INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

AGENTIC AI IN REAL-WORLD SYSTEMS: A CRITICAL REVIEW OF AUTONOMOUS WORKFLOWS, TOOL INTEGRATION, APPLICATIONS, AND PERFORMANCE EVALUATION

Laiba Saeed[1], Hamid Ghous[2], Muhammad Zubair[3], Muzammil Hussain[4], Muhammad Akmal Shahzad[5], Ghulam Muhy Ud Deen Raee[*6]

[1]*Institute of Computer and Software Engineering, Khawaja Fareed UEIT, Rahim Yar Khan 64200, Pakistan*
[2,5,6]*Department of Computer Science and Information Technology, University of Southern Punjab (USP), Multan 60000, Punjab, Pakistan*
[3,4]*Department of Computer Science, National University of Modern Languages (NUML), Islamabad 44000, Pakistan*
[*6]*Department of Computer Science and Information Technology, School Education Department, Government of the Punjab, Muzaffargarh 34200, Pakistan*

[1]laibasaeed026@gmail.com, [2]hamidghous@usp.edu.pk, [3]zubairraee@gmail.com, [4]muzammilghalou@gmail.com, [5]akmalshahzadgurmani@gmail.com, [*6]gmraees@gmail.com

DOI: https://doi.org/10.5281/zenodo.24361597

Keywords
Agentic AI; LLM-based agents; multi-agent systems; tool use; Model Context Protocol; agent benchmarking; prompt injection; AI governance

Article History
Received: 05 August 2026
Accepted: 19 October 2026
Published: 19 November 2026

Copyright @Author
Corresponding Author: * Ghulam Muhy Ud Deen Raee

Abstract

Agentic artificial intelligence (AI) systems, in which a learned model plans, invokes tools and acts over multiple steps toward a goal, are moving rapidly from research prototypes into enterprise, clinical, industrial and software-engineering settings. Existing reviews organise this literature by capability and rarely separate what has been demonstrated on benchmarks from what has been deployed. This critical review synthesises 80 sources published mainly between 2024 and 2026, identified through an executed, single-index scoping search and targeted supplementary retrieval, and appraises them by validation tier, metric definition and comparability. Thirty-five comparative tables cover agent architectures, workflow execution, tool integration and interoperability protocols, orchestration, failure recovery, domain applications, benchmarks, metrics, efficiency, security and governance. Four findings recur. First, none of the 27 application cases analysed reports a production deployment with measured outcomes; 10 are benchmark-only and 5 are laboratory-tier. Second, reported performance varies more with evaluation configuration (benchmark subset, version, metric and trial count) than with system design, so cross-benchmark rankings are not defensible. Third, agentic execution carries an order-of-magnitude token and latency overhead, yet no reviewed study reports cost per successful task. Fourth, detection-based defences against prompt injection fail under adaptive attack, whereas structural privilege separation produced the only near-zero attack success rate in the corpus. The review contributes a workflow taxonomy based on execution semantics, a validation-tier label for application evidence, a comparability rule for benchmark scores, and a proposed integrated evaluation perspective that links task performance, reliability, interoperability, cost, security and scalability, together with a prioritised research roadmap.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

INTRODUCTION

Evolution of Artificial Intelligence Towards Agentic AI

The historical record supports a two-epoch periodisation: a pre-LLM stage (1950s–2020s) encompassing symbolic, reactive, reinforcement-learning-based and meta/transfer agents, followed by a large-model stage (2022–present) in which planning, tool use and memory are implemented through a foundation model rather than through hand-coded control logic (Su et al., 2026). Within the first epoch, three behavioural architectures were distinguished. Deliberative agents deliberate on their goals, plans or actions and act on a sense–model–plan–act cycle, which requires a symbolic representation of the world; reactive agents implement simple behavioural patterns and respond to events in a stimulus–response way, requiring no model of the world; hybrid designs combine both (Sado et al., 2022).

Table 1. Evolution and Major Milestones of Agentic AI

Period/Year	Major Development	Enabling Technology	Agentic Capability	Technical Significance	Supporting Reference
Pre-LLM (1950s–2020s)	Symbolic/ deliberative agents	Symbolic representation of the world	Sense–model–plan–act cycle	Establishes deliberation over goals ahead of action	(Sado et al.,2022; Su et al., 2026)
Pre-LLM (1980s–1990s)	Reactive architectures	Stimulus–response behavioural patterns	Reaction to environmental events	Removes the world-model requirement; permits real-time response	(Sado et al.,2022)

Critical interpretation. Three findings emerge that a descriptive chronology would miss. First, **the milestones are not monotonic in autonomy**: the reactive turn was a deliberate reduction of representational commitment, and LLM agents reproduce that trade-off implicitly, holding no explicit world model and relying on context rather than state — the reason the deliberative/reactive distinction remains analytically useful is that it names exactly what was given up (Sado etal., 2022).

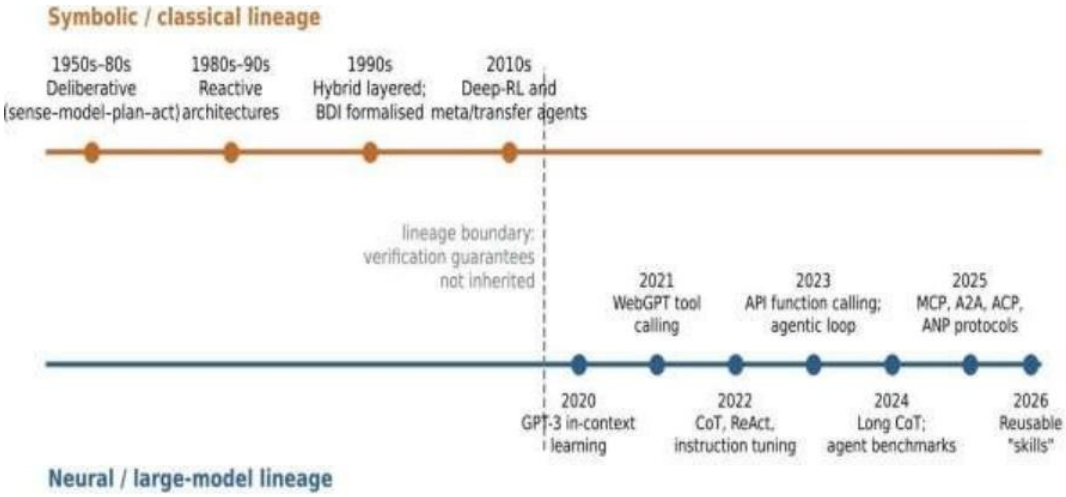


Figure 1 places the milestones of Table 1 on two tracks, making the break between the symbolic and neural lineages explicit.

Figure 1. Two-epoch evolution of agentic AI. The symbolic lineage (deliberative, reactive, hybrid/BDI and learned agents) and the neural lineage (in-context learning, tool calling, chain-of-thought and ReAct, API function calling, agent benchmarks, interoperability protocols and reusable skills) are drawn as separate tracks because contemporary agents reproduce BDI-style loops without inheriting their verification guarantees. Milestones and sources follow Table 1.

Characteristics and Significance of Real-World Agentic AI

Autonomy and agency are routinely conflated in the applied literature, but the distinction is decisive for deployment claims. Autonomy denotes a system's capacity to operate independently of direct humanintervention, whereas agency denotes goal-directed behaviour incorporating intention, contextual awareness and decision-making; agentic AI synthesises the two by initiating tasks, dynamically ranking goals, monitoring progress and adjusting behaviour through feedback (AbouAli et al., 2025). A characterisation carrying more analytical weight for deployment is that an agent differs from a generative model in holding **delegated action rights under constraints** — the right to invoke tools and mutate external state without per-action approval (Mustafa & Aysan, 2026).

Table 2 contrasts traditional, generative and agentic systems along the attribute set that governs real-world operation — not only capability but failure mode, evaluation object and human role — so that the axes on which the paradigms differ in kind are separated from those on which they differ only in degree.

International Journal of Advanced Research

Table 2. Comparative Characteristics of Traditional AI, Generative AI, and Agentic AI

Characteristic	Traditional AI	Generative AI	Agentic AI	Real-World Implication	Supporting Reference
Primary output	Prediction, class, score	Text, code, image	Actions executed in, and on, the environment	Appraisal must inspect external state, not output plausibility	(Zhao et al., 2026)
Autonomy level	None (invoked per input)	None (single completi on)	Task initiation, goal ranking, progress monitoring	Requires accountability structures beyond model evaluation	(AbouAli et al.,2025)

Critical interpretation. The table exposes an asymmetry under-theorised in existing reviews: agentic systems are stronger on capability axes (action space, horizon, adaptivity) and weaker on assurance axes (determinism, verifiability, comparability of evaluation). Two consequences follow. First, evaluating agents with generative-AI metrics systematically overstates readiness, because accuracy and preference scores do not observe the external state the agent left behind — hence the argument that benchmarks must test process correctness and not only final answers (Kou et al., 2026).

Research Motivation, Gaps, and Objectives

Agentic AI now has no shortage of reviews; at least a dozen appeared between 2024 and 2026 covering construction frameworks, architectures, optimisation, security, evaluation and domain applications. The gap is not coverage but **integration**: no existing review simultaneously classifies workflow execution semantics, audits tool-integration protocols, assigns a validation tier to each reported application, and gates performance comparison on metric comparability. Reviews that attempt quantitative synthesis inherit their primary studies' reporting weaknesses; one survey documents that a widely

used environment, WebArena, was found vulnerable to gaming, undermining its construct validity as a capability measure (Peykani et al., 2026).

Table 3 critiques 13 representative reviews, recording scope, method, principal contribution, the omitted dimension, and the resulting gap. The purpose is not to diminish prior work but to establish transparently why a further review is warranted and what it must do differently.

Table 3. Critical Analysis of Existing Agentic AI Reviews and Research Gaps

Author/ Year	Review Scope	Methodology	Major Contribution	Missing Research Dimension	Identified Gap	DOI/Reference
L. Wang et al. (2024)	Agent construction (profile, memory, planning, action), applications, evaluation	Narrative survey with capability - coding table	Unified construction framework; dated capability matrix	Validation-tier coding; cost analysis	Cannot separate demonstrated from deployed capability	10.1007/ s11704-024-40231-1 (L. Wang et al.,20 24)
Zhao et al. (2026)	General LLM survey incl. agent components	Comprehen sive survey	Memory/ planning/ execution decomposition; long-horizon failure analysis	Workflow execution semantics; protocol-level tool integration	Architectural taxonomy not linked to deployment outcomes	10.1007/ s11704-026-60308-3 (Zhao et al., 2026)

International Journal of Advanced Research

Critical interpretation. The gap pattern is systematic rather than incidental: six of thirteen reviews omit cost or efficiency entirely, five omit security, and none applies a per-case deployment tier; where comparability problems are acknowledged they are described but not converted into an analytic rule (Peykani et al., 2026). The convergence is explained by a shared methodological choice — all thirteen are organised by capability (what an agent can do) rather than by evidence (what has been demonstrated, where, and under what conditions).

Scope and Contributions of the Review

The scope is bounded deliberately. The review covers systems in which a learned model plans, calls tools, or acts over multiple steps toward a specified objective; excludes single-turn generation, recommendation and classical predictive modelling except as baselines; covers 2020–October 2026 with pre-2020 sources admitted only as foundations; and restricts domains to healthcare, finance and business processes, manufacturing and industrial systems, cybersecurity and software engineering, and enterprise IT, because these are the domains with any deployment evidence. Benchmarks are treated as measurement instruments, not applications, and analysed separately in Section 6.

