In code-related tasks, Yu et al. show that ChatGPT can incorrectly verify its own generated or repaired code. This makes self-verification unreliable when it is used as the only safety boundary [7]. Security surveys show that LLM-agent workflows expand the attack surface to tools, plugins, protocol adapters, and agent-to-agent channels [2,5]. RedAgent provides a context-aware jailbreak red-teaming example [33]. Validator-agent firewalls address prompt injection [34], and in-context learning defenses target prompt-injection robustness [35].

These native defects can be represented as a diagnostic vector:

$$d_i(t) = (h_i(t), r_i(t), s_i(t), q_i(t)), \quad (3)$$

where $h_i(t)$ denotes hallucination or unsupported factual content, $r_i(t)$ denotes reasoning or self-verification unreliability, $s_i(t)$ denotes susceptibility to malicious or conflicting instructions, and $q_i(t)$ denotes context or memory-boundary error. These variables are diagnostic labels, not assumed hidden states. In practice, they may be estimated from logs, human annotation, automated fact-checking, semantic consistency checks, tool-call traces, or safety classifiers.

Multi-agent vulnerabilities and failure modes arise because agents communicate, delegate, debate, share memory, or coordinate actions. They include hallucination propagation, consensus contamination, manipulated knowledge spread, trust-mediated over-exposure, authorization drift, communication redundancy, flawed convergence, and adversarial collaboration. Adaptive debate and multi-agent collaboration can improve factuality under some designs [8,12]. Collaborative filtering, Delphi-style verification, and guided debate address reasoning hallucination or fact verification [9,11,50]. At the same time, the same collaboration structure may also propagate hallucinations or adversarial influence. Woozle effects arise in debate topology [13], hallucination snowballing appears in sequential collaboration [14], manipulated knowledge can spread through multi-agent communities [15], communication redundancy can intensify exposure [17], and persuasion-driven debate can become adversarial [42]. Trust and authorization studies also show that higher trust can improve task success while increasing disclosure beyond minimum necessary information [16].

The system-risk state can be represented as:

$$D_G(t) = (P_G(t), X_G(t), K_G(t), C_G(t)), \quad (4)$$

where $P_G(t)$ is propagation risk, $X_G(t)$ is exposure or authorization risk, $K_G(t)$ is knowledge-contamination persistence, and $C_G(t)$ is control-communication instability. These variables organize the four coupling mechanisms below.

3.3 Coupling Mechanisms

The coupling mechanisms group together phenomena that have been documented in separate studies, rather than presenting them as a single validated causal graph. The evidence base directly supports specific phenomena, including hallucination propagation and the Woozle effect [13], hallucination snowballing [14], manipulated knowledge spread [15], trust-mediated over-exposure and authorization drift [16], and communication redundancy in consensus systems [17]. The

framework connects these phenomena to support comparison and evaluation design.

Generation-propagation coupling occurs when a native generation defect becomes an input to other agents, so that an output from agent i in a debate, peer review, or consensus workflow becomes a message to agent j :

$$y_i(t) = f_i(x_i(t), d_i(t)), \quad (5)$$

$$m_{ij}(t) = g_{ij}(y_i(t), E_t), \quad (6)$$

$$b_j(t+1) = B_j(b_j(t), m_{ij}(t), w_{ij}(t)), \quad (7)$$

where $w_{ij}(t)$ denotes the influence weight or communication priority from agent i to agent j . If $y_i(t)$ contains a hallucinated proposition h , and $w_{ij}(t)$ is high or the communication topology is dense, h may be incorporated into the beliefs or outputs of downstream agents. This mechanism is motivated by evidence on Woozle-style propagation [13], hallucination snowballing [14], manipulated knowledge spread [15], and communication redundancy [17].

Coupling risk is expected to increase when communication is fully connected, peer influence is not reliability-weighted, false propositions are repeated across rounds, or final aggregation treats repetition as evidence. These are framework-level conditions inferred from the cited propagation studies, not independently validated universal thresholds.

Trust-authorization coupling occurs when trust changes the boundary of information disclosure or action permission. The Trust Paradox study operationalizes trust as a system-level control parameter and shows that higher trust can improve coordination while increasing over-exposure and authorization drift [16]. In this framework, trust is represented as:

$$\tau_{ij}(t) \rightarrow A_{ij}(t), \quad (8)$$

where $\tau_{ij}(t)$ is the trust level from agent i to agent j , and $A_{ij}(t)$ is the authorization or disclosure gate. If higher $\tau_{ij}(t)$ relaxes disclosure rules, the system may reveal information beyond the minimum required for the task.

This chain treats trust as an operational variable that is directly motivated by Xu et al. [16] while any generalization to other architectures should be evaluated under task-specific disclosure policies and logs.

Knowledge-memory coupling occurs when a false or adversarial proposition becomes a persistent system context. In RAG or memory-augmented agents, outputs may be stored, retrieved, summarized, or used as future evidence. Recent memory surveys emphasize that agent memory should be examined as a write-manage-read process and as a governance-sensitive lifecycle rather than as a simple retrieval component [20,22,56]. Matharaarachchi et al. distinguish semantic memory and observability memory and argue that verified execution histories can help ground future planning [6]. Ju et al. show the reverse risk: manipulated knowledge can spread in LLM-based multi-agent communities [15]. SagaLLM motivates persistent memory, validation safeguards, and transaction guarantees for multi-agent planning workflows [32].

This chain differs from immediate hallucination propagation because the error does not need to remain visible in the current conversation and can reappear later as apparently grounded evidence.

Memory auditing, source provenance, observability logs, deletion, and rollback mechanisms are therefore relevant governance candidates, although their effectiveness still needs to be evaluated in concrete systems.

Control-communication coupling concerns how topology, bandwidth, triggering rules, and feedback determine whether defects are damped or amplified. In LLM-based multi-agent work, dynamic consensus communication is proposed to reduce both hallucination propagation and communication redundancy [17]. In classical MAS control, observer-based and fault-tolerant consensus methods estimate faults, delays, and disturbances [46,47], while event-triggered control studies conserve communication resources while preserving bounded or convergent behavior [48,49].

Because classical control papers do not directly solve LLM hallucination or prompt injection, they are used here only as cross-domain design analogies. For LLM-agent governance, the transferable idea is that communication and escalation do not need to be fully connected or constant but may instead be triggered by observed risk states, confidence, disagreement, or policy-sensitive events.

