

Agentic AI for Complex Decision-Making: A Systematic Critical Review of Planning, Memory, Tools, Safety and Assurance in Autonomous Systems

Mohan Babu

Vellore Institute of Technology

Faranak Nejatia

Universiti Tunku Abdul Rahman (UTAR)

Madhan E S

`madhan.es@vit.ac.in`

Vellore Institute of Technology

Research Article

Keywords: Agentic AI, Large language model agents, Autonomous decision-making, Planning and reasoning, Multi-agent systems, AI safety and assurance

Posted Date: October 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11226881/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Agentic AI for Complex Decision-Making: A Systematic Critical Review of Planning, Memory, Tools, Safety and Assurance in Autonomous Systems

Mohan Babu A¹ Faranak Nejatia² Madhan E S³

^{1,3}School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India.

Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman (UTAR), Sungai Long Campus, Kajang, 43200, Selangor, Malaysia.

¹Mohan Babu A : mohanbabu.a2022@vitstudent.ac.in.

²Faranak Nejatia : Faranak@utar.edu.my

³Corresponding author: Madhan E S : madhan.es@vit.ac.in

Abstract

Large language models (LLMs) are increasingly deployed as agents that plan, remember, call tools, coordinate with other agents and act over long horizons with limited supervision. Evidence on whether such agents work reliably is scattered and sometimes contradictory, and recent reviews largely describe architectures, benchmarks or threats in isolation. This systematic, taxonomy-driven critical review follows PRISMA 2020 reporting where applicable, without meta-analytic pooling, and synthesises 98 verified sources published up to 30 June 2026. It differs from prior reviews in four ways. First, it weighs evidence: eight headline claims are graded by an explicit rule, and contradictions are traced to moderators such as feedback source, equal-compute comparison and task verifiability. Second, it specifies a closed-loop reference framework operationally, adding a verification stage and a safety-and-authorisation gate, and codes ten published systems against it; none fully implements both control stages. Third, it links capabilities to evaluation hooks, readiness levels and re-assurance triggers for agents that change after deployment. Fourth, its evidence base is auditable through re-executed searches with measured recall, verified references and a released evidence matrix. High-confidence findings are that intrinsic self-correction is unreliable, long-horizon autonomous planning remains weak and tool-using agents are broadly vulnerable to injection. Multi-agent benefits are conditional (moderate confidence), whereas claims about benchmark validity and memory consistency rest mainly on preprints. A reliability model quantifies the value of independent verification, and an eighteen-direction roadmap is ranked by an explicit priority rule.

Keywords Agentic AI · Large language model agents · Autonomous decision-making · Planning and reasoning · Multi-agent systems · AI safety and assurance

1 Introduction

1.1 From predictive AI to generative AI to agentic AI

Research on AI for decision-making can be read as a sequence of three overlapping phases. In the first, predictive phase, learned controllers reached strong performance in narrow and precisely specified settings. Deep reinforcement learning matched human players on a suite of Atari games (Mnih et al. 2015), and pairing deep networks with tree search produced a program that defeated professional Go players (Silver et al. 2016). Such systems were impressive within their task but carried little over to anything else. The second, generative phase began with the Transformer (Vaswani et al. 2017) and with the finding that sufficiently large language models can learn new tasks from a handful of in-context examples (Brown et al. 2020). Instruction tuning with human feedback then made these models usable for open-ended requests (Ouyang et al. 2022). Even so, a generative model remains reactive: it maps one context to one output and does not pursue a goal through a sequence of interactions.

The third, agentic phase changes the role of the model. The LLM becomes the reasoning component inside a larger system that observes, decides and acts repeatedly. A turning point was the demonstration that interleaving natural-language reasoning with task actions lets a model plan, keep track of progress and adjust to what the environment returns (Yao et al. 2023b). A substantial literature has since formed around cognitive architectures for language agents (Sumers et al. 2024), component-based designs for autonomous agents (Wang L et al. 2024) and the idea of agent societies (Xi et al. 2025). The present review concentrates on this phase and, within it, on complex decisions that unfold over many steps.

1.2 How agentic systems differ from conventional LLM applications

A chatbot or a retrieval-based question-answering interface behaves, for practical purposes, like a stateless function: a query goes in and an answer comes out. Agentic systems depart from this pattern in four ways. They act, issuing calls that change external state through application programming interfaces (APIs) (Schick et al. 2023; Qin et al. 2024), executing code (Wang X et al. 2024) or operating a browser or desktop (Zhou et al. 2024; Xie et al. 2024). They persist, carrying memory across many steps and sessions (Park et al. 2023; Packer et al. 2024). They observe the effects of their actions and change course, often through explicit self-reflection (Shinn et al. 2023; Madaan et al. 2023). And they can split work among several cooperating agents (Wu et al. 