Table 4. Scope and Contributions of the Present Review Compared with Existing Studies

Research Dimension	Coverage in Existing Reviews	Identified Limitation	Present Review Contribution	Supporting Evidence
Organising principle	All 13 reviews in Table 3 are organised by capability (what an agent can do).	Capability framing does not distinguish what was demonstrated from what was deployed.	Evidence-based organisation that separates benchmark, laboratory and deployment evidence.	Table 3; Sections 2–5
Workflow execution semantics	Architectures and components are described; execution is not classified.	Cannot compare who holds control or how failures propagate across workflows.	Workflow taxonomy based on execution semantics.	Table 13; Figure 5
Tool integration and interoperability	Tool use is covered at component level, rarely at protocol level.	Interoperability protocols are not audited against one another.	Comparative audit of tool-integration techniques and interoperability protocols (including MCP).	Table 14; Figure 5
Application validation	No existing review applies a per-case deployment tier.	Benchmark and laboratory results are read as deployment readiness.	Validation-tier label applied to 27 application cases; none reports production deployment with measured outcomes.	Tables 17–20; Figure 7
Benchmark comparability	Comparability problems are acknowledged but only described.	Scores from different subsets, versions, metrics and trial counts are ranked together.	Comparability rule that restricts comparison to within benchmark families.	Tables 21–24; Section 6.4
Cost and efficiency	Six of thirteen reviews omit cost or efficiency entirely.	Token and latency overhead is unquantified; no cost per successful task.	Efficiency synthesis showing an order-of-magnitude overhead and the absence of cost-per-success reporting.	Table 23; Section 6.3
Security and prompt injection	Five of thirteen reviews omit security.	Defences are listed without comparing measured attack success under adaptive attack.	Threat and mitigation comparison: detection-based defences fail under adaptive attack; structural privilege separation gave the only near-zero attack success rate.	Table 25; Section 7.1

Research Dimension	Coverage in Existing Reviews	Identified Limitation	Present Review Contribution	Supporting Evidence
Failure and recovery	Failures are discussed qualitatively within architecture or security sections.	No link between failure type, monitoring approach and recovery strategy.	Failure taxonomy mapping each failure type to monitoring and recovery strategies.	Tables 16 and 26; Section 7.2
Governance and oversight	Treated as a closing discussion point in most reviews.	Human-oversight and auditability mechanisms are not tied to deployment barriers.	Comparison of trustworthiness, oversight and governance mechanisms alongside deployment barriers.	Tables 27–28; Section 7.4
Evaluation framework and roadmap	Future directions are listed without a shared evaluation basis.	No single perspective links performance, reliability, cost, security and scalability.	Integrated six-dimension evaluation perspective and prioritised research roadmap.	Figure 10; Table 32

Critical interpretation. Differentiation is strongest where existing reviews are most confident. On coordination, prior reviews assert that multi-agent designs improve specialisation and fault tolerance (Chowa et al., 2026) and that decentralisation improves robustness under incomplete global state (Zhao et al., 2026), while controlled comparison finds collaboration benefits shrink with model capability and that an independent ensemble isolates parallelism from coordination effects (Y. Kim et al., 2026).

Review Methodology

Literature Search Strategy and Scientific Databases

The search was executed on **31 October 2026** as a structured scoping review against an **OpenAlex-indexed scholarly corpus**, using 15 semantically rich query strings; **70 records** were returned, representing **69 distinct identifiers**, and collapsing four cross-listed duplicates (OSWorld; the hospital agentic-operating-system paper; the privilege-separation study; the MCP/modelling paper) yielded **65 unique publications**. Scopus, Web of Science, IEEE Xplore, ACM Digital Library, ScienceDirect and SpringerLink were **not queried live**, so no per-database retrieval, deduplication or PRISMA screening counts are reported for them. This review consequently distinguishes the **executed scoping protocol** (Table 5, rows 1–15, fully traceable) from the **specified protocol for the full systematic search** (rows 16–21, unexecuted).

Table 5. Scientific Databases, Search Strategies, and Literature Coverage

Database	Search String	Search Period	Search Date	Records Retrieved	Search Restrictions
OpenAlex	LLM-based autonomou s agents survey: planning, memory, tool use	≥2023	31 Oct 2026	5	English; scholarly index only

Database	Search String	Search Period	Search Date	Records Retrieved	Search Restrictions
OpenAlex	Agentic AI benchmark evaluation: WebArena, OSWorld, SWE-bench, GAIA, τ-bench	All years	31 Oct 2026	5	English

Critical interpretation. Two limitations follow and are carried into the limitations statement. First, **the executed corpus is single-index and therefore subject to coverage bias**: OpenAlex aggregates broadly but does not reproduce venue-specific indexing rules, so conference proceedings that dominate agentic benchmark work may be over- or under-represented relative to a Scopus/WoS union.

Inclusion and Exclusion Criteria

Eligibility was operationalised at ten decision points, each with an inclusion rule, exclusion rule and scientific justification, so that a second reviewer could reproduce every decision. The criteria are asymmetric on two points. First, **framework specifications are admissible** for statements about what a framework exposes, because for such statements the specification is the correct evidentiary source and a secondary citation would introduce error; such sources are flagged as non-peer-reviewed. Second, **benchmark**

International Journal of Advanced Research

Table 6. Inclusion and Exclusion Criteria for Literature Selection

Criterion	Inclusion Rule	Exclusion Rule	Scientific Justification
Object of study	System that plans, calls tools, or acts over multiple steps	Prompt-only application with no action loop	The unit is the acting system, not the generator (Sapkota et al.,2025)
Time window	2020–Oct 2026; older only as foundation	Pre-2020 non-foundational	Agentic behaviour postdates the 2022–2023 transition (Breaux & Singhal,2026)

Critical interpretation. These criteria are stricter than those of several reviews in the corpus, which admit domain applications on reported utility rather than reported protocol. The strictness has an identifiable cost

Study Selection and Quality Assessment

Selection proceeded in three passes — title/abstract screen, full-text assessment, quality appraisal — with the instrument applied only at the third pass, so that quality judgements could not retroactively bias retrieval. Ten criteria were scored, of which three are gating rather than graded: validation tier, metric definition, and security assessment for tool-using systems. A study failing a gate is labelled rather than excluded, which preserves coverage while preventing the labelled weakness from being silently absorbed into synthesis.

Table 7. Study Quality Assessment Criteria and Evaluation Framework

Quality Criterion	Operational Definition	Assessment Method	Scoring/Decision Rule	Relevance
Validation tier	Production / field trial / lab prototype / simulation /benchmark-only	Manual coding from environment description	Gate: tier assigned or flagged unreported	Sections 5.1–5.4
Baseline adequacy	Comparison with non-agentic or single-agent baseline	Checklist	0 = none, 1 = weak, 2 = adequate	Sections 3.2, 6.4

Critical interpretation. Applying a ten-point instrument to 65 records produced one dominant pattern: quality was highest on validation tier and sample size (studies are explicit about where they ran and how many tasks they used) and lowest on stochasticity, efficiency and threat models (they are largely silent). The methodological implication is direct: because variance is rarely reported, **no claim of statistical superiority between agent systems is made anywhere in this manuscript**, and differences in reported point estimates are described as reported differences only.

Data Extraction and Thematic Analysis

Extraction used fourteen variables in six categories, chosen so that each variable feeds a specific analytic artefact rather than a narrative section: architecture coding feeds the taxonomy in Section 3.2, workflow coding feeds Section 4.1, benchmark-version coding feeds the comparability gate in Section 6.4, validation-tier coding feeds Section 5, and efficiency coding feeds Section 6.3. Variables that could not be extracted for a majority of studies were retained in the schema precisely so that their absence becomes a reportable finding

Table 8. Data Extraction Variables and Thematic Classification Framework

Data Category	Extracted Variable	Operational Definition	Analytical Purpose	Relevant Review Section
Bibliographic	Study identifier, authors, year, venue, peer-review status	Standard bibliographic fields	Provenance; trend analysis; preprint flagging	2.1–2.3
System	Agent architecture	Single / independent ensemble / centralised / decentralised / hierarchical /hybrid	Topology taxonomy	3.2

Foundations and Architectures of Agentic AI

Core Components and Operational Principles

Current surveys decompose the agent into memory, planning and execution. Memory is the information-storage backbone that retains historical interactions and contextual knowledge; planning decomposes complextasks into structured sub-task sequences for long-horizon decision-making and is often performed iteratively so that strategy can be revised on newly observed environmental feedback; execution carries out the actions specified by planning and interacts directly with the environment (Zhao et al., 2026). An earlier and influential decomposition adds a profile and treats action as its own module, giving profile, memory, planning and action (L. Wang et al., 2024).

Table 9 maps each component to its technical function, operational mechanism, representative approach and characteristic limitation, establishing that each component carries a specific and non-interchangeable failure mode —

the analytical basis for the failure taxonomy in Section 7.2.

Table 9. Core Components and Operational Characteristics of Agentic AI Systems

Component	Technical Function	Operational Mechanism	Representative Approach	Advantages	Limitations	Reference
Environment interaction	Act on and observe external state	Tool or actuator invocation returning observations	API calls; browser actions; code execution	Grounds the loop in real systems	Observation fidelity not separately evaluated in the reviewed corpus	(Zhao et al., 2026)
Reasoning	Produce intermediate inference before acting	Intermediate reasoning steps, extended at test time	Chain-of-thought prompting; long reasoning	Makes reasoning inspectable and supports multi-step problems	Reasoning quality varies with task complexity and environment modelling	(Breaux& Singhal,2026 ; Chowa et al.,2026)

Critical interpretation. Three observations are consequential. First, **components are not independent**: planning consumes retrieved memory, so a retrieval failure is externally indistinguishable from a planning failure, which is why post-hoc error attribution in agent traces is unreliable — a point that recurs in the auditability problem of Section 7.4. Second, **verification is the weakest component relative to itsimportance**: it is the only component whose function is to prevent irreversible external action, yet the best available agent judges satisfy only a fraction of stated requirements with high inter-rater disagreement (Zhuge et al., 2024).

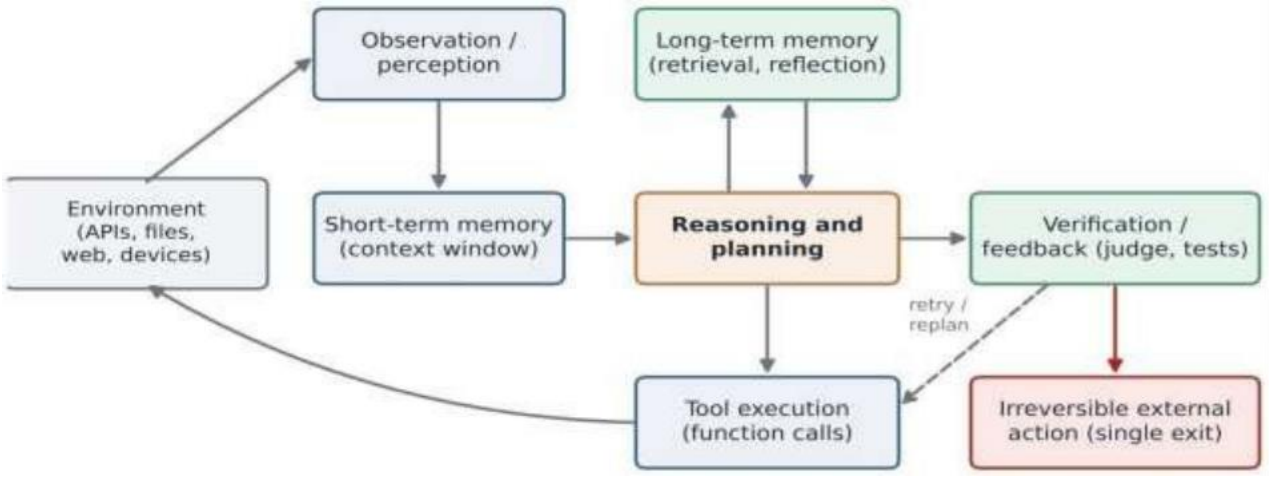


Figure 3 shows how the components in Table 9 connect in a single agent loop, including the verification step that guards the only irreversible exit.

Single-Agent and Multi-Agent Architectures

The literature converges on a small set of topologies — single-agent, independent ensemble, centralised, decentralised, hierarchical and hybrid — and on a corresponding set of trade-offs. Centralised arrangements concentrate control in a coordinating agent, simplifying decision-making and reducing conflicts but introducing single

points of failure and communication delays that impair responsiveness to environmental change (Li etal., 2024). In such designs a single LLM may act as central planner, achieving centralised training with decentralised execution and thereby reducing context requirements (Li et al., 2024). Decentralised arrangements distribute control so that agents negotiate and decide locally, improving robustness and scalability when global state information is incomplete (Zhao et al., 2026); their cost is that consensus and synchronisation protocols must maintain global coherence, adding complexity and overhead (B. Liu et al., 2025).

Table 10 compares eleven architecture classes across coordination mechanism, reported benefits, scalability evidence and limitations, separating two questions that are routinely conflated: does coordination help?

Table 10. Comparative Analysis of Single-Agent and Multi-Agent Architectures

Architecture	Representative Study/System	Coordination Mechanism	Benefits	Scalability	Limitations	Reference
Single agent	Tool-using single agents	None; internal loop only	Lowest communication cost; cleanest attribution	Bounded by context window	Accumulated execution errors over long horizons	(Zhao et al., 2026)
Independent ensemble	Independent configuration	Output concatenation without cross-validation	Isolates parallelism from coordination effects	Linear in agent count	No error correction; cost multiplies	(Y. Kim et al., 2026)

Critical interpretation. The table supports a narrower claim than the literature typically makes. Coordination is neither uniformly beneficial nor uniformly costly: the independent-ensemble condition shows parallelism alone can produce gains without communication, so studies comparing "multi-agent" to "single-agent" without an ensemble control cannot attribute improvement to coordination (Y. Kim et al., 2026).