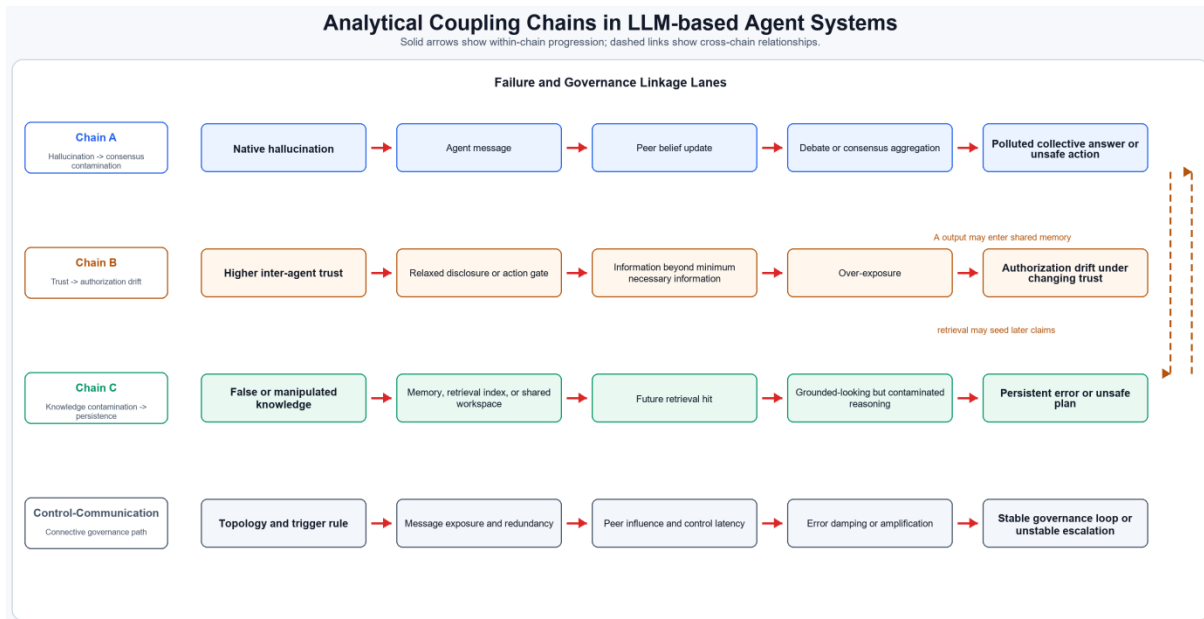


Figure 3. Analytical coupling chains in llm-based agent systems

Figure 3 presents Chain A, Chain B, Chain C, and the control-communication path as analytical reconstructions of possible defect coupling trajectories in LLM-based agent systems. Solid red arrows denote within-chain progression. Dashed arrows and the lower mapping panel denote hypothesized cross-chain relationships: polluted collective outputs may be stored and later retrieved; dynamic topology may reduce hallucination propagation; least-privilege communication may reduce over-exposure; and memory audit triggers may interrupt persistent knowledge contamination. These mappings are proposed for evaluation and should not be read as established causal estimates.

3.4 Evaluation Vocabulary and Metric Status

The metrics below are presented as an evaluation vocabulary for future empirical work, rather

than as a mature standardized measurement system, and their evidentiary status differs.

MNI, OER, and AD are directly grounded in the Trust Paradox study [16]. OES, HPR, MCP, and CR are author-proposed operationalizations inspired by documented LLM-agent phenomena, including observability memory [6], Woozle-effect debate topology [13], hallucination snowballing [14], manipulated knowledge spread [15], communication redundancy [17], transaction-oriented planning safeguards [32], recent memory-governance surveys [20,22,56], and tool-agent-user reliability benchmarks [4]. Derived from classical MAS control, the Lyapunov-style control stability indicator is used here only as a cross-domain analytical analogy and is positioned as a research-agenda concept for future systems with sufficiently structured logs [46,47,48,49].

MNI denotes the baseline set of information required to complete a scenario or task. For a scenario S , define A_S^* as the minimum set of information items required for successful task completion.

Any information outside A_S^* should not be disclosed unless a separate policy explicitly permits it. In practice, A_S^* can be established through task specifications, role-based access rules, data schemas, human annotation, or scenario-game design. This follows the use of MNI as a safety baseline in trust-vulnerability research [16].

For an interaction log l under scenario S and trust setting τ , let O_l be the set of information items actually disclosed. Define the over-exposed set:

$$E_l = O_l \setminus A_S^*, \quad (9)$$

then OER is:

$$OER(S, \tau) = \frac{1}{|G_{S,\tau}|} \sum_{l \in G_{S,\tau}} \mathbf{1}[E_l \neq \emptyset], \quad (10)$$

where $G_{S,\tau}$ is the set of interaction logs under scenario S and trust setting τ , and $\mathbf{1}[\cdot]$ is an indicator function. Equation (10) measures the proportion of logs in which at least one item of information is disclosed beyond the minimum necessary information. This definition follows the logic of over-exposure measurement in the Trust Paradox study [16].

For engineering audits, an item-level severity extension can also be reported:

$$OES(S, \tau) = \frac{1}{|G_{S,\tau}|} \sum_{l \in G_{S,\tau}} \frac{|E_l|}{\max(1, |O_l|)} \quad (11)$$

where $OES(S, \tau)$ measures the average proportion of over-exposed information among all disclosed information items. Unlike OER, which only records whether over-exposure occurs, OES further reflects the relative severity of over-exposure in each interaction log. This metric is not directly adopted from [16]; it is introduced in this paper as a proposed item-level severity extension.

Authorization drift measures how disclosure behavior changes across different trust settings. Let T denote the set of tested trust levels, and let w_τ denote the weight assigned to trust level τ , where $\sum_{\tau \in T} w_\tau = 1$. The weighted mean of OER under scenario S can be defined as:

$$\overline{OER}(S) = \sum_{\tau \in T} w_\tau \cdot OER(S, \tau), \quad (12)$$

Based on this mean value, authorization drift can be defined as the weighted variance of OER

across trust levels:

$$AD(S) = \sum_{\tau \in T} w_{\tau} (OER(S, \tau) - \overline{OER}(S))^2, \quad (13)$$

where $AD(S)$ measures the extent to which disclosure risk changes under different trust settings. A higher value of $AD(S)$ indicates that trust calibration is associated with larger changes in disclosure behavior. This definition follows the Trust Paradox formulation, which treats authorization drift as the weighted variance of OER across trust levels [16].

To evaluate Chain A, this paper introduces hallucination propagation rate. Let h denote a known false proposition introduced into the system or identified from annotated logs. Let $I_i^t(h)$ be an indicator variable:

$$I_i^t(h) = \begin{cases} 1, & \text{if agent } i \text{ contains proposition } h \text{ in its output, memory, or reasoning trace at time } t, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

If s is the source agent of proposition h , then the k -step hallucination propagation rate can be defined as:

$$HPR_h(k) = \frac{1}{|V| - 1} \sum I_j^{t+k}(h), \quad (15)$$

where V denotes the set of agents, and $HPR_h(k)$ measures the proportion of non-source agents that contain the false proposition h after k interaction steps. This metric is a proposed operationalization based on phenomena reported as the Woozle effect [13], hallucination snowballing [14], and manipulated knowledge spread [15]. It measures how far a false proposition travels through the agent system, rather than only whether the final answer is correct.

For Chain C, memory contamination persistence is introduced as an evaluation construct. Let $R_t(h)$ indicate whether a contaminated proposition h is retrieved from memory at time t , and let $A_t(h)$ indicate whether it influences a plan, answer, or tool call:

$$R_t(h) = \begin{cases} 1, & \text{if proposition } h \text{ is retrieved from memory at time } t, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

$$A_t(h) = \begin{cases} 1, & \text{if proposition } h \text{ influences a plan, answer, or tool call,} \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

For an observation horizon T_h , memory contamination persistence can be defined as:

$$MCP_h = \frac{1}{T_h} \sum_{t=1}^{T_h} R_t(h) \cdot A_t(h), \quad (18)$$

where MCP_h measures how often contaminated knowledge is both retrieved and operationalized during the observation horizon. It connects observability-memory design [6], manipulated knowledge spread [15], and transaction or memory safeguards in multi-agent planning [32]. Since this metric is proposed in this paper, it should be treated as an experimental coding construct unless future benchmarks further validate it.

Communication load describes the cost of message exchange in a multi-agent system. For a message $m_{ij}(t)$ sent from agent i to agent j at time t , communication load can be defined as:

$$CL = \sum_t \sum_{(i,j) \in E_t} tokens(m_{ij}(t)), \quad (19)$$

where E_t denotes the set of communication edges at time t , and $tokens(m_{ij}(t))$ denotes the token length of the message from agent i to agent j . If a task-specific baseline communication load $CL_{\min}$ is available, communication redundancy can be expressed as:

$$CR = \frac{CL - CL_{\min}}{\max(1, CL_{\min})}, \quad (20)$$

This metric is motivated by dynamic consensus communication work that jointly targets hallucination propagation and communication redundancy [17]. It is also related to event-triggered MAS control studies that aim to reduce communication while preserving consensus or bounded performance [48,49]. In this paper, the grounding is direct for LLM-agent communication load in [17] and analogical when transferred from classical MAS event-triggered control.