Reasoning, Planning, Memory, and Decision-Making

Planning research has differentiated into strategies with distinct failure profiles. Task-decomposition planning breaks complex goals into manageable subtasks, reducing problem complexity but degrading when subtasks are highly interdependent and require careful sequential coordination; ReAct-style planning interleaves reasoning and action so plans are revised in real time, at the cost of dependence on accurate environment modelling; Reflexion-style planning adds feedback loops and self-correction; and explicit replanning improves robustness under uncertainty while frequent replanning introduces computational overhead (Chowa et al., 2026). Chain-of-thought is reported as effective in code generation and robotics contexts (Chowa et al., 2026).

Table 11 maps eleven techniques to technical principle, agentic function, reported outcomes and limitations, testing a hypothesis arising from the preceding subsections: that **reported outcomes are systematically confounded with benchmark and model choice**, so the literature supports statements about mechanism but not about comparative effectiveness.

Table 11. Comparison of Reasoning, Planning, and Memory Mechanisms

Technique	Representative Study	Technical Principle	Agentic Function	Reported Outcomes	Limitations	Reference
Chain-of-thought prompting	CoT line of work	Intermediate reasoning steps before answer	Multi-step reasoning substrate	Effective in code generation and robotics contexts	Effectiveness varies with task complexity and environment modelling	(Chowa et al., 2026)

Critical interpretation. No technique here has an outcome comparable to any other, because none is reported on the same benchmark, model and metric triple. The literature supports a **mechanism-level** conclusion — decomposition helps until interdependence dominates, reflection helps on repeated tasks, replanning helps under uncertainty at a cost — but not a **ranking**. Two further points deserve emphasis.

Agentic AI Frameworks and Development Platforms

Framework comparisons differentiate platforms by orchestration mechanism rather than by reasoning approach. LangChain performs prompt chaining, orchestrating linear sequences of LLM calls and API tools; AutoGen implements multi-agent conversation among collaborative agents; CrewAI applies role-based workflow by assigning roles and goals to a team and managing their interaction; Semantic Kernel composes plugins and pre-written code functions as skills; and LlamaIndex centres retrieval-augmented generation, injecting retrieved information into the model's context (AbouAli et al., 2025). A second strand adds explicit control of state: LangGraph provides a lower-level abstraction for stateful agentic applications with fine-grained control over an agent's behaviour (Satyadhar Joshi, 2025), and its runtime is described as a finite-state machine expressed as a graph with persistent nodes (Raza et al., 2026).

Table 12 compares twelve frameworks and platforms across architecture, tool integration, multi-agent support, memory/state, deployment features and limitations, and is explicit about provenance: entries derive from specifications and comparison reviews, because no controlled cross-framework benchmark currently exists.

Table 12. Comparative Evaluation of Agentic AI Development Frameworks

Framework	Architecture	Tool Integration	Multi-Agent Support	Memory/ State Support	Deployment Features	Limitations	Reference
LangChain	Prompt-chaining orchestration	Broad integration; retrievers and vector stores	Limited native support	Memory buffers; persistent stores	Large integration catalogue	Linear-call orchestration model constrains control flow	(AbouAli et al.,2025 ; Ehtesham et al., 2025)
LangGraph	Graph / state-machine orchestration	Custom modules as graph nodes	Supported through graph topology	Persistent nodes across steps	Visual orchestration of agent graphs	Structural complexity; steep learning curve	(S. Du et al., 2026; Satyadhar Joshi, 2025; Raza et al., 2026)

Critical interpretation. Three conclusions follow, all limitations of the evidence rather than of the frameworks. First, **the comparison is specification-based, not measurement-based**: no study in the corpus runs the same task suite across frameworks, so the matrix supports capability statements and not performance statements — a distinction that must survive into any downstream claim.

Autonomous Workflows and Tool Integration

Autonomous Task Planning and Workflow Execution

Workflow design in agentic systems is organised around how much of the plan is fixed before execution begins. At one pole, iterative control interleaves reasoning and action: planning decomposes goals into sub-tasks and executes them sequentially, with iterative planning permitting adaptation to environmental changes at the cost of dependence on accurate environment modelling (Chowa et al., 2026). At the other pole, the plan is committed and execution follows it, with decomposition reducing problem complexity while degrading when sub-tasks are highly interdependent and require careful sequential coordination (Chowa et al., 2026). Between them sit reflection-based workflows, which replan after observing outcomes, and adaptive planners that refine a plan both in-plan and out-of-plan so that actions stay aligned with predictions and respond to surprise (Chowa et al., 2026).

Table 13 compares twelve workflow models across architecture, planning strategy, execution model, reported performance and limitations. Only four rows carry a quantitative result, and the pattern of which four is itself the finding: the quantitative rows are all benchmark- or software-workflow studies, while the rows describing enterprise and industrial workflows report no performance figures at all.

Table 13. Comparative Analysis of Autonomous Workflow Models and Execution Strategies

Study/ System	Workflow Architecture	Pla nni ng Strategy	Execut ion Model	Applicatio n	Reported Performa nce	Limitation s	Reference
ReAct-style iterative loop	Single-agent sequenti al loop	Interlea ved reason–act	One model call per step	General tool use	NR in reviewed corpus	Latency and cost grow with step count; no parallelisati on	(L. Wang et al., 2024)
Decomp osition-based planner	Single-agent, staged	Goal broken into ordered sub-tasks	Sequential execution of sub-tasks	General agentic tasks	NR	Degrade s when sub-tasks are highly interdependent	(Chowa et al.,2026)

Critical interpretation. The table exposes a structural weakness in the workflow literature: **the models with the clearest architectures have the least performance evidence, and the models with performance evidence are the least architecturally ambitious**. The quantitative rows describe single-agent software workflows (Chen et al., 2026; Y. Zhang et al., 2024), while every multi-agent and governed workflow row reports NR.

Tool Use, API Integration, and Function Calling

Tool integration has become the site of the field's most concrete engineering problems, and the problems are schematic before they are cognitive. Tool definitions are typically static, so enabling an application to use an external tool requires implementing call handling and initialising tool definitions, and when a schema changes re-initialisation is required for the new version (Ding, 2025). The adapter burden this creates

scales with the product of agents and tools rather than their sum, which is precisely the motivation for protocol standardisation (Cabot, 2025).

Table 14 classifies twelve integration mechanisms by mechanism, interaction pattern, advantage and limitation, deliberately placing security and technical limitations in one column because in tool-integrated systems they are not separable.

Table 14. Tool Integration Techniques and Interoperability Mechanisms in Agentic AI

Tool/ Protocol	Integration Mechanism	Agent Interaction	Application	Advantages	Security/ Technical Limitations	Reference
Static tool definitions	Schema bound at initialisation; handler implemented per tool	Model selects function and fills arguments	General product APIs	Simple; deterministic contract	Schema change requires re-initialisation for the new version	(Ding,2025)

Critical interpretation. Three conclusions are supported. First, **the integration problem is combinatorial and standardisation is the only structural answer**: per-pair adapters scale with agents × tools, so protocol mediation is not a convenience but an economic necessity (Cabot, 2025; Ehtesham et al., 2025). Second, **the standardised solution relocates risk rather than removing it**: the moment tool sets are discovered at runtime, the authority boundary moves from the design document to the registry and the server, which is why privilege separation rather than detection has produced the most favourable security result in the corpus (Cheng & Tsao, 2026).

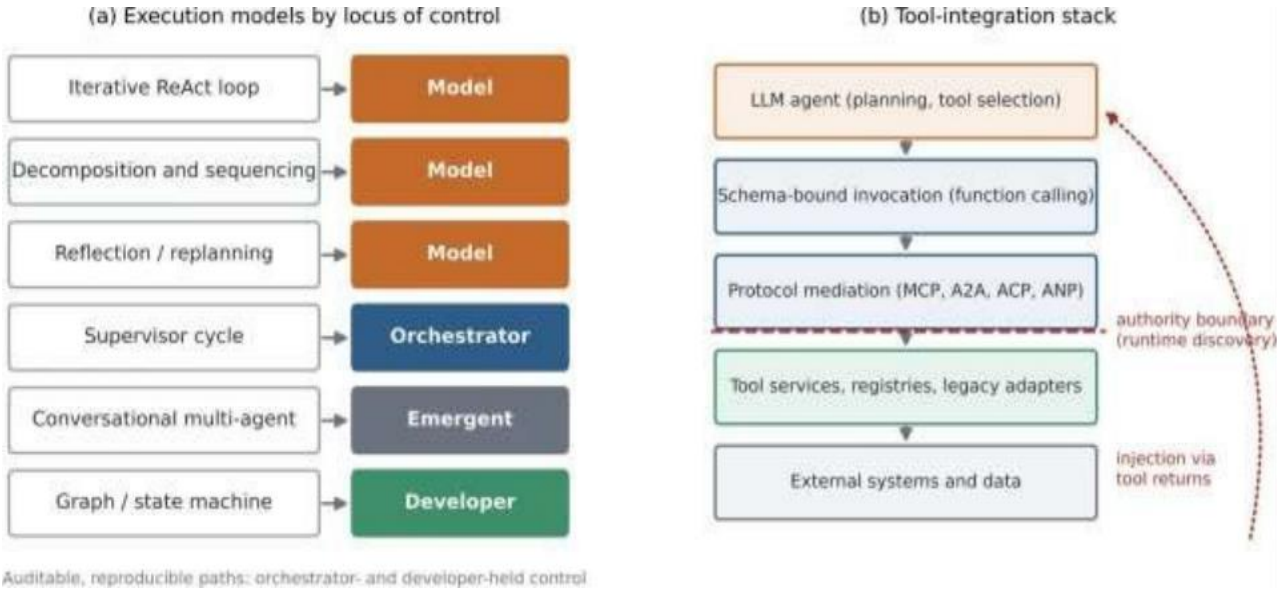


Figure 5 combines the workflow models of Section 4.1 with the integration stack analysed in this subsection.

Figure 5. Autonomous workflow execution and agent–tool integration. (a) Execution models classified by who holds control (model, orchestrator, emergent dialogue or developer); only orchestrator- and developer-held control yields reconstructable execution paths. (b) Integration stack from the agent through schema-bound invocation and protocol mediation to tool services; the authority boundary moves to runtime discovery, and adversarial content can re-enter the agent through tool returns.

Agent Orchestration and Multi-Agent Collaboration

Orchestration research has consolidated on the supervisor pattern, and the consolidation is better explained by human factors than by performance evidence. A supervisor decomposes high-level goals into sub-tasks and delegates them to role-specialised sub-agents, each executing a compact ReAct-style loop with optional evaluation and memory, while shared or federated state maintains coherence; the supervisor receives reports that trigger retries or re-planning when outcomes deviate (Bandi et al., 2025). The pattern is favoured because it reduces the cognitive burden on the end-user, abstracting away the complexity of interacting with multiple autonomous agents — managing a distributed system directly is cognitively demanding, whereas a supervisor can plan, delegate and coordinate internally (Schömbs et al., 2025).

Table 15 compares eleven orchestration frameworks and patterns by model, communication strategy, task allocation and documented limitation, distinguishing implemented frameworks from proposed architectures.

Table 15. Comparative Analysis of Agent Orchestration and Coordination Frameworks

Framework/ System	Orchestration Model	Communication Strategy	Task Allocation	Application	Strengths	Limitations	Reference
Supervisor /hierarchical	Central supervisor over role sub-agents	Report-and-assign dialogue	Supervisor-decided, optional shared state	General agentic applications	Reduces user cognitive burden; supports re-planning on deviation	Coordination overhead; supervisor is a single point of failure	(Bandi et al.,2025)
Group-chat conversation	Conversational multi-agent	Broadcast dialogue among agents	Emergent from conversation	General LLM applications	Flexible composition; reusable templates	High debugging cost; resource-intensive	(Wu et al.,2023)

Critical interpretation. The evidence supports a narrower conclusion than the design literature implies. Hierarchical orchestration is adopted for **human-centred reasons** — it presents a distributed system as one conversant interface — and that adoption preceded controlled comparison (Bandi et al.,2025; Schömbs et al., 2025). The empirical literature offers no case in which a supervisor pattern is shown to outperform a single agent on a matched task with reported uncertainty, and the design literature's own trade-off statement is symmetric: centralisation buys consistency at the cost of responsiveness, decentralisation buys responsiveness at the cost of coherence (Li et al., 2024).

Workflow Monitoring, Error Handling, and Recovery

Recovery research treats agent workflows as long-running transactions, and the dominant finding is that naive retry is harmful. Error accumulation across a reasoning chain can amplify problems until the quality of resultsis affected, so a robust mechanism for detection and correction is required — yet effective backtracking without crashing or stalling the pipeline is described as a major unresolved challenge (Shen,2024). This is a structural property of sequential chains rather than a defect of particular models: an error introduced upstream is inherited as a premise downstream.

Table 16. Workflow Failure Types, Monitoring Approaches, and Recovery Strategies

Failure Mode	Root Cause	Detection Mechanis m	Recovery Strategy	Supporti ng Evidence	Remainin g Limitation	Reference
Error accumulatio n in reasoning chains	Sequential dependenc y; each step consumes prior output	Intermediate -result validation	Backtracking and correction	Error accumulatio n can amplify problems along the chain	Effective backtracking without stalling remains a challenge	(Shen,2024)
Irreversible state change without counterp art	State-changing action issued with no compensa tion defined	Post-hoc state comparison	Compensati ng transaction paired with each instructio n	Pattern requires every instruction to have a reversible or idempotent counterpart	Not all external effects are compensab le	(Devarakonda,2026)

Critical interpretation. The table supports three conclusions. First, **recovery is a transaction-design problem before it is a model problem**: the only mechanism in the corpus with a specified correctness condition is the compensating transaction, because it constrains what an agent may do rather than how it should recover (Devarakonda, 2026). Second, **detection is limited by trace fidelity**, and trace fidelity is limited by cost, so improving recovery is not an algorithmic problem but an instrumentation-budget allocation problem (Devarakonda, 2026).

Real-World Applications of Agentic AI

Every row of Tables 17–20 is labelled by validation tier: production deployment, field trial, laboratory prototype or experiment, simulation, benchmark-only, or conceptual/review. Where fewer than ten defensible rows exist, the shortfall is reported and no adjacent-domain study is substituted. Cells marked NR denote a datum not reported in the verified evidence base.

Healthcare and Intelligent Decision Support

Healthcare provides the clearest evidence that the gap between agentic capability and clinical deployment is structural rather than technical. A comparison of agent systems on clinical decision tasks concludes that current agentic designs offer modest performance benefits at significant computational and workflow cost, and calls for more accurate, efficient and clinically viable agent systems (Y.Liu et al.,2026). That conclusion is the more telling because the same literature documents why agent support was pursued: initial machine-learning decision support was narrow, domain-shifted and fragile, and most

tools supported isolated actions rather than multi-step clinical reasoning, constraining real-world clinical integration (Y. Liu et al., 2026).

Table 17. Agentic AI Applications in Healthcare: Methods, Results, and Limitations

Author/ Year	Healthcare Application	Agent Architecture	Data/Environment	Validation Type	Reported Outcome	Limitations	DOI
Y. Liu et al. (2026)	Clinical decision tasks	Multiple agent designs compared	Clinical decision benchmarks	**Benchmark-only**	Modest performance	Benchmarks differ from	10.1038/s41746-026-02443-6 (Y. Liu et al. (2026))

Critical interpretation. The healthcare evidence is homogeneous in an informative way: every row sits at benchmark, prototype or review tier, and none reports a deployed clinical agent. The most defensible reading is that the field has correctly identified its evaluation problem and not yet solved its deployment problem — the FHIR-compatible environment exists precisely so that agents can migrate into live systems (Y. Jiang et al., 2025), but migration has not been reported.

Finance and Business Process Automation

Finance is the domain in which the definition of agentic AI does the most analytical work, because the move from advising to acting changes the risk surface, the control architecture and the accountability expectations (Mustafa & Aysan, 2026). Agentic deployment is defined by three properties — operating as goal-directed agents, integrating with enterprise tools and workflows, and exercising delegated action rights under specified constraints — and the definition is explicitly contrasted with generation: generative models may be a component of an agentic system, but agentic deployment is about execution rather than text generation per se (Mustafa & Aysan, 2026).

Table 18 collects the finance and business-process rows, of which most are conceptual or review-tier. This distribution is the finding: the domain with the highest volume of agentic claims has the lowest volume of measured outcomes in the verified corpus.

Table 18. Comparative Evidence on Agentic AI in Finance and Business Automation

Study	Industry/ Process	Agentic Method	Automation Task	Validation Setting	Performance Evidence	Limitations	Reference
Mustafa & Aysan (2026)	Banking, insurance, exchanges	Adoption and control framework	Delegated action rights under constraints	**Conceptual framework**	Agentic defined by goal-directed operation, enterprise integration and delegated action rights	No deployment data; instrument proposed, not applied	10.61351/mf.v4i1.557 (Mustafa& Aysan,2026)

Study	Industry/ Process	Agentic Method	Automation Task	Validation Setting	Performance Evidence	Limitations	Reference
Kou et al. (2026)	RegTech compliance	Bounded agent with permissions and approval gates	Regulatory monitoring, obligation mapping, gap analysis, task creation	**Conceptual design**	Approval gates, grounding and action logs specified as boundary conditions	Static-benchmark inadequacy asserted; no empirical test	10.1186/ s40854-026-00941-w (Kou et al., 2026)

Critical interpretation. The finance evidence supports a claim that healthcare cannot: **the domain has already articulated the correct evaluation question and has not yet answered it**. Process-correctness evaluation — tool selection, constraint respect, state maintenance, approval requests, action logging, error recovery — is a complete specification of what a deployable finance agent must do (Kou et al., 2026), and no row in the table reports results against it.

Smart Manufacturing, Robotics, and Industrial Systems

Industrial agentic research is the corpus's strongest evidence of a functional deployment and its weakest evidence of generalisable performance. Digital-twin integration supplies the clearest worked case: an architecture combining IIoT sensing, a digital twin and LLM integration uses four specialised agents — supervisor, machine-expertise, visualisation and fault-diagnosis — to bridge legacy and smart factory machines (Gautam et al., 2025). The design choice is deliberate: the LLM is one component among several, and specialised agents carry domain functions rather than a single general agent covering them all.

Table 19. Agentic AI Applications in Manufacturing, Robotics, and Industrial Systems

Study / System	Industrial Application	Agent Architecture	Integrated Technologies	Validation Type	Reported Outcomes	Challenges	Reference
Gautam et al. (2025)	Legacy and smart factory machines	Four specialised agents: supervisor, machine expertise, visualisation, fault diagnosis	IIoT sensing, digital twin, LLM integration	**Laboratory prototype**	Functional twin bridging legacy and smart equipment; no comparative performance reported	No quantitative benchmark; single configuration	10.1016/j.jmsy.2025.03.022 (Gautam et al., 2025)
D. Gao et al. (2026)	Flexible manufacturing digital twin	LLM-powered multi-agent twin	Self-evolving twin on an experimental sorting line	**Laboratory experiment**	47% reduction in changeover and recovery times vs. conventional methods	Single line; no variance or replication reported; plant-scale transfer unverified	10.1016/j.dte.2026.1 0009 9 (D. Gao et al., 2026)

Critical interpretation. Industrial evidence is the corpus's best answer to whether agentic architectures work in real systems, and the answer is conditional: the deployed cases are **component-integrated**, with the LLM as one agent among specialised others and a digital twin or instrument providing the ground truth (M. Du et al., 2026; Gautamet al., 2025). The pattern is consistent with the review's wider finding that autonomy scales inversely with the cost of an incorrect action.

Cybersecurity, Software Engineering, and Enterprise Automation

Software engineering is the domain with the largest volume of reproducible agentic results and the sharpest demonstration of a capability boundary. Augmenting LLMs with code search allows a search-based agent to locate and fix issues at repository level, achieving 19% on a lite benchmark subset within four minutes (Y. Zhang et al., 2024). Experience-driven approaches raise this substantially by accumulating prior trajectories, reporting 73% pass@1 on a verified benchmark with a frontier model (Chen et al., 2026); because the two figures come from different subsets they are not comparable, and the experience store introduces a contamination channel that a static benchmark cannot detect.

Table 20. Agentic AI Deployments in Cybersecurity, Software Engineering, and Enterprise Systems

Author/ Year	Domain	Agent/ System	Workflow	Tools Integrated	Evaluation Evidence	Results	Limitations	DOI
Y. Zhang et al. (2024)	Software engineering	AutoCodeRover	Fault localisation → code search → patch → test	Code search API, test runner, AST analysis	**Benchmark-only**	19% on a lite subset within 4 minutes	Lite subset; single snapshot	10.1145/3650212.3680384 (Y. Zhang et al., 2024)
Chen et al. (2026)	Software engineering	SWE-Exp	Retrieve prior trajectory → apply → repair	Experience store, test execution	**Benchmark-only**	73% pass@1 with a frontier model	Experience store may leak test information; subset differs from lite	10.2139/ssrn.7234653 (Chen et al.,2026)

Critical interpretation. Software engineering is the only domain in the corpus with a large, reproducible and internally consistent evidence base, and it defines the boundary precisely: **capability is strong on localised, verifiable changes and collapses on multi-file, long-horizon work** (L. Zhang et al., 2025). Two consequences follow. First, agent adoption in software engineering creates a maintenance debt in proportion to its success, because agents perform markedly worse on agent-specific issues than on conventional ones (Rahardja et al., 2025).

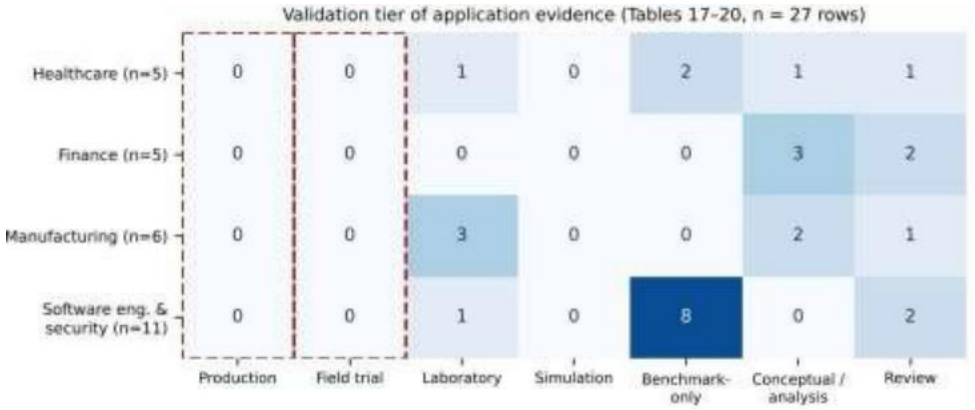


Figure 7 cross-tabulates the 27 application rows of Tables 17–20 by domain and validation tier.

Figure 7. Real-world application taxonomy by domain and validation tier (n = 27 rows from Tables 17–20, coded with the validation-tier rule of Section 2.3). The dashed outline marks the production and field-trial columns, which are empty for every domain in the verified corpus.

Performance Evaluation and Benchmarking

Evaluation Frameworks and Benchmark Datasets

Evaluation infrastructure has become the field's most rapidly maturing layer, and it is organised by environment rather than by capability. Four families are distinguishable on that basis: executable software environments, in which an agent edits repositories and is scored by test execution; interactive digital environments, in which an agent operates a browser, desktop or operating system; policy-constrained dialogue environments, in which an agent interacts with a simulated user under procedural rules; and knowledge-and-tool environments, in which success depends on retrieval and multi-step information seeking. A fifth family — domain-specific clinical, enterprise and security environments — has emerged most recently.

Table 21 catalogues fourteen environments across the five families, recording the evaluated capability, task characteristics, metric and limitations. Where an environment's task count or scoring detail was not available in the verified corpus, the cell is marked NR rather than reconstructed from general knowledge.

Critical interpretation. Three findings follow. First, **environment design determines the ceiling**, and the ceilings are inconsistent across families: human reference performance is 92% on GAIA (Mialon et al., 2023) but 54% on a 2025 web-search benchmark (Gou et al., 2025) and 72.36% on an operating-system benchmark (T.Xie et al., 2024). A single "agent performance is X% of human" statement is therefore meaningless without its environment.

International Journal of

Table 21. Major Agentic AI Benchmark Datasets and Evaluation Environments

Benchmark	Year/ Version	Evaluated Capability	Environment	Task Characteristics	Evaluation Metric	Limitations	Original Reference
WebArena	2024 (ICLR)	Web navigation and task completion	Simulated web	812 long-horizon tasks on self-hosted websites	Functional task success rate	Found vulnerable to gaming, undermining construct validity	ICLR 2024 (Zhou et al., 2024); gaming critique in Peykani et al. (2026)
OSWorld	2024	Open-ended computer use	Real computer environments	Open-ended desktop tasks	Task success	Human ceiling of 72.36%, leaving substantial headroom	10.52202/079017-1650 (T.Xie et al., 2024)
SWE-bench	2023/2024	Repository-level issue resolution	Executable code repositories	2,294 GitHub issues	Percentage of issues resolved	Earliest models scored very low (Claude 2: 1.96%), creating floor effects	ICLR 2024 (Jimenez et al.,2024)

Task Success, Accuracy, and Reliability Metrics

Metric definitions in agentic evaluation are not interchangeable, and the confusion is a definitional problem before it is a measurement problem. Reviews of agentic evaluation distinguish consistency and robustness as separate objects of measurement, and locate the rationale for repeated-trial scoring precisely in the observation that a system succeeding once in k attempts has not demonstrated reliability (Mohammadi et al., 2025). This review accordingly formalises eleven metrics explicitly, with numbered equations, so that every figure in Table 24 attaches to exactly one definition.