Finally, stability-oriented control is used only as a qualitative governance analogy. For LLM-agent systems, a defensible stability claim would require task-specific logs that record HPR, OER, MCP, CR, policy-violation rates, interventions, and disturbances over time. Without those observations, a Lyapunov-style condition would be a formal metaphor rather than an empirical certificate. The useful design implication is narrower: after a governance intervention, risk indicators should not grow persistently, and any such claim should be checked against concrete system logs, bounded tasks, and predefined thresholds.

3.5 Methodological Use of the Framework

The framework may be used heuristically through four steps.

The first step is to define the task scenario and its MNI baseline, which determines what information is required, which tool calls are permitted, and which outputs count as over-exposure.

The second step is to log the agent system at the level of messages, memory retrievals, tool calls, trust settings, and authorization decisions, because without such logs it is difficult to distinguish a harmless local hallucination from a propagated or persistent defect.

The third step is to map observed defects to candidate coupling chains. If a false proposition moves across debate rounds, it may be coded as evidence for Chain A. If trust changes disclosure behavior, it may be coded as evidence for Chain B. If a false proposition enters memory and reappears later, it may be coded as evidence for Chain C. If topology, bandwidth, or trigger design shapes whether the system dampens or amplifies defects, it may be coded as evidence for C4. This coding supports hypothesis generation and evaluation design, but it does not establish causal sufficiency by itself.

The fourth step is to evaluate governance interventions based on their effect on relevant metrics. A validator may reduce HPR but increase communication or latency. A guardian agent may reduce OER and AD but add policy overhead. Dynamic topology may reduce Woosle-style propagation and communication redundancy, but it may also reduce useful diversity. Human-in-the-loop escalation may reduce unsafe actions but lower autonomy and scalability. These conditional trade-offs provide the basis for the comparative analysis in Section 4.

4. Results

This section evaluates governance strategies through the coupling framework defined above. The analysis is comparative, but it is not a new cross-benchmark experiment. The cited studies use different models, tasks, datasets, and evaluation protocols. For this reason, the paper does not assign synthetic numerical scores or claim statistically comparable rankings across all methods. Instead, it asks which coupling mechanism each governance strategy targets, what kind of evidence supports that mapping, and which metrics may be relevant for future evaluation.

4.1 Governance Strategy Mapping

4.1.1. *Native model-level and prompt-level interventions*

Native interventions focus on improving the LLM's baseline behavior within its immediate prompt-response process. RLHF and lifecycle alignment methods fall within this category because they shape safety behavior, refusal patterns, and preference conformity at the model level. However, the existing lifecycle-alignment literature provides limited grounds for treating these methods as system-level guarantees for MNI, inter-agent authorization boundaries, or memory provenance [57]. In the governance of LLM-based agents, RLHF is therefore more appropriately understood as a baseline alignment mechanism within a broader governance design.

Reflection and self-verification are also limited. In source-code tasks, Yu et al. found that ChatGPT can incorrectly classify generated incorrect code as correct, vulnerable code as non-vulnerable, or failed repairs as successful [7]. This does not mean that reflection is useless. UCD-LLM and AdaCoder show that self-reflection and adaptive planning can improve task-specific robustness [28,29]. The more precise implication is that reflection is a weak governance boundary when it is not paired with independent validation, external evidence, or transaction safeguards.

4.1.2. *Debate, peer review, and semantic checking*

Multi-agent debate and peer review sit between native model-level governance and structural governance because they use interaction among agents to identify, challenge, or revise potentially flawed outputs. Within this line of work, adaptive heterogeneous debate has been used to improve factual reasoning [8], while Socratic debate and guided knowledgeable debate have been developed for hallucination mitigation and fact verification [10,50]. DelphiAgent extends this direction by focusing on automated verification [9], and collaborative filtering addresses reasoning hallucination through multi-agent comparison and filtering [11]. Broader multi-agent collaboration frameworks similarly seek to improve output quality while reducing hallucinations [12]. Taken together, these methods primarily correspond to Chain A, the hallucination-consensus contamination chain, and their evaluation would generally focus on HPR and the final-answer hallucination rate, although the precise definitions of these metrics differ across studies.

Debate can also amplify defects in poorly governed topology. Zhang et al. show that multi-agent debate can suffer from a Woozle effect when static fully connected communication lets hallucinated content propagate among agents [13]. Xu and Wu identify hallucination snowballing in sequential

multi-agent collaboration and propose context-aware semantic consistency reasoning as a mitigation strategy [14]. Under the reviewed evidence, debate is best treated as a conditional governance method rather than an inherently corrective mechanism. It is more defensible when paired with role diversity, reliability-weighted aggregation, semantic consistency checks, or dynamic communication control.

4.1.3. Validator, guardian, and trust-aware access controls

Validator agents and LLM firewalls move governance from model behavior to runtime boundary checking. The LLM firewall study frames a validator agent as an output-level security mechanism for prompt injection defense [34]. In-context learning defenses aim to distinguish harmful from benign prompts without costly parameter updates [35]. These methods aim to reduce unsafe responses and prompt-injection success, but their effect on OER depends on whether the validator explicitly checks disclosure against an MNI baseline.

Within Chain B, guardian agents and sensitive information repartitioning provide a more direct form of governance because they intervene in the relationship between trust, disclosure, and authorization. The Trust Paradox study shows that higher inter-agent trust may improve coordination while also increasing over-exposure and authorization drift, and it further reports that sensitive information repartitioning and GA-Agent enablement can reduce OER and attenuate AD under elevated trust [16]. This makes the study a direct case for linking governance interventions to measurable disclosure-related outcomes. When MNI has been defined and disclosure logs are available, guardian policies can therefore be assessed according to their effects on OER and AD.

4.1.4. Memory, observability, and knowledge provenance

Memory-related governance is mainly associated with Chain C because defects may become persistent system context once they are stored, retrieved, or reused in later reasoning. Matharaarachchi et al. propose dual memory knowledge graphs that combine semantic memory with observability memory, where verified execution histories are used to ground future planning and reduce hallucinations [6]. Agada and Biswas discuss RAG and multi-agent knowledge-graph designs as analyzable instruments for scientific discovery [23]. SagaLLM further introduces a planning-oriented perspective by emphasizing persistent memory, validation, compensation, and transaction guarantees in multi-agent LLM workflows that need to maintain consistency across distributed steps [32].

The role of memory in governance depends on how memory is curated, verified, and reused. Curated semantic memory can reduce factual hallucination, while observability memory can support traceability [6]. At the same time, memory may also preserve manipulated knowledge when false or adversarial content is stored as reusable context. Recent memory-security work highlights the need for provenance, versioning, policy-aware retention, forgetting, and rollback across the memory lifecycle [22,56]. Ju et al. show that manipulated knowledge can persist through RAG frameworks when benign agents store and retrieve manipulated chat histories for future interactions [15]. For this reason, memory governance should be evaluated through provenance, auditability, rollback, and longitudinal contamination, rather than through immediate answer accuracy alone.