Table 22 maps each metric to its evaluation objective, measurement procedure, interpretation and limitations. The table's analytic function is to make explicit which metrics measure capability and which measure reliability, because the two answer different deployment questions and diverge sharply under repetition.

Table 22. Task Performance, Accuracy, and Reliability Metrics for Agentic AI

Metric	Mathematical Definition	Evaluation Objective	Measurement Procedure	Interpretation	Limitations
Task success rate	Eq. (1)	End-state achievement	Score each task once against specified end state	Proportion of tasks completed	Ignores how the end state was reached
Resolve rate pass@k pass^k	Eq. (2) Eq. (3) Eq. (4)	Issue or bug resolution Capability under sampling Reliability across repeated trials	Execute test suites after patch generation k independent attempts per task k trials per task; success required on all	Proportion of issues resolved Upper bound on attainable capability Proportion solved on every trial	Test adequacy bounds validity Rewards sampling budget, not dependability Measurement cost scales with k; collapses on long horizons

Critical interpretation. The definitional audit yields three conclusions that constrain every later quantitative claim. First, **pass@k and pass^k measure opposite properties** and both are legitimately called "performance": the first rises with sampling budget, the second falls with trial count, and a system can improve on one while degrading on the other. Reported degradation of reliability across increasing on a policy-constrained suite (Fang et al., 2025) alongside rising single-trial pass rates elsewhere is therefore consistent rather than contradictory, and any synthesis that averages them is invalid.

Latency, Computational Efficiency, Cost, and Scalability

Efficiency is the least reported dimension in the corpus and the one with the most consistent direction of finding: agentic architectures are substantially more expensive than single-call generation, and the multiple is measured in multiples rather than percentages. An industry engineering report estimates that single agents consume about four times, and multi-agent systems about fifteen times, the tokens of a chat interaction (Hadfield et al., 2025), and peer-reviewed evaluation in clinical decision tasks reports more than 10× token usage with more than 2× latency relative to baseline models (Y.Liu etal., 2026). The two estimates come from different settings and only one is peer-reviewed, but they agree in direction and order of magnitude.

Table 23 assembles the efficiency evidence across ten studies. The column structure deliberately separates reported cost from scalability evidence, because several studies report scale behaviour without reporting cost and vice versa.

Table 23. Computational Efficiency, Operational Cost, and Scalability Evaluation

Study/ System	Task/ Benchmark	Hardware/Model Setting	Latency	Token/ Compute Usage	Reported Cost	Scalability Evidence	Reference
Dynamic-reasoning cost analysis	ReAct-style agent workloads	NR	Latency variance widens with added test-time compute	NR	NR	Diminishing accuracy returns per unit of compute	(J.Kim et al., 2026)
Constrained placement agents	Cloud container placement	Single GPU	Approximately 90 minutes end-to-end for 100 decisions	Approximately 2 million tokens per 100 decisions	NR	Uneconomic for high-frequency loops	(Sarkar et al., 2026)

Critical interpretation. Three conclusions are defensible. First, the efficiency penalty of agency is large and directionally consistent: an industry estimate of roughly 15× token consumption for multi-agent systems (Hadfield et al., 2025) and a peer-reviewed clinical measurement of >10× token usage with >2× latency (Y.Liu et al., 2026) arise from different settings, and the penalty follows from the loop itself rather than from implementation inefficiency.

Comparative Performance Analysis of Existing Agentic AI Systems

Comparison is admissible only within benchmark families, and the corpus supplies enough within-family evidence to make the constraint concrete. Within the repository-level family, reported resolution rates span 1.96% for an early model on the original benchmark (Jimenez et al., 2024), 19% for a search-augmented agent on a lite subset within four minutes (Y. Zhang et al., 2024), 48% for single-file patches under five lines and below 10% for patches touching three or more files or exceeding 100 lines (L. Zhang et al., 2025), and 73% pass@1 for an experience-driven agent on a human-validated subset with a frontier model (Chen etal., 2026). The union across ten independent agents exceeds the average single-agent resolve rate by more than 1.5× in all four reported settings and by more than 2× in two of them (K. Zhang et al., 2024).

Table 24 is therefore organised by benchmark family, with the family as a grouping dimension rather than a column, and records evaluation setting, metric, reported score and limitations.

Table 24. Empirical Performance Comparison of Existing Agentic AI Systems

Agent/ System	Base Model	Benchmark Version	Evaluation Setting	Metric	Reported Score	Evaluation Date/Source	Limitations
Repository-level software family							
Claude 2 baseline	Claude 2	SWE-bench original	Benchmark-only	Resolve rate	1.96% of 2,294 issues	Author-reported (Jimenez et al.,2024)	Floor effect; historical baseline only

Critical interpretation. The table supports four conclusions and forecloses several others. First, **the largest variation within any family is produced by evaluation configuration, not by system capability**: the repository-level range spans subsets, dates and metrics simultaneously, so the corpus cannot support a ranking of software agents, and no such ranking is offered here. Second, **human reference points are environment-specific and must not be concatenated**: 54%, 72.36% and 92% describe three different task populations (Gou et al., 2025; Mialon etal.,2023; T.Xie etal.,2024).

Technical Challenges and Deployment Limitations

Security, Privacy, and Data Protection Challenges

Agentic systems fail security in a way that generative systems do not: the attack does not need to make the model produce a harmful string, only to make it invoke a legitimate tool with illegitimate arguments. This structural shift is captured by framing agents as systems holding **delegated action rights under constraints** (Mustafa& Aysan,2026) — the security question becomes what those rights permit, not what the model will say. Surveys of autonomy-induced risk in large-model agents converge on the same diagnosis: as autonomy increases, the attack surface migrates from content generation to decision-making and tool invocation (Su et al.,2026).

International Journal of Advanced Research

Table 25. Security Vulnerabilities, Threat Mechanisms, and Mitigation Strategies

Threat	Attack Surface	Technical Mechanis m	Potential Impact	Mitigation	Supporting Study/Stan dard	Residual Limitation
Indirect prompt injection	Retrieved documents, tool returns, web content	Instruction s embedded in consumed content are treated as directives	Unauthori sed tool invocation; data exfiltration	Input provenance tagging; structural privilege separation	InjecAgent benchmar k, 1,054 cases (Zhan et al., 2024)	Benchmark uses simulated tools; production transfer untested
Adaptive evasion of defences	Defence pipeline itself	Attacker optimises payloads against known defences	Systematic defence bypass	Architectural restriction rather than content inspection	Adaptive attack study (Zhan et al., 2025)	Attack success above 50% under adaptive conditions; effectiveness conditional on attacker model

Critical interpretation. Three conclusions are supported and one widely held assumption is not. First, **the mechanism of attack has shifted from content to authority**: every row in which the impact is severe involves the agent performing a legitimate operation with illegitimate parameters, which is why content-level defences underperform (Shi et al., 2026; Zhan et al., 2025). Second, **the only mitigation class with a favourable measured result restricts what the agent may do rather than what it may read** (Cheng& Tsao,2026), and this asymmetry is consistent across the corpus.

Hallucinations, Error Propagation, and System Failures

Agentic failure differs from generative failure in two respects: errors propagate through a workflow rather than terminating at the output boundary, and the propagation is asymmetric because later stages inherit earlier outputs as premises. Review evidence documents error accumulation along reasoning chains in which problems amplify until result quality is affected, and identifies effective backtracking without crashing or stalling the pipeline as an unresolved challenge (Schilling-Wilhelmi et al., 2024). The mechanism is not model-specific: any sequential composition of model calls inherits this property. Table 26 organises twelve failure types by cause, workflow consequence, detection method, mitigation and supporting evidence. The table's analytic function is to separate failures that are **detectable and recoverable** from those that are **neither**, because the second class determines whether autonomous operation is defensible.

Table 26. Failure Modes, Error Propagation, and Reliability Challenges

Failure Type	Underlying Cause	Workflow Consequence	Detection Method	Mitigation Strategy	Supporting Evidence
Error accumulation in reasoning chains	Sequential dependency; each step consumes prior output	Progressive degradation of result quality	Intermediate-result validation	Backtracki ng and correction	Error accumulation can amplify problems along the chain (Schilling-Wilhelmi et al., 2024)
Hallucinatio n propagation across agents	Shared context consumed as accepted premise	One error becomes every agent's belief	Provenance inspection	Quarantine and re-derivation	Cooperation amplifies hallucination through inter-agent propagation (Deng et al., 2025)
Long-horizon execution failure	Error compoundin g across many steps	Task abandoned or completed incorrectly	Step-level success tracking	Stepwise verification; early termination	Every multi-agent variant tested degraded sequential-reasoning performance by 39–70% (Y. Kim et al., 2026)

Critical interpretation. The table supports a sharper claim than the general observation that agents fail. **The failures most likely to cause harm are the ones least amenable to detection**: trace incompleteness, dynamical instability, judge mis-scoring and unanticipated failure classes share the property that the workflow cannot itself reveal that it has failed (Darabi et al., 2026; Devarakonda,2026; Reid et al., 2025).

Interoperability, Scalability, and Infrastructure Constraints

Interoperability is the constraint that most clearly distinguishes agentic from generative deployment, because an agent must bind to systems it did not design. The base problem is schematic: tool definitions are typically static, so enabling an application to use an external tool requires implementing call handling and initialising tool definitions, and a schema change requires re-initialisation for the new version (Ding,2025). Adapter cost therefore scales with the product of agents and tools rather than their

sum, which is the precise motivation for the M×N integration argument and for protocol mediation as its structural answer (Cabot, 2025).

Table 27. Deployment Barriers, Interoperability Challenges, and Infrastructure Requirements

Constraint	Technical Origin	Affected Component	Deployment Consequence	Existing Solution	Remaining Research Gap	Reference
Legacy systems without API access	Mainframe and on-premise systems expose no programmatic interface	Tool integration layer	Agent cannot act; scope limited to what is exposed	Adapter and RPA-style bridging	No standard for stateful legacy transactions	(Devarakonda,2026)
Heterogeneous API schemas	Provider-specific request and response formats	Tool definitions	Separate code path per provider	Static tool definitions	Re-initialisation required on every schema change	(Ding,2025)

International Journal of Advanced Research

Trustworthiness, Human Oversight, and Governance Challenges

Governance research on agentic systems has converged on a small set of mechanisms while remaining uncertain about their effectiveness. The strongest normative claim in the corpus is that evaluation itself must change: a static benchmark asks whether a model answers correctly, whereas an agent benchmark must ask whether the system selects the right tools, respects constraints, maintains state, asks for approval when needed, records its actions and recovers from errors — so process correctness, not only final-answer correctness, is the object of assessment (Kou et al., 2026). This reframes governance from an external constraint into an evaluation criterion, which is the review's principal bridge between Sections 6 and 7.

Table 28. Trustworthiness, Human Oversight, and Governance Mechanisms

Governance Dimension	Associated Risk	Oversight Mechanism	Implementation Approach	Relevant Standard/Study	Limitations
Transparency	Reasoning and action trace is opaque to overseers	Decision logging with rationale capture	Structured logs recording tool calls, arguments and stated rationale	Trust, risk and security management review (Raza et al., 2026)	Logs record outputs, not internal justification
Accountability	Responsibility diffused across agents	Named owner per workflow	RACI-style mapping of agent roles to institutional owners	Ethics review of multi-agent clinical systems (Z. Xie et al., 2026)	No mechanism assigns obligation to a non-human actor

Critical interpretation. The table yields one central and uncomfortable conclusion: **the governance mechanisms recommended in the literature are systematically stronger than the mechanisms shown to be implementable.** Immutable role-granular logging is recommended

as a control (Raza etal., 2026; Z. Xie et al., 2026), while complete trace capture is reported as infeasible at scale with unvalidated surrogates as the substitute (Devarakonda, 2026); independent audit is proposed as the accountability backstop (Falco et al., 2021), yet it depends on records whose completeness is precisely what is contested.

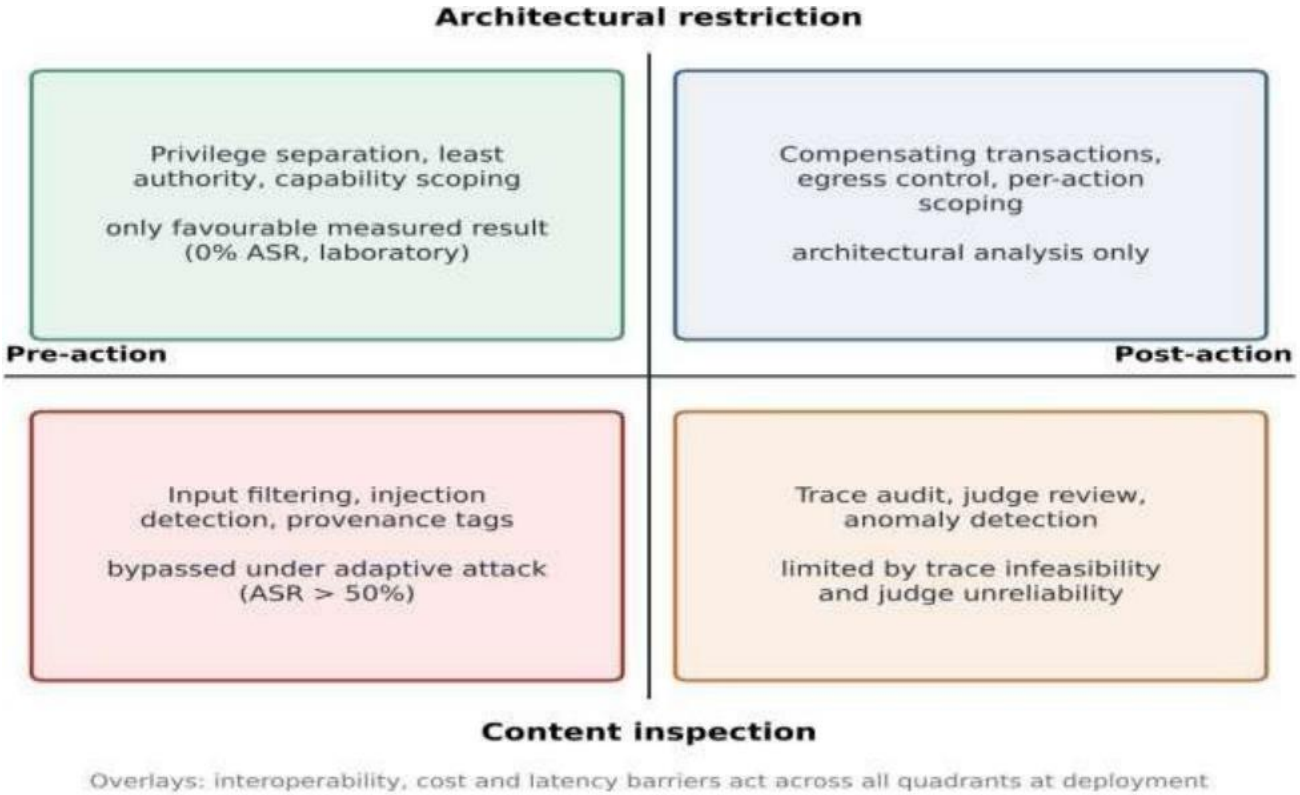


Figure 9 organises the security and governance controls of Tables 25 and 28 by timing and mechanism.

Figure 9. Security and deployment-challenge taxonomy. Controls are arranged by timing (pre-action versus post-action) and mechanism (architectural restriction versus content inspection). Only the pre-action architectural quadrant has a favourable measured result in the corpus (Cheng & Tsao, 2026); content inspection is bypassed under adaptive attack (Zhan et al., 2025). Interoperability, cost and latency act as deployment-stage overlays across all quadrants.

Research Gaps and Future Directions

Critical Analysis of Existing Research

Read together, Sections 3–7 show an evidence base that is broad in what it describes and thin in what it measures. The 80 sources cover every layer of the agentic stack, but measured outcomes concentrate in a few places. Repository-level software engineering (Jimenez et al., 2024; L. Zhang et al., 2025; Y. Zhang et al., 2024), policy-constrained dialogue (Barres etal., 2025; Yao etal., 2024), web and computer-use environments (Gou et al., 2025; T.Xie etal.,2024; Zhou et al., 2024) and prompt-injection security (Shi et al.,2026; Zhan et al., 2024, 2025) have defined metrics and reproducible environments. Architecture, orchestration, governance and most domain applications are described through specifications, conceptual frameworks and reviews (Bandi et al., 2025; Kou et al., 2026; Mustafa & Aysan, 2026; Raza et al., 2026).

Table 29. Comparative Synthesis of Existing Agentic AI Research and Limitations

Research Theme	Representative Studies	Main Contributions	Evidence Strength	Methodological Weakness	Identified Gap
Single- vs multi-agent architecture	Y. Kim et al. (2026); Li et al. (2024); B. Liu et al. (2025)	Controlled ablation of five topologies; topology trade-off vocabulary	Moderate (one large controlled study)	Most topology claims lack an independent-ensemble control	Coordination gain separated from parallelism in only one study
Reasoning, planning and memory	Chowa et al. (2026); Zeyu Zhang et al. (2025); Zhao et al. (2026)	Taxonomies of planning strategies and memory mechanisms	Low	No technique reported on a shared benchmark–model–metric triple	No comparative effectiveness ranking is possible

Critical interpretation. Two patterns stand out. First, evidence strength is inversely related to deployment proximity: the themes closest to production (governance, finance, orchestration and recovery) carry the weakest evidence, and the themes with the strongest evidence (software benchmarks and injection security) are the furthest from operational conditions. Second, several influential claims in the wider literature rest on sources from venues with limited peer-review transparency, notably on protocol integration and enterprise trace capture (Ayyagari, 2025; Chugani, 2026; Devarakonda, 2026).

Unresolved Challenges in Real-World Deployment

The challenges that remain unresolved are not the ones the field is working hardest on. Reported repository-level results now reach 73% pass@1 on a human-validated subset (Chen et al., 2026), against 1.96% for an early model on the original benchmark (Jimenez et al., 2024); the figures are not directly comparable, but they show how quickly reported capability on localised tasks has moved. Over the same period, the properties that decide whether an agent can be trusted with delegated action rights have barely been measured: reliability under repetition, behaviour under adaptive attack, cost per completed task, and the integrity of approval and audit controls.

Table 30. Unresolved Technical Challenges in Real-World Agentic AI Deployment

Challenge	Existing Approaches	Evide nce of Limitation	Practical Consequence	Research Need	Supporting Reference
Absence of production evidence	Benchmarks; laboratory prototypes; conceptual frameworks	0 of 27 application rows report a production deployment with outcomes	Deployment decisions rest on benchmark extrapolation	Prospective field studies with pre-registered outcome measures	Tables 17–20; Mehandru et al. (2024)
Long-horizon and multi-file tasks	Decompositio n; supervisor orchestration	Success below 10% for patches touching ≥3 files or >100 lines	Agents confined to localised, verifiable changes	Planning methods evaluated on interdepende nt tasks	L. Zhang et al. (2025); Y. Kim et al. (2026)

Critical interpretation. The challenges fall into two groups. Six concern measurement (production evidence, reliability, cost, judge reliability, construct validity and definitions) and could be addressed with existing methods if reporting norms changed. Six concern assurance (adaptive attack, authority over discovered tools, trace completeness, error propagation, gate integrity and the long-horizon boundary) and need new technical work.

Emerging Technologies and Research Opportunities

Several technical directions in the corpus address the challenges above, but they are at very different stages of maturity. Table 31 records, for each emerging technology, its current development, the evidence available for it, and the limitation that currently stops that evidence from supporting deployment. Two directions stand out because they work by constraining the agent rather than by improving it. Structural privilege separation achieved 0% attack success for a full pipeline in a laboratory evaluation (Cheng & Tsao, 2026), and compensating transactions give recovery a correctness condition (Devarakonda,2026). By contrast, directions that improve capability, such as experience reuse (Chen et al., 2026) and environment scaling (Fang et al., 2025), report benchmark gains without the reliability, cost or contamination controls that would show those gains transfer.

Table 31. Emerging Agentic AI Technologies and Research Opportunities

Emerging Technology	Current Development	Potential Contribution	Available Evidence	Research Limitation	Future Opportunity	Reference
Agent interoperability protocols (MCP,	Competing specifications with distinct discovery	Reduce M×N adapter cost	Descriptive surveys and architectural analyses	No conformance or security testing	Cross-protocol conformance suites	Ehtesham et al. (2025); Ayyagari (2025)
ACP, A2A, ANP)	and transport models					
Experience-driven memory	Reuse of prior repair trajectories	Higher repository-level resolution	73% pass@1 on SWE-bench Verified	Possible absorption of test information	Contamination-controlled evaluation	Chen et al. (2026)
Environment scaling for agent training	Training across many simulated environments	General agentic capability	Benchmark gains; pass^k profile reported	Compute cost; transfer unverified	Transfer studies to unseen real tools	Fang et al. (2025)
Live and refreshed benchmarks	Continuously updated issue streams	Resistance to memorisation	Complexity-stratified resolution rates	Scores unstable across dates	Versioned reporting with date stamps	L. Zhang et al. (2025)

Critical interpretation. The table separates technologies that change what can be measured (live benchmarks, reliability metrics, judge agreement) from technologies that change what an agent is allowed to do (privilege separation, compensating transactions, capability-scoped protocols). The second group has the stronger link to deployment risk but the thinner evidence: one preprint and one architectural analysis.

Future Roadmap for Reliable and Scalable Agentic AI

The roadmap follows from three observations made earlier in the review: comparison is meaningful only within benchmark families (Section 6.4), the controls with the best measured results restrict authority (Section 7.1), and deployment-relevant metrics are the least reported (Section 6.3). On that basis the review proposes an integrated evaluation perspective (Figure 10). It is offered as a reporting framework, not as a validated instrument. An agentic system would be reported on six dimensions at once, each tied to a metric already defined in Section 6.2: task performance (task success rate or resolve rate, Eqs. 1–2); reliability (pass^k and run-to-run variance, Eq.

Table 32 converts the gaps in Tables 29–30 into ten research priorities, ordered by the reviewers' judgement of how much each would improve the field's ability to make defensible deployment claims.

Table 32. Prioritised Future Research Roadmap for Real-World Agentic AI

Priority	Basis	Supportin g Evidence	Proposed Research Question	Potential Method	Evaluation Metric	Expected Scientific Contribution
1. Joint reporting standard	Evidence-based	Cost, variance and validation tier rarely reported (Tables 22–24)	Does joint reporting change the ranking of agent designs?	Re-evaluation of published agents under a six-dimension protocol	Rank stability across dimensions	Comparable, deployment-relevant evidence base
2. Reliability - first benchmar king	Evidence-based	Reliability decays with k (Yao et al.,2024 ; Barres et al.,20 25)	How large is the gap between pass@k and pass^k across benchmar k families?	Repeate d-trial evaluati on with fixed seeds and budgets	pass^k; run-to-run variance	Separation of peak capability from dependability

Critical interpretation. The roadmap deliberately places reporting and evaluation reforms (priorities 1–3) ahead of new capability. The reasoning is practical: without joint reporting, reliability metrics and adaptive-attack testing, gains from priorities 4–10 could not be verified, and the field would repeat the current pattern in which benchmark progress is read as deployment readiness.

Conclusion

Summary of Major Findings

This review set out to separate what agentic AI has demonstrated from what it has delivered in real systems. The evidence supports a clear but limited answer. Agents perform well on localised, verifiable tasks in executable environments. Their reliability, cost, security and governance in operational settings are mostly unmeasured. Table 33 consolidates the principal finding for

each dimension of the review, with its supporting evidence and the limitation that bounds it.

Table 33. Consolidated Summary of Major Review Findings

Research Dimension	Principal Finding	Supporting Evidence	Limitation	Overall Implication
Architecture	Coordination gains are task-contingent and shrink as base models improve	Y. Kim et al. (2026)	One large controlled study	Multi-agent design needs justification per task
Reasoning and memory	Mechanisms are well described but not comparatively evaluated	Chowa et al. (2026); Zhao et al. (2026)	No shared benchmark–model–metric triple	No technique ranking is defensible
Workflow execution	Clear architectures lack outcomes; measured workflows are single-agent	Tables 13 and 15	Specification-heavy literature	Workflow choice rests on plausibility
Tool integration	Integration is combinatorial; protocols relocate rather than remove risk	Ehtesham et al. (2025); Cabot (2025)	No protocol conformance or security tests	Interoperability is an authority problem
Applications	No production deployment with outcomes among 27 coded cases	Tables 17–20	Single-index corpus	Deployment claims outrun evidence
Benchmarking	Configuration drives more variance than system design	Tables 21 and 24	Author-reported values only	Rank only within benchmark families
Reliability	Repeated-trial reliability falls below single-trial capability	Yao et al. (2024); Barres et al. (2025)	pass^k rarely reported	Report pass^k and variance
Efficiency	Order-of-magnitude token and latency overhead	Y. Liu et al. (2026); Hadfield et al. (2025)	Industry estimate not peer-reviewed	Cost per success is the missing metric
Security	Detection fails under adaptive attack; authority restriction works in the laboratory	Zhan et al. (2025); Cheng & Tsao (2026)	Laboratory conditions; one preprint	Prefer architectural restriction

Critical interpretation. The findings are consistent across dimensions in one respect: wherever measurement exists, it shows narrower capability and higher cost than the descriptive literature implies. That consistency strengthens confidence in the overall conclusion, even though individual findings rest on modest evidence.