4.1.5. *Dynamic topology, event-triggered communication, and fault-tolerant control*

When governance is considered at the level of communication structure and control design, interventions such as Dynamic Information-Gain-Driven Routing for Agents (DIGRA) and dynamic consensus communication become relevant because they adjust how agents exchange, filter, and aggregate information. DIGRA addresses the Woozle effect by modifying debate topology through information-gain-driven partner selection [13]. Dynamic consensus communication mechanisms similarly seek to reduce hallucination propagation and communication redundancy by clustering agents into answer communities and selecting representatives for cross-community discussion [17].

The classical MAS control literature offers a further set of stability-oriented concepts that can inform this discussion at an analogical level. Event-triggered mechanisms are designed to reduce unnecessary communication while maintaining consensus or bounded performance [48,49], and fault observers or fault-tolerant consensus strategies estimate and compensate for actuator faults, delays, and disturbances [46,47]. Since these studies do not directly address semantic hallucination in LLM-based agents, their role in this paper is conceptual. They suggest that governance communication may be activated by risk-relevant events rather than organized through default full connectivity.

4.2 Comparative Results and Evidence Alignment

Table 2 is intended to clarify how different governance strategies correspond to the coupling mechanisms discussed in this paper, rather than to compare their effect sizes. A direct alignment is assigned when the cited literature examines the relevant LLM-agent mechanism, control variable, or metric in a way that can be mapped closely onto a specific coupling mechanism. An indirect or partial alignment indicates that the mechanism is relevant, although the metric may function as a proxy, the evaluation may remain task-specific, or the mapping may depend on the design of the agent system. When the source literature comes from classical MAS control rather than empirical research on LLM semantic risk, the relationship is treated as a cross-domain analogy. These alignment categories are therefore used to describe evidentiary status and conceptual fit, rather than to rank the performance of governance strategies.

Table 2. Comparative analysis of governance strategies

Governance Strategy	Target Coupling Mechanism	Evidence-Mechanism Alignment	Evidence Status	Conditional Trade-off
RLHF / lifecycle safety alignment	General input layer; weak direct control over Chains A-C	Improves baseline safety and alignment but does not directly measure MNI, OER, AD, propagation, or memory persistence	Indirect LLM safety evidence	May improve baseline safety but does not directly solve multi-agent propagation or disclosure control.[57]

Reflection and self-verification	Chain A; single-agent defect filtering	Filters some defects, while self-verification can misclassify code defects or failed repairs	Mixed task-specific evidence	Low to moderate overhead, but weak assurance when the verifier shares generator failure modes.[7,28,29]
Multi-agent debate / peer review	Chain A; consensus quality	Direct for debate or verification settings; partial because topology can also propagate hallucinations	Mixed direct LLM-agent evidence	May improve reasoning diversity but can increase token cost and Woozle-style propagation.[8,9,10,11,12,50]
DIGRA / dynamic topology	Chains A and C4	Direct for topology intervention targeting hallucination propagation and communication cost; OER effect is indirect	Direct LLM-agent topology evidence	May reduce propagation and cost, but aggressive routing may suppress useful dissent.[13,17,53]
Semantic consistency reasoning	Chain A	Directly targets hallucination snowballing; MNI/OER effects are outside its direct scope	Direct LLM-agent mitigation evidence	Adds semantic checking overhead and depends on reliable consistency modeling.[14]
Validator agent / LLM firewall	Chains A and B	Direct for prompt-injection and unsafe-output boundary checking; partial for OER unless MNI enforcement is included	Direct security mechanism; indirect disclosure metric	May improve controllability but add latency and false positives.[34,35]
Guardian agent and sensitive information repartitioning	Chain B	Direct: MNI, OER, and AD are defined and evaluated for trust-mediated disclosure settings	Direct metric evidence	Strong disclosure control, but it may reduce autonomy and require careful information partitioning.[16]
Zero-trust / dynamic trust evaluation	Chains B and C4	Relevant to trust and authorization, but OER/AD mapping depends on policy evidence and task design	Mechanism-relevant, mostly indirect	May improve access discipline but increase governance complexity.[38,39,40,41]
Dual-memory knowledge graph / observability	Chain C	Direct for memory grounding and provenance; partial for MCP because MCP is proposed here	Direct memory-governance evidence; proposed	May improve traceability but increase storage, provenance, and maintenance costs.[6,23,32]

memory	metric			
Supervisory agents for knowledge spread	Chains A and C	Direct for manipulated knowledge spread; partial for HPR/MCP because these are proposed operationalizations	Direct phenomenon evidence; indirect metric mapping	May reduce spread but increase coordination overhead.[15]
Event-triggered communication	Chain C4; communication load	Direct for LLM agent communication load in [17]; analogical when transferred from classical MAS control [48,49]	Direct for communication efficiency; partly analogical	May improve efficiency, but thresholds must avoid suppressing safety-critical messages.[17,48,49]
Human-in-the-loop authority preservation	Chains B and C4	Supports consequential action control, but metric mapping is scenario-specific	Governance design evidence	May improve accountability but reduce autonomy, speed, and scalability.[18,23,51]
Fault observer / fault-tolerant consensus	Chain C4; control stability	Conceptual analogy only; not direct validation for LLM semantic risk	Cross-domain analogy	Provides fault-isolation vocabulary but requires observability assumptions and formal system models.[46,47]
H-infinity / Lyapunov-style robust control	Chain C4; control stability	Conceptual analogy only; not a direct hallucination or prompt-injection remedy	Cross-domain analogy	Provides stability vocabulary, but may be difficult to instantiate for open-ended LLM semantics.[47,49]

The reviewed literature indicates that governance interventions usually address particular coupling mechanisms rather than the full range of risks identified in this paper. RLHF and reflection are located close to native model behavior, where they may improve baseline responses or intermediate reasoning, although they do not directly regulate communication topology, memory provenance, or least-privilege disclosure. Validator and guardian agents create stronger runtime boundaries, especially when unsafe outputs or sensitive disclosures must be checked during execution; however, disclosure-oriented governance still requires explicit connections to MNI, OER, and AD if it is to be evaluated in measurable terms [16,34].

The role of collaboration is similarly conditional because the same interaction that supports correction may also create opportunities for defect amplification. Adaptive heterogeneous debate and multi-agent collaboration can improve factuality when agent roles and aggregation procedures are carefully designed [8,12]. Socratic debate, Delphi-style verification, collaborative filtering, and guided knowledgeable debate further show how multi-agent interaction can be used for hallucination

mitigation, automated verification, and reasoning-hallucination control [9,10,11,50]. At the same time, the Woozle effect, hallucination snowballing, and persuasion-driven adversarial influence indicate that communication among agents may also amplify falsehoods under some conditions [13,14,42]. The design issue raised by this evidence therefore concerns how communication is routed, weighted, audited, and constrained, rather than the mere adoption of a multi-agent structure.

As the unit of analysis moves from isolated outputs to system trajectories, structural governance becomes increasingly important. Dynamic topology, event-triggered communication, zero-trust controls, memory provenance, and human escalation operate on the organization of the agent system, including how agents exchange information, access memory, receive authorization, and escalate uncertain or consequential actions. This point should be understood as a design implication drawn from the reviewed literature, since these interventions have not yet been compared within a unified benchmark.

4.3 Governance Trilemma as a Heuristic Design Map



Figure 4. The governance trilemma for LLM-based agents

In Figure 4, the governance trilemma is used to show how different governance strategies may be positioned in relation to safety and controllability, efficiency and scalability, and autonomy and adaptability. The positions shown in the figure are qualitative and are derived from the literature mapping in Table 2, so they should be read as heuristic placements rather than empirical scores. The figure therefore serves as a design map for illustrating how emphasis on one governance objective may create pressure on the other two.