Theoretical and Practical Implications

For theory, the review suggests that agentic AI should be studied as a problem of delegated authority under uncertainty rather than as an extension of generative modelling. Many of the field's central difficulties appear in that framing as one problem: verifiability lost at the transition from symbolic to neural agents (Dennis& Fisher, 2020), errors inherited across agents (Deng

etal.,2025), and attacks that work through legitimate tools (Shiet al., 2026). For practice, the most robust guidance is architectural: restrict what agents may do, require approval only for irreversible actions, and measure reliability and cost before scaling. Table 34 sets out the implications for each stakeholder group.

Table 34. Theoretical and Practical Implications of Real-World Agentic AI Research

Stakeholder/Research Area	Key Finding	Theoretical Implication	Practical Implication	Supporting Evidence
Agent researchers	Coordination benefit is conditional on task structure and model capability	Treat topology as a task-dependent variable, not a design default	Include independent-ensemble and single-agent controls	Y. Kim et al. (2026)
Benchmark designers	Construct validity and versioning dominate reported variance	Benchmarks are measurement instruments that need validation	Publish versions, dates, trial counts and variance	Peykani et al. (2026); L. Zhang et al. (2025)

Critical interpretation. The implications divide stakeholders into those who can act on current evidence (security engineers, system developers and industrial engineers, who have architectural guidance) and those who must wait for better evidence (healthcare and finance, where outcome data do not yet exist). For the second group the honest recommendation is to evaluate, not to deploy.

Concluding Perspectives

Agentic AI has moved from demonstration to operational trial faster than its evaluation methods have matured. This review's contribution is mainly methodological: a taxonomy based on execution semantics, a validation-tier label, a comparability rule and an integrated evaluation perspective that together let claims be read at the strength their evidence supports. Table 35 summarises, by research domain, where the evidence stands and what would most change it. To limit overlap with Tables 32–34, it records expected impact ratherthan methods or stakeholders.

Table 35. Key Research Priorities and Future Outlook for Agentic AI Systems

Research Domain	Current Evidence	Remaining Challenge	Future Priority	Expected Impact
Healthcare	Benchmark and prototype tier	No live clinical deployment	Prospective shadow-mode trials	Outcome-level evidence for clinical agents
Finance	Conceptual frameworks	No measured deployment	Process-correctness benchmarks	Auditable delegated action rights

Critical interpretation and limitations. Several limitations bound these conclusions. The executed search used a single scholarly index with 15 queries; Scopus, Web of Science, IEEE Xplore, ACM Digital Library, ScienceDirect and SpringerLink were not searched, and no PRISMA flow is reported. Screening and appraisal were not performed in duplicate, so no inter-rater agreement is available.

REFERENCES

Abbas, M., & Ghous, H. (2022). Early detection of breast cancer tumors using linear discriminant analysis feature selection with different machine learning classification methods. https://doi.org/10.5121/cseij.2022.12117

Abou Ali, M., Dornaika, F., & Charafeddine, J. (2025). Agentic AI: a comprehensive survey of architectures, applications, and future directions. Artificial Intelligence Review, 59(1). https://doi.org/10.1007/s10462-025-11422-4

Ayyagari, V. (2025). Model Context Protocol for Agentic AI: Enabling Contextual Interoperability Across Systems. International Journal of Computational and Experimental Science and Engineering, 11(3). https://doi.org/10.22399/ijcesen.3678

Aziz, A., & Ghous, H. (2021). Fraudulent transactions detection in credit card by using data mining methods: A review. International Journal of Scientific Progress and Research, 79(179), 31–48. Article PDF

Bandi, A., Kongari, B., Naguru, R., Pasnoor, S., & Vilipala, S. V. (2025). The Rise of Agentic AI: A Review of Definitions, Frameworks, Architectures, Applications, Evaluation Metrics, and Challenges. Future Internet, 17(9), Article 404. https://doi.org/10.3390/fi17090404

Barres, V., Dong, H., Ray, S., Si, X., & Narasimhan, K. (2025). τ^2-Bench: Evaluating conversational agents in a dual-control environment [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2506.07982

Bibi, F., Tahreem, D., Azhar, J., Jamshed, S., & Ghous, H. (2025). Cybercrime detection through Min Max Ants System-based feature selection and classification techniques. Southern Journal of Computer Science, 1(1), 81–97. Article page

Bogavelli, T., Sharma, R. C., & Subramani, H. (2025). AgentArch: A Comprehensive Benchmark to Evaluate Agent Architectures in Enterprise [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2509.10769

Botti, V. (2025). Agentic AI and Multiagentic: Are We Reinventing the Wheel? [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2506.01463

Breaux, T. D., & Singhal, A. (2026). The impact of GenAI on the future of requirements engineering. Requirements Engineering, 31(2). https://doi.org/10.1007/s00766-026-00477-6

Cabot, J. (2025). Vibe Modeling: Challenges and Opportunities. In Lecture Notes in Computer Science (pp. 105–118). https://doi.org/10.1007/978-3-032-08620-4_7

Campo, D. N., Conde, J., Alonso, Á., Huecas, G., Salvachúa, J., & Reviriego, P. (2026). Real-time Spatial Retrieval Augmented Generation for Urban Environments. ACM Transactions on Intelligent Systems and Technology. https://doi.org/10.1145/3831675

Chen, S., Lin, S., Shi, Y., Lian, H., Gu, X., Yun, L., Chen, D., Cao, L., Liu, J., Xia, N., & Wang, Q. (2026). SWE-Exp: Experience-Driven Software Issue Resolution [Preprint]. SSRN. https://doi.org/10.2139/ssrn.7234653

Cheng, D., & Tsao, W.-K. (2026). Agent Privilege Separation in OpenClaw: A Structural Defense Against Prompt Injection [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2603.13424

Chhabra, A., Datta, S., Nahin, S. K., & Mohapatra, P. (2026). Agentic AI Security: Threats, Defenses, Evaluation, and Open Challenges. IEEE Access, 14, 49455–49482. https://doi.org/10.1109/access.2026.3675554

Chowa, S. S., Alvi, R., Rahman, S. M. A. U., Rahman, M. A., Raiaan, M. A. K., Islam, M. R., Hussain, M., & Azam, S. (2026). From language to action: a review of large language models as autonomous agents and tool users. Artificial Intelligence Review, 59(2). https://doi.org/10.1007/s10462-025-11471-9

Chugani, H. (2026). Agentic AI Systems: Architecture, Data Infrastructure, and Context Management. International Journal of Computer Trends and Technology, 74(2), 16–29. https://doi.org/10.14445/22312803/ijctt-v74i2p104

Darabi, N., Naik, D., Tayebati, S., Jayasuriya, D., & Trivedi, A. R. (2026). Resilience in Ambient Multi-Agent LLMs via Decentralized Bio-Autonomic Control and Immune-Inspired Anomaly Detection. In Proceedings of the AAAI Conference on Artificial Intelligence (pp. 37332–37340). https://doi.org/10.1609/aaai.v40i44.41065

Deng, Z., Guo, Y., Han, C., Ma, W., Xiong, J., Wen, S., & Xiang, Y. (2025). AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways. ACM Computing Surveys, 57(7), 1–36. https://doi.org/10.1145/3716628

Dennis, L. A., & Fisher, M. (2020). Verifiable Self-Aware Agent-Based Autonomous Systems. Proceedings of the IEEE, 108(7), 1011–1026. https://doi.org/10.1109/jproc.2020.2991262

Devarakonda, S. (2026). Autonomous AI agents in enterprise systems: A review of architectural gaps. Journal of Computer Science and Technology Studies. https://doi.org/10.32996/jcsts.2026.8.10.3

Ding, P. (2025). ToolRegistry: A Protocol-Agnostic Tool Management Library for Function-Calling LLMs [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2507.10593

Du, M., Luo, Y., Banerjee, S., Wojcik, M., Popovic, J., & Cherukara, M. J. (2026). Experiment automation agents: automating materials characterization with vision language model agents. npj Computational Materials. https://doi.org/10.1038/s41524-026-02213-8

Du, S., Zhao, J., Shi, J., Xie, Z., Jiang, X., Bai, Y., & He, L. (2026). A Survey on the Optimization of Large Language Model-based Agents. ACM Computing Surveys, 58(9), 1–37. https://doi.org/10.1145/3789261

Ehtesham, A., Singh, A., Gupta, G. K., & Kumar, S. (2025). A survey of agent interoperability protocols: Model Context Protocol (MCP), Agent Communication Protocol (ACP), Agent-to-Agent Protocol (A2A), and Agent Network Protocol (ANP) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2505.02279

Falco, G., Shneiderman, B., Badger, J., Carrier, R., Dahbura, A., Danks, D., Eling, M., Goodloe, A. E., Gupta, J., Hart, C., Jirotka, M., Johnson, H. H., LaPointe, C., Llorens, A. J., Mackworth, A. K., Maple, C., Pálsson, S. E., Pasquale, F., Winfield, A., & Yeong, Z. K. (2021). Governing AI safety through independent audits. Nature Machine Intelligence, 3(7), 566–571. https://doi.org/10.1038/s42256-021-00370-7

Fang, R., Cai, S., Li, B., Wu, J., Li, G., Yin, W., Wang, X., Wang, X., Su, L., Zhang, Z., Wu, S., Tao, Z., Jiang, Y., Xie, P., Huang, F., & Zhou, J. (2025). Towards General Agentic Intelligence via Environment Scaling [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2509.13311Jiang, Y., Xie, P., Huang, F., & Zhou, J. (2025). Towards General Agentic Intelligence via Environment Scaling [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2509.13311

Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., & Li, Y. (2024). Large language models empowered agent-based modeling and simulation: a survey and perspectives. Humanities and Social Sciences Communications, 11(1). https://doi.org/10.1057/s41599-024-03611-3

Gao, D., Wen, X., Lv, J., Si, J., Wang, C., & Bao, J. (2026). Self-Evolving Digital Twins in Flexible Manufacturing: A Large Language Model-Powered Multi-Agent Approach. Digital Engineering, Article 100099. https://doi.org/10.1016/j.dte.2026.100099

Gautam, A., Aryal, M. R., Deshpande, S., Padalkar, S., Nikolaenko, M., Tang, M., & Anand, S. (2025). IIoT-enabled digital twin for legacy and smart factory machines with LLM integration. Journal of Manufacturing Systems, 80, 511–523. https://doi.org/10.1016/j.jmsy.2025.03.022

Gou, B., Huang, Z., Ning, Y., Gu, Y. J., Lin, M. S., Qi, W., Kopanev, A., Yu, B., Gutiérrez, B. J., Shu, Y.,Song, C. H., Wu, J., Chen, S., Moussa, H. N., Zhang, T., Xie, J., Li, Y., Xue, T., Liao, Z., . . . Su, Y. (2025). Mind2Web 2: Evaluating Agentic Search with Agent-as-a-Judge [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2506.21506

Hadfield, J., Zhang, B., Lien, K., Scholz, F., Fox, J., & Ford, D. (2025, June 13). How we built our multi-agent research system. Anthropic. https://www.anthropic.com/engineering/multi-agent-research-system

Jiang, W., & Hu, F. (2025). Artificial Intelligence Agent-Enabled Predictive Maintenance: Conceptual Proposal and Basic Framework. Computers, 14(8), Article 329. https://doi.org/10.3390/computers14080329

Jiang, Y., Black, K. C., Geng, G., Park, D., Zou, J., Ng, A., & Chen, J. H. (2025). MedAgentBench: A Virtual EHR Environment to Benchmark Medical LLM Agents. NEJM AI, 2(9). https://doi.org/10.1056/aidbp2500144

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2024). SWE-bench: Can language models resolve real-world GitHub issues? In The Twelfth International Conference on Learning Representations (ICLR 2024). https://openreview.net/forum?id=VTF8yNQM66

Joshi, S. (2025). Review of autonomous systems and collaborative AI agent frameworks. International Journal of Science and Research Archive, 14(2), 961–972. https://doi.org/10.30574/ijsra.2025.14.2.0439

Joshi, S. (2025). A Comprehensive Review of Gen AI Agents: Applications and Frameworks in Finance, Investments and Risk Domains. International Journal of Innovative Science and Research Technology, 1339–1355. https://doi.org/10.38124/ijisrt/25may964

Kim, J., Shin, B., Chung, J., & Rhu, M. (2026). The Cost of Dynamic Reasoning: Demystifying AI Agents and Test-Time Scaling from an AI Infrastructure Perspective. In Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA) (pp. 1–16). https://doi.org/10.1109/hpca68181.2026.11408569

Kim, Y., Gu, K., Park, C., Park, C., Schmidgall, S., Heydari, A. A., Yan, Y., Zhang, Z., Zhuang, Y., Liu, Y.,Malhotra, M., Liang, P. P., Park, H. W., Yang, Y., Xu, X., Du, Y., Patel, S., Althoff, T., McDuff, D., & Liu, X. (2026). Capable language models can outgrow the benefits of collaboration. Nature Machine Intelligence, 8(7), 1157–1172. https://doi.org/10.1038/s42256-026-01268-y

Kou, G., Li, Y., Wang, H., & Wang, X. (2026). Human–AI hybrid finance: from AI tools to decision systems. Financial Innovation, 12(1). https://doi.org/10.1186/s40854-026-00941-w

Li, X., Wang, S., Zeng, S., Wu, Y., & Yang, Y. (2024). A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. Vicinagearth, 1(1). https://doi.org/10.1007/s44336-024-00009-2

Liu, B., Li, X., Zhang, J., Wang, J., He, T., Hong, S., Liu, H., Zhang, S., Song, K., Zhu, K., Cheng, Y.,

Wang, S., Wang, X., Luo, Y., Jin, H., Zhang, P., Liu, O., Chen, J., Zhang, H., . . . Wu, C. (2025). Advances and Challenges in Foundation Agents: From Brain-Inspired Intelligence to Evolutionary, Collaborative, and Safe Systems [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2504.01990

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., . . . Tang, J. (2024). AgentBench: Evaluating LLMs as agents. In The Twelfth International Conference on Learning Representations (ICLR 2024). https://openreview.net/forum?id=zAdUB0aCTQ

Liu, Y., Carrero, Z. I., Jiang, X., Ferber, D., Wölflein, G., Zhang, L., Jayabalan, S., Lenz, T., Hui, Z., & Kather, J. N. (2026). Benchmarking large language model-based agent systems for clinical decision tasks. npj Digital Medicine, 9(1). https://doi.org/10.1038/s41746-026-02443-6

Mehandru, N., Miao, B. Y., Almaraz, E. R., Sushil, M., Butte, A. J., & Alaa, A. M. (2024). Evaluating large language models as agents in the clinic. npj Digital Medicine, 7(1), Article 84. https://doi.org/10.1038/s41746-024-01083-y

Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., & Scialom, T. (2023). GAIA: a benchmark for General AI Assistants [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2311.12983

Mishra, V., Saeidi, A., Raj, S., Nakamura, M., Srinivasa, J., Liu, G., Payani, A., & Baral, C. (2025). How Can Input Reformulation Improve Tool Usage Accuracy in a Complex Dynamic Environment? A Study on τ-bench [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2508.20931

Mohammadi, M., Li, Y., Lo, J., & Yip, W. (2025). Evaluation and Benchmarking of LLM Agents: A Survey.In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2025) (pp. 6129–6139). https://doi.org/10.1145/3711896.3736570

Mustafa, A. u., & Aysan, A. F. (2026). Measuring agentic AI adoption and control frameworks in finance.Modern Finance, 4(1). https://doi.org/10.61351/mf.v4i1.557

Peykani, P., Ghanidel, S., Javadi-Sisi, I., Snášel, V., & Mirjalili, S. (2026). A Holistic Review of Agentic AI Frameworks, Applications, and Research Trajectories. Archives of Computational Methods in Engineering. https://doi.org/10.1007/s11831-026-10675-8

Qi, J., Li, M., Liu, J., Shu, Y., Yu, D. S. F., Ma, S., Cui, W., Zhao, Y., Chen, Y., Jiang, R., King, I., & Xu, Z.(2026). Towards trustworthy agentic AI: a comprehensive survey of safety, robustness, privacy, and system security. Academia AI and Applications, 2(2). https://doi.org/10.20935/acadai8260

Quyyam, T., & Ghous, H. (2021). Sentiment analysis of Amazon customer product reviews: A review. International Journal of Scientific Research and Engineering Development, 4(1), 564–595. https://doi.org/10.5281/zenodo.4505616

Rahardja, A. W., Liu, J., Chen, W., Chen, Z., & Lou, Y. (2025). Can Agents Fix Agent Issues? [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2505.20749

Raza, S., Sapkota, R., Karkee, M., & Emmanouilidis, C. (2026). TRiSM for Agentic AI: A review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems. AI Open, 7, 71–95. https://doi.org/10.1016/j.aiopen.2026.02.006

Rehman, I., & Ghous, H. (2021). Structured critical review on market basket analysis using deep learning & association rules. International Journal of Scientific & Engineering Research, 12(1), 1–24. https://doi.org/10.5281/zenodo.5751728

Reid, A., O'Callaghan, S., Carroll, L., & Caetano, T. S. (2025). Risk Analysis Techniques for Governed LLM-based Multi-Agent Systems [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2508.05687

Ren, Y., Liu, Y., Ji, T., & Xu, X. W. (2025). AI Agents and Agentic AI–navigating a plethora of concepts for future manufacturing. Journal of Manufacturing Systems, 83, 126–133. https://doi.org/10.1016/j.jmsy.2025.08.017

Sado, F., Loo, C. K., Liew, W. S., Kerzel, M., & Wermter, S. (2022). Explainable Goal-driven Agents and Robots - A Comprehensive Review. ACM Computing Surveys, 55(10), 1–41. https://doi.org/10.1145/3564240

Sapkota, R., Roumeliotis, K. I., & Karkee, M. (2025). AI Agents vs. Agentic AI: A Conceptual taxonomy, applications and challenges. Information Fusion, 126, Article 103599. https://doi.org/10.1016/j.inffus.2025.103599

Sarkar, S., Hazra, A., Mittal, A., & Sahu, A. (2026). LLM-Based Multi-Agent Collaboration for Constrained Multi-Objective Container Placement. In Proceedings of the IEEE International Conference on Cloud Computing (CLOUD) (pp. 61–67). https://doi.org/10.1109/cloud72782.2026.00018

Schilling-Wilhelmi, M., Ríos-García, M., Shabih, S., Gil, M. V., Miret, S., Koch, C. T., Marquez, J. A., & Jablonka, K. M. (2024). From text to insight: large language models for chemical data extraction. Chemical Society Reviews, 54(3), 1125–1150. https://doi.org/10.1039/d4cs00913d

Schömbs, S., Zhang, Y., Gonçalves, J., & Johal, W. (2025). From Conversation to Orchestration: HCI Challenges and Opportunities in Interactive Multi-Agentic Systems. In Proceedings of the International Conference on Human-Agent Interaction (HAI 2025) (pp. 158–168). https://doi.org/10.1145/3765766.3765795

Shen, Z. (2024). LLM With Tools: A Survey [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2409.18807

Shi, J., Yuan, Z., Tie, G., Zhou, P., Gong, N., & Sun, L. (2026). Prompt Injection Attack to Tool Selection in LLM Agents. In Proceedings of the Network and Distributed System Security Symposium (NDSS 2026). https://doi.org/10.14722/ndss.2026.230675

Su, H., Luo, J., Liu, C., Yang, X. H., Zhang, Y., Dong, Y., & Zhu, J. J. (2026). A Survey on Autonomy-Induced Security Risks in Large Model-Based Agents. IEEE Transactions on Pattern Analysis and Machine Intelligence, 48(10), 11382–11401. https://doi.org/10.1109/tpami.2026.3688650

Wan, Z., Qian, J., Du, Y., Jabbour, J., Du, Y., Zhao, Y. K., Raychowdhury, A., Krishna, T., & Reddi, V. J. (2025). Generative AI in Embodied Systems: System-Level Analysis of Performance, Efficiency and Scalability. In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2025) (pp. 26–37). https://doi.org/10.1109/ispass64960.2025.00013

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J.-R. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18(6). https://doi.org/10.1007/s11704-024-40231-1

Wang, Y., Pan, Y., Su, Z., Deng, Y., Zhao, Q., Du, L., Luan, T. H., Kang, J., & Niyato, D. (2025). Large Model-Based Agents: State-of-the-Art, Cooperation Paradigms, Security and Privacy, and Future Trends. IEEE Communications Surveys & Tutorials, 28, 1906–1949. https://doi.org/10.1109/comst.2025.3576176

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., & Wang, C. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2308.08155

Xie, T., Zhang, D., Cheng, Z., 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., & Yu, T. (2024). OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In Advances in Neural Information Processing Systems 37 (NeurIPS 2024) (pp. 52040–52094). https://doi.org/10.52202/079017-1650

Xie, Z., Wang, H., Dai, L., Wang, Z., Song, H., & Qian, J. (2026). Ethical issues in multi-agent AI systems for healthcare: a narrative review. Frontiers in Public Health, 14, Article 1792627. https://doi.org/10.3389/fpubh.2026.1792627

Xu, Y., Zhuang, Y., Liu, X., Zhang, T. B., Xiao, B., Xu, X., Jiang, D., Wang, J., & Hu, H. (2026). LLM agents security duality: a comprehensive survey of self-security and empowered cybersecurity. Artificial Intelligence Review, 59(8). https://doi.org/10.1007/s10462-026-11563-0

Yang, W., Qiu, H., Zhang, B. Y., Li, C., Huang, Z., Feng, X., Yu, R., & Dong, J. (2026). When OpenClaw Meets Hospital: Toward an Agentic Operating System for Dynamic Clinical Workflows [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2603.11721

Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2406.12045

Yehudai, A., Eden, L., Li, A., Uziel, G., Zhao, Y., Bar-Haim, R., Cohan, A., & Shmueli-Scheuer, M. (2025). Survey on Evaluation of LLM-based Agents [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2503.16416

Zhan, Q., Liang, Z., Ying, Z., & Kang, D. (2024). InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. In Findings of the Association for Computational Linguistics: ACL 2024 (pp. 10471–10506). https://doi.org/10.18653/v1/2024.findings-acl.624

Zhan, Q., Fang, R., Panchal, H. S., & Kang, D. D. (2025). Adaptive Attacks Break Defenses Against Indirect Prompt Injection Attacks on LLM Agents. In Findings of the Association for Computational Linguistics: NAACL 2025 (pp. 7101–7117). https://doi.org/10.18653/v1/2025.findings-naacl.395

Zhang, K., Yao, W., Liu, Z., Feng, Y., Liu, Z., Murthy, R., Lan, T., Li, L., Lou, R., Xu, J., Pang, B., Zhou, Y., Heinecke, S., Savarese, S., Wang, H., & Xiong, C. (2024). Diversity Empowers Intelligence: Integrating Expertise of Software Engineering Agents [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2408.07060

Zhang, L., He, S., Zhang, C., Kang, Y., Li, B., Xie, C., Wang, J., Wang, M., Huang, Y., Fu, S., Nallipogu, E., Lin, Q., Dang, Y., Rajmohan, S., & Zhang, D. (2025). SWE-bench Goes Live! [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2505.23419

Zhang, Y., Ruan, H., Fan, Z., & Roychoudhury, A. (2024). AutoCodeRover: Autonomous Program Improvement. In Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024) (pp. 1592–1604). https://doi.org/10.1145/3650212.3680384

Zhang, Z., Dai, Q., Bo, X., Ma, C., Li, R. L., Chen, X., Zhu, J., Dong, Z., & Wen, J.-R. (2025). A Survey on the Memory Mechanism of Large Language Model-based Agents. ACM Transactions on Information Systems, 43(6), 1–47. https://doi.org/10.1145/3748302

Zhang, Z., Yao, Y., Zhang, A., Tang, X., Ma, X., He, Z., Wang, Y., Gerstein, M., Wang, R., Liu, G., & Zhao, H. (2025). Igniting Language Intelligence: The Hitchhiker's Guide from Chain-of-Thought Reasoning to Language Agents. ACM Computing Surveys, 57(8), 1–39. https://doi.org/10.1145/3719341

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Chen, Y., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., . . . Wen, J.-R. (2026). A Survey of Large Language Models. Frontiers of Computer Science, 20(12). https://doi.org/10.1007/s11704-026-60308-3

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., & Neubig, G. (2024). WebArena: A realistic web environment for building autonomous agents. In The Twelfth International Conference on Learning Representations (ICLR 2024). https://openreview.net/forum?id=oKn9c6ytLx

Zhuge, M., Zhao, C., Ashley, D., Wang, W., Khizbullin, D., Xiong, Y., Liu, Z., Chang, E., Krishnamoorthi, R., Tian, Y., Shi, Y., Chandra, V., & Schmidhuber, J. (2024). Agent-as-a-Judge: Evaluate Agents with Agents [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2410.10934