In this paper, the notion of a governance trilemma is used to describe a recurring design tension among the three objectives rather than to propose a universal law or a quantitatively validated frontier. Its purpose is to support conceptual comparison across governance strategies, especially when different interventions strengthen different aspects of agent governance while introducing distinct

constraints.

Validators, guardian agents, zero-trust authorization, MNI-based disclosure rules, fault observers, and human escalation are located closer to the safety and controllability side of the trilemma because they place additional checks on unsafe outputs, excessive disclosure, and consequential actions. Validator agents are mainly used to address unsafe prompt-injection behavior [34], while the interventions discussed in the Trust Paradox study focus on reducing over-exposure and authorization drift [16]. Related work on dynamic trust and zero-trust governance further supports access discipline in multi-agent settings [38,39], and fault-observer studies provide a control-theory analogy for reasoning about unstable system behavior [46,47]. These mechanisms can strengthen runtime control, although they may also introduce latency, reduce discretionary autonomy, and require explicit policy maintenance.

Efficiency and scalability are more closely associated with mechanisms that regulate the amount and structure of communication among agents. Dynamic topology, representative communication, event-triggered messaging, and lightweight filtering all seek to reduce unnecessary interaction while preserving task-relevant coordination. Dynamic consensus communication, for example, targets communication redundancy and misleading influence in LLM-based multi-agent systems [17]. Event-triggered MAS control studies separately motivate communication economy under bounded-performance assumptions [48,49]. The main risk in this direction is that overly aggressive pruning may remove useful dissent, suppress safety-critical messages, or obscure rare failure signals.

Debate, self-refinement, exploratory tool use, persistent memory, and dynamic adaptation to new contexts place greater weight on autonomy and adaptability because they allow agents to revise, extend, and coordinate their behavior across open-ended tasks. These mechanisms can support flexible problem solving, but the reviewed evidence also indicates that they may open additional pathways for prompt or protocol exploitation [5], hallucination propagation [13], hallucination snowballing [14], manipulated knowledge persistence [15], and authorization drift [16].

As a heuristic design map, the trilemma does not assume that any one objective must be fully abandoned when another is emphasized. It instead indicates that governance choices need to be calibrated according to the task context, evidence from the relevant coupling chains, and the observed risk state of the agent system. For example, a low-risk brainstorming task may allow greater autonomy and broader debate, whereas a privacy-sensitive or safety-critical task may call for guardian agents, MNI enforcement, and human approval. In high-throughput monitoring scenarios, event-triggered communication and dynamic topology may be used to preserve efficiency, while escalation rules remain necessary when risk indicators increase. These examples should be understood as design implications drawn from the reviewed literature, rather than as prescriptions already validated across all LLM-agent systems.

5. Discussion

The preceding analysis indicates that risks in LLM-based agents cannot be explained by a single model defect alone. A hallucination may initially appear as a textual error, yet its system-level

significance depends on whether it is transmitted through debate, incorporated into memory, reinforced by trust settings, shaped by authorization rules, converted into tool calls, or affected by control loops. The research focus of this paper is therefore the coupling process by which local defects become trajectory-level risks. On this basis, the following discussion identifies two directions for future research and engineering: runtime adaptive governance and authority-preserving architecture. These directions are grounded in the reviewed literature, while remaining research-oriented proposals rather than fully validated deployment models.

5.1 Runtime Adaptive Governance

Static governance is likely to face limits when autonomous agents operate across tasks with changing levels of risk. Guardrails, fixed prompts, and one-time model alignment can reduce certain unsafe outputs, although they do not continuously observe how defects develop during execution. In LLM-based multi-agent systems, the active risk state may shift with the task phase, communication topology, memory retrieval, trust level, and tool access. Open debate may be acceptable in a low-risk ideation task, while privacy-sensitive delegation may require stricter MNI enforcement, and a high-impact tool call may require escalation and audit. From this perspective, governance can be understood, in a limited analytical sense, as a runtime problem of monitoring and intervention.

A runtime adaptive governance loop may be described as Figure 5.

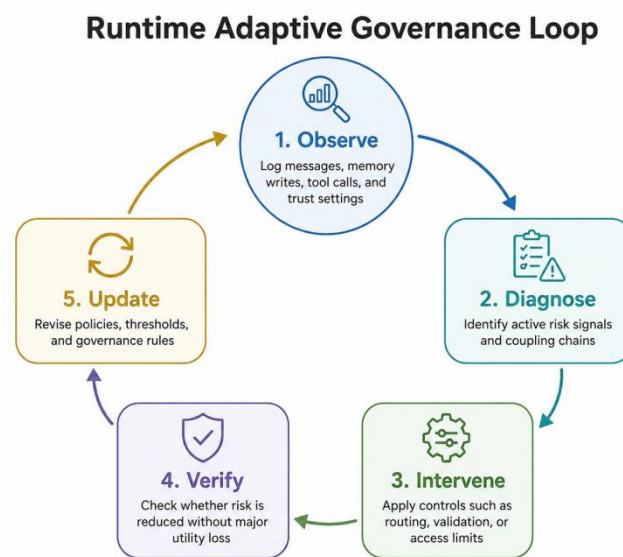


Figure 5. Runtime adaptive governance loop

As shown in Figure 5, the loop begins with observation, where the system records messages, retrieved evidence, memory writes, tool calls, authorization decisions, trust settings, and human approvals. This reflects the importance of observability memory proposed for hallucination mitigation [6] and also connects to the need for behavioral traceability discussed in broader agentic AI surveys [18].

These recorded events are then used for diagnosis by mapping them to coupling-chain indicators.

Hallucination propagation can be diagnosed through HPR, disclosure boundary failure through OER and AD, memory pollution through MCP, communication overload through CR, and unsafe action through policy-violation or escalation metrics. The Trust Paradox study provides a concrete example by tying MNI, OER, and AD to trust-mediated disclosure behavior [16].

Once the active coupling chain has been identified, the system can select governance actions that correspond to the observed risk pattern. When Chain A is active, semantic consistency reasoning, dynamic topology, debate rerouting, or validator agents may be used [13,14,17,34]. When Chain B is active, the system may tighten authorization, repartition sensitive information, activate guardian agents, or reduce trust-derived disclosure privileges [16,38,39]. When Chain C is active, memory provenance may be audited, contaminated retrieval entries may be quarantined, or unverified memory writes may be rolled back [6,15,32]. If C4 instability is observed, the system may adjust communication thresholds or escalate to human review, while observer-based and event-triggered MAS control remain analogical design vocabulary [46,47,48,49].

After intervention, verification is needed to determine whether the risk state has been reduced without causing unacceptable utility loss. Dynamic topology should be examined to determine whether it reduces HPR or communication load without suppressing necessary dissent. Guardian agents should be assessed according to whether they reduce OER and AD without blocking task-critical information. Memory auditing should be evaluated by whether it lowers MCP without removing valid domain knowledge. This step is necessary because governance mechanisms may themselves introduce failure modes such as over-refusal, excessive latency, loss of useful context, or coordination deadlock as a MAS analogy.

The loop concludes with updates to policies, trust weights, memory metadata, routing rules, or escalation thresholds. Such updates do not require uncontrolled self-modification and can instead be limited to auditable configuration changes, benchmarked policy revisions, or human-approved rule updates. The broader implication is that governance can become measurable and revisable when agents operate across changing tasks.

The runtime governance loop also points to several future prospects for research and engineering:

First, agent benchmarks could move beyond final-answer accuracy and include trajectory-level metrics. A system that reaches the correct final answer after spreading sensitive information or storing false knowledge has not necessarily been governed well. Benchmarks should therefore log intermediate messages, trust changes, memory writes, and tool calls.

Second, governance metrics need scenario grounding. MNI cannot be computed without a task specification, a role model, or a disclosure policy. OER and AD are meaningful only when information boundaries are defined. Governance evaluation therefore requires task design and policy design, not only model evaluation.

Third, communication topology should be treated as an experimental variable. The Woozle effect and dynamic consensus work suggest that who talks to whom, when, and with what message granularity can influence whether collaboration corrects or amplifies errors [13,17]. Future studies could compare fully connected, role-based, event-triggered, reliability-weighted, and dynamically

rewired topologies under identical tasks.

Fourth, memory governance requires longitudinal evaluation. Short-horizon evaluations may miss persistent contamination. Future work could test whether false or manipulated knowledge reappears after multiple tasks, whether provenance metadata changes agent reliance, whether learned forgetting and rollback reduce MCP without harming useful personalization, and whether memory security can be evaluated across the full write-store-retrieve-execute-share lifecycle [22,56].

5.2 Authority-Preserving Architecture

Human-in-the-loop governance is often described as adding human approval to an automated process. For LLM-based agents, the reviewed literature suggests a more specific question: which authority remains human when it is invoked, and how the system prevents uncertain reasoning from being converted into irreversible action.



Figure 6. Authority-preserving architecture

In Figure 6, the left arrow indicates that agent authority and operational risk increase from the advisory layer to the reserved layer. The right arrow indicates that human oversight should also increase as agent actions become more consequential, irreversible, or rights-affecting.

This layered view is consistent with the governance-trilemma heuristic because different levels of human oversight may shift the balance among controllability, efficiency, and autonomy. Requiring human approval for every step can strengthen controllability, although it may also reduce efficiency and limit agent autonomy. By contrast, granting agents full autonomy may improve adaptability while increasing exposure to hallucination propagation, authorization drift, and unsafe tool use. A practical architecture may therefore preserve human authority in situations involving high stakes, uncertainty, or irreversibility, while allowing bounded autonomy for tasks that are routine, reversible, and well-instrumented.

This architecture also depends on how trust is calibrated across agents and tasks. The Trust

Paradox study shows that increasing trust may improve coordination while also amplifying over-exposure or authorization drift [16]. For this reason, trust is better defined in relation to task requirements, available evidence, temporal context, and policy boundaries, rather than treated as a fixed label assigned to an agent. CogTrust and zero-trust situation-awareness research similarly point toward dynamic trust evaluation and continuous verification in multi-agent environments [38,39]. Requirements-engineering work on trust in multi-agent LLM chatbots further supports the view that trust should be incorporated into system design, instead of being assumed only after deployment [40].

An authority-preserving architecture can be organized around five related design principles:

First, least-privilege information flow. Agents receive the minimum necessary information for their role rather than full context by default. This operationalizes MNI and reduces the surface for OER as defined in trust-mediated disclosure settings [16].

Second, separable roles for generation, validation, authorization, and execution. A generator should not be the sole authority validating its own output, authorizing sensitive disclosure, and executing irreversible actions. The evidence on unreliable self-verification supports this separation [7].

Third, escalation by risk state rather than by fixed task type alone. A routine task can become high-risk if HPR, OER, AD, MCP, or policy-violation indicators rise. Conversely, a low-risk task can remain autonomous when the risk state is stable.

Fourth, memory provenance and auditability. Agents should distinguish externally verified facts, retrieved documents, user-provided claims, agent-generated summaries, and execution logs. Observability memory supports this distinction [6]. Manipulated knowledge spread shows why contaminated histories matter [15], and SagaLLM motivates validation and transaction safeguards for persistent planning state [32].

Fifth, reversible execution and rollback. Where possible, agents should act through staged transactions, dry runs, compensation steps, or approval checkpoints. SagaLLM's emphasis on validation, compensation, and transaction guarantees is relevant here [32].

The broader research challenge is to define human authority in operational rather than symbolic terms. Under the reviewed evidence, human authority is most relevant where goals, policies, escalation thresholds, high-impact actions, and post-incident accountability are at stake. Future research should therefore examine how authority-preserving architectures can be benchmarked against trajectory-level metrics, tool-use reliability, and memory-governance outcomes. This claim should be read as a governance implication from the reviewed literature rather than a complete institutional design theory.

6. Conclusion

This paper examined how local defects in LLM-based agents can become system-level risks when agents are connected to memory, tools, authorization rules, and other agents. The proposed defect-coupling framework organizes this problem around four mechanisms: generation-propagation, trust-authorization, knowledge-memory, and control-communication. This framing clarifies why a

hallucination or weak verification step should not be evaluated only as an isolated output error, because its significance depends on whether it is propagated, stored, trusted, authorized, or converted into action.

The review also shows that governance interventions should be matched to the relevant coupling pathway. Debate and peer review may improve factuality under suitable designs, but poor topology may allow false claims to spread. Validator and guardian agents are more relevant to runtime checking and least-privilege disclosure, whereas memory provenance, rollback, dynamic topology, and human escalation become important when risk is understood as a trajectory rather than a final-answer property. The paper therefore treats governance as adaptive, evidence-status-aware, and authority-preserving rather than as a single universal defense.

Future research can extend this conceptual synthesis through empirical evaluation. Promising directions include the development of trajectory-level benchmarks, scenario-grounded disclosure metrics, controlled comparisons of communication topologies, longitudinal investigations of memory contamination, and operational architectures that preserve human authority over high-impact actions. By distinguishing native defects, system vulnerabilities, downstream failure modes, and the evidentiary status of proposed metrics, the framework provides a more precise vocabulary for examining and advancing safer LLM-based agent ecosystems.

Acknowledgements

This article received no financial or funding support.

Conflicts of Interest

The authors confirm that there are no conflicts of interest.

References

- [1] Ferrag, M.A., Tihanyi, N. and Debbah, M. From LLM Reasoning to Autonomous AI Agents: A Comprehensive Review. *IEEE Access*, 2026, 14, 84237-84285. DOI: 10.1109/access.2026.3698694.
- [2] Tang, Y., Liu, Y., Lan, J., Yan, Z. and Gelenbe, E. Security of LLM-based agents regarding attacks, defenses, and applications: A comprehensive survey. *Information Fusion*, 2026, 127, 103941. DOI: 10.1016/j.inffus.2025.103941.
- [3] Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T.J., Cheng, Z., Shin, D., Lei, F., Liu, Y., Xu, Y., Zhou, S., Savarese, S., Xiong, C., Zhong, V. and Yu, T. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. *arXiv preprint*, 2024. DOI: 10.48550/arXiv.2404.07972.
- [4] Yao, S., Shinn, N., Razavi, P. and Narasimhan, K. τ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *arXiv preprint*, 2024. DOI: 10.48550/arXiv.2406.12045.
- [5] Ferrag, M.A., Tihanyi, N., Hamouda, D., Maglaras, L., Lakas, A. and Debbah, M. From prompt injections to protocol exploits: Threats in LLM-powered AI agents workflows. *ICT Express*, 2026, 12(2), 353-383. DOI: 10.1016/j.icte.2025.12.001.

- [6] Matharaarachchi, A., Moraliyage, H., Mills, N., Gamage, G., De Silva, D. and Manic, M. Addressing hallucinations in generative AI agents using observability and dual memory knowledge graphs. *Knowledge-Based Systems*, 2026, 338, 115469. DOI: 10.1016/j.knosys.2026.115469.
- [7] Yu, X., Liu, L., Hu, X., Keung, J.W., Liu, J. and Xia, X. Fight Fire With Fire: How Much Can We Trust ChatGPT on Source Code-Related Tasks? *IEEE Transactions on Software Engineering*, 2024, 50(12), 3435-3453. DOI: 10.1109/tse.2024.3492204.
- [8] Zhou, Y. and Chen, Y. Adaptive heterogeneous multi-agent debate for enhanced educational and factual reasoning in large language models. *Journal of King Saud University Computer and Information Sciences*, 2025, 37(10), 330. DOI: 10.1007/s44443-025-00353-3.
- [9] Xiong, C., Zheng, G., Ma, X., Li, C. and Zeng, J. DelphiAgent: A trustworthy multi-agent verification framework for automated fact verification. *Information Processing & Management*, 2025, 62(6), 104241. DOI: 10.1016/j.ipm.2025.104241.
- [10] Huang, J., Han, T., Yu, N. and Yi, X. Socratic Elenchus-inspired multi-agent debate for mitigating hallucinations in large language models. *Expert Systems with Applications*, 2026, 320, 132208. DOI: 10.1016/j.eswa.2026.132208.
- [11] Shi, J., Zhao, J., Wu, X., Xu, R., Jiang, Y.H. and He, L. Mitigating reasoning hallucination through Multi-agent Collaborative Filtering. *Expert Systems with Applications*, 2025, 263, 125723. DOI: 10.1016/j.eswa.2024.125723.
- [12] Shen, S. H., Sun, D. D., Qin, Z. X., Su, Y., Zhu, L. B., Lu, J. Y., Huang, Z. Y. and Wang, S. J. Towards higher quality and fewer hallucinations: A multi-agent collaboration framework for LLMs. *Information Processing & Management*, 2026. DOI: 10.1016/j.ipm.2026.104953.
- [13] Zhang, Z., Zhang, Y., Zeng, D., Liu, K. and Zhao, J. Beware of the Woozle Effect: Exploring and Mitigating Hallucination Propagation in Multi-Agent Debate. *IEEE Transactions on Audio, Speech and Language Processing*, 2026, 34, 2021-2032. DOI: 10.1109/taspro.2026.3675803.
- [14] Xu, X. and Wu, J. Mitigating LLM Hallucination Snowballing in Multiagent Systems via Context-Aware Semantic Consistency Reasoning. *IEEE Transactions on Neural Networks and Learning Systems*, 2026, 1-15. DOI: 10.1109/tnnls.2026.3655508.
- [15] Ju, T., Wang, Y., Hua, Y., Ma, X., Cheng, P., Zhao, H., Wang, Y., Liu, L., Xie, J., Zhang, Z. and Liu, G. Flooding spread of manipulated knowledge in LLM-based multi-agent communities. *Science China Information Sciences*, 2026, 69(7), 172103. DOI: 10.1007/s11432-024-4663-2.
- [16] Xu, Z., Qi, M., Wu, S., Zhang, L., Wei, Q., He, H. and Li, N. The Trust Paradox in LLM-Based Multi-Agent Systems: When Collaboration Becomes a Security Vulnerability. *IEEE Transactions on Computational Social Systems*, 2026, 1-12. DOI: 10.1109/tcss.2026.3695070.
- [17] Yang, L., Li, S. and Deng, A. Dynamic Consensus Communication Mechanism for Large Language Model-Based Multi-Agent Systems. *Journal of Signal Processing Systems*, 2026, 98(1), 10. DOI: 10.1007/s11265-025-01978-3.
- [18] Khapre, S., Mersha, M.A., Olalekan, Z., Shakil, H., Iskander, S., Moustafa, A. and Kalita, J. Agentic AI systems: A systematic survey of multi-agent architectures, cognitive foundations, interaction, explainability, security, and performance evaluation. *Neurocomputing*, 2026, 696, 134049. DOI: 10.1016/j.neucom.2026.134049.
- [19] Sapkota, R., Roumeliotis, K.I. and Karkee, M. AI Agents vs. Agentic AI: A Conceptual taxonomy, applications and challenges. *Information Fusion*, 2026, 126, 103599. DOI: 10.1016/j.inffus.2025.103599.
- [20] Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z. and Wen, J.R. A Survey on the Memory Mechanism of Large Language Model based Agents. *arXiv preprint*, 2024. DOI: 10.48550/arXiv.2404.13501.

- [21] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O. and Zhang, X. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. arXiv preprint, 2024. DOI: 10.48550/arXiv.2402.01680.
- [22] Du, P. Memory for Autonomous LLM Agents: Mechanisms, Evaluation, and Emerging Frontiers. arXiv preprint, 2026. DOI: 10.48550/arXiv.2603.07670.
- [23] Agada, J.O. and Biswas, A. LLMs, RAG systems, and agents in crystalline materials discovery and characterization: A systematic review. MRS Bulletin, 2026, 51(5), 481-515. DOI: 10.1557/s43577-026-01057-3.
- [24] Wang, S. and Ye, Q. Enhancing Large Language Models With AI Agents for Chronic Gastritis Management: Comprehensive Comparative Study. JMIR Medical Informatics, 2025, 13, e73857. DOI: 10.2196/73857.
- [25] Deng, L., Hu, H., Lu, K. and He, P. LLM-augmented multi-agent cooperative framework for medical case retrieval in cardiology. Journal of King Saud University Computer and Information Sciences, 2025, 37(9), 267. DOI: 10.1007/s44443-025-00311-z.
- [26] Salehi, S., Singh, Y., Horst, K.K., Hathaway, Q.A. and Erickson, B.J. Agentic AI and Large Language Models in Radiology: Opportunities and Hallucination Challenges. Bioengineering, 2025, 12(12), 1303. DOI: 10.3390/bioengineering12121303.
- [27] Qian, K.Y., Liu, S.Q., Li, T.G., Rakovic, M., Li, X.Y., Guan, R., Molenaar, I., Nawaz, S., Swiecki, Z., Yan, L.X. and Gasevic, D. Towards reliable generative AI-driven scaffolding: Reducing hallucinations and enhancing quality in self-regulated learning support. Computers & Education, 2026, 240, 105448. DOI: 10.1016/j.compedu.2025.105448.
- [28] Liu, X., Liu, Y., Zhuang, Y. and Hou, W. UCD-LLM: A use case diagram requirement modeling multi-agent framework with large language model. Information and Software Technology, 2026, 190, 107955. DOI: 10.1016/j.infsof.2025.107955.
- [29] Zhu, Y., Liu, C., He, X., Ren, X., Liu, Z., Pan, R. and Zhang, H. AdaCoder: An Adaptive Planning and Multi-Agent Framework for Function-Level Code Generation. IEEE Transactions on Software Engineering, 2026, 52(2), 631-650. DOI: 10.1109/tse.2025.3642621.
- [30] Wang, Z., Chen, J., Wei, Z., Lin, L. and Tong, G. SafeCodeRL: Security-Constrained Multi-Agent Reinforcement Learning for Trustworthy LLM-Generated IoT/CPS Software. Sensors, 2026, 26(11), 3502. DOI: 10.3390/s26113502.
- [31] He, J., Treude, C. and Lo, D. LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead. ACM Transactions on Software Engineering and Methodology, 2025, 34(5), 1-30. DOI: 10.1145/3712003.
- [32] Chang, E.Y. and Geng, L.L. SagaLLM: Context Management, Validation, and Transaction Guarantees for Multi-Agent LLM Planning. Proceedings of the VLDB Endowment, 2025, 18(12), 4874-4886. DOI: 10.14778/3750601.3750611.
- [33] Xu, H., Zhang, W., Wang, Z., Xiao, F., Zheng, R., Ba, Z. and Ren, K. RedAgent: An Autonomous Agent for Context-Aware Red Teaming of LLM Jailbreaks. IEEE Transactions on Dependable and Secure Computing, 2026, 23(3), 6506-6521. DOI: 10.1109/tdsc.2026.3665230.
- [34] Podpora, M., Baranowski, M., Chopian, M., Kwasniewicz, L. and Radziejewicz, W. LLM Firewall Using Validator Agent for Prevention Against Prompt Injection Attacks. Applied Sciences, 2026, 16 (1), 85. DOI: 10.3390/app16010085.
- [35] Sato, G., Egami, S., Tahara, Y., Ohsuga, A. and Sei, Y. Addressing Prompt Injection in Large Language Models via In-Context Learning. Computers, Materials & Continua, 2026, 87(2), 1-10. DOI: 10.32604/cmc.2026.078188.

- [36] Xu, Q., Tian, Z., Wu, H., Huang, Z., Liang, D., Song, Y., Pan, Z., Liu, F. and Li, D. Disguise While Defense: Avoid Refusal Responses in LLM's Defense via a Multi-Agent Attacker-Disguiser Game. *IEEE Transactions on Information Forensics and Security*, 2026, 21, 1006-1020. DOI: 10.1109/tifs.2025.3626228.
- [37] Ke, Y., Yang, R., Lie, S.A., Lim, T.X.Y., Ning, Y., Li, I., Abdullah, H.R., Ting, D.S.W. and Liu, N. Mitigating Cognitive Biases in Clinical Decision-Making Through Multi-Agent Conversations Using Large Language Models: Simulation Study. *Journal of Medical Internet Research*, 2024, 26, e59439. DOI: 10.2196/59439.
- [38] Wang, J., Yang, S., Li, J., Zhao, Y., Wang, Y. and Wu, S. CogTrust: Cognitive Logic-Based framework for dynamic trust evaluation in multi-agent systems. *Expert Systems with Applications*, 2026, 313, 131535. DOI: 10.1016/j.eswa.2026.131535.
- [39] Cao, X., Nan, G., Guo, H., Mu, H., Wang, L., Lin, Y., Zhou, Q., Li, J., Qin, B., Cui, Q., Tao, X., Fang, H., Du, H. and Quek, T.Q.S. Exploring LLM-Based Multi-Agent Situation Awareness for Zero-Trust Space-Air-Ground Integrated Network. *IEEE Journal on Selected Areas in Communications*, 2025, 43(6), 2230-2247. DOI: 10.1109/jsac.2025.3560042.
- [40] Axetorn, J., Edholm, F., Dobslaw, F. and Gren, L. Addressing trust requirements in the design of an open-source multi-agent LLM-based domain-specific chatbot. *Requirements Engineering*, 2026, 31(1), 3. DOI: 10.1007/s00766-026-00457-w.
- [41] Chatzistefanidis, I. and Nikaein, N. Symbiotic agents: A novel paradigm for trustworthy AGI-driven networks. *Computer Networks*, 2025, 273, 111749. DOI: 10.1016/j.comnet.2025.111749.
- [42] Kraidia, I., Qaddara, I., Almutairi, A., Alzaben, N. and Belhouari, S.B. When collaboration fails: persuasion driven adversarial influence in multi agent large language model debate. *Scientific Reports*, 2026, 16(1), 11640. DOI: 10.1038/s41598-026-42705-7.
- [43] Hu, J., Zhang, J. and Li, Z. Breaking Barriers: Multi-Agent Prompt Fusion for Automated LLM Jailbreaks. *Cognitive Computation*, 2026, 18(1), 32. DOI: 10.1007/s12559-026-10552-3.
- [44] Bissoli, A., Mollaeefar, M., Van Landuyt, D. and Ranise, S. Benchmarking the effectiveness of multi-agent LLMs in collaborative privacy threat modeling with LINDDUN GO. *Journal of Information Security and Applications*, 2026, 100, 104489. DOI: 10.1016/j.jisa.2026.104489.
- [45] Du, P. and Li, G. Enhancing PIBT for Multi-Agent Path Finding via MLP-Based Candidate Selection and Priority Perturbation. *IEEE Access*, 2025, 13, 193403-193413. DOI: 10.1109/access.2025.3630099.
- [46] Xia, D. and Fu, X. Observer-based sliding-mode fault-tolerant consistent control for hybrid event-triggered multi-agent systems. *International Journal of Applied Mathematics and Computer Science*, 2024, 34(3). DOI: 10.61822/amcs-2024-0026.
- [47] Liu, Y., Chen, J., Yang, Y. and Fu, H. Simultaneous Fault Estimation and Fault-Tolerant Consensus Control for Multi-Agent Systems Against Multiple Time Delays. *IEEE Access*, 2025, 13, 93551-93565. DOI: 10.1109/access.2025.3573827.
- [48] Liu, X., Hui, M., Fan, L., Yu, W. and Cao, J. Adaptive fuzzy funnel control of nonlinear multi-agent systems via dual-channel event-triggered strategy. *Information Sciences*, 2025, 712, 122129. DOI: 10.1016/j.ins.2025.122129.
- [49] Qiu, X.J., Meng, W.C., Wang, Y.C. and Yang, Q.M. Data-driven-based fully distributed event-triggered control for nonlinear multi-agent systems. *Applied Mathematics and Computation*, 2025, 495, 129307. DOI: 10.1016/j.amc.2025.129307.
- [50] Ma, X., Rao, G., Xu, L., Wang, X., Fan, Z. and Zhang, Z. Guided and knowledgeable multi-agent debate for fact

verification. *Expert Systems with Applications*, 2026, 299, 130103. DOI: 10.1016/j.eswa.2025.130103.

- [51] Alqahtani, H. and Ahuja, P. Secure autonomous cyber defense with LLM agents: A systematic review of autonomy, tool-augmented reasoning, and governance constraints. *Computers and Electrical Engineering*, 2026, 135, 111184. DOI: 10.1016/j.compeleceng.2026.111184.
- [52] Alqahtani, H. and Kumar, G. Large Language Models for Cybersecurity Intelligence: A Systematic Review of Emerging Threats, Defensive Capabilities, and Security Evaluation Frameworks. *Computers, Materials & Continua*, 2026, 87(3), 1-10. DOI: 10.32604/cmc.2026.077367.
- [53] Yang, Y., Ma, Y., Feng, H., Cheng, Y. and Han, Z. Minimizing Hallucinations and Communication Costs: Adversarial Debate and Voting Mechanisms in LLM-Based Multi-Agents. *Applied Sciences*, 2025, 15(7), 3676. DOI: 10.3390/app15073676.
- [54] Zou, X., Li, X., Hu, P. and Dong, M. MRBalance: A framework for enhancing event causality identification in multi-agent debates via role assignment. *Knowledge-Based Systems*, 2025, 330, 114470. DOI: 10.1016/j.knosys.2025.114470.
- [55] Dong, Y. and Ito, T. From consensus theory to LLM agents: Practical consensus-building for multi-issue negotiation. *Expert Systems with Applications*, 2026, 322, 132250. DOI: 10.1016/j.eswa.2026.132250.
- [56] Lin, Z., Hao, X., Fu, R., Cui, S., Chen, K., Li, C., Li, Z. and Xiong, F. A Survey on Long-Term Memory Security in LLM Agents: Attacks, Defenses, and Governance Across the Memory Lifecycle. *arXiv preprint*, 2026. DOI: 10.48550/arXiv.2604.16548.
- [57] Hao, Z., Fei, H., Liu, C., Fu, X. and Luo, B. Aligning large language models across the lifecycle: A survey on safety-usability trade-offs from pre-training to post-training. *Neural Networks*, 2026, 202, 108996. DOI: 10.1016/j.neunet.2026.108996.

Copyright© by the authors, Licensee Intelligence Technology International Press. The article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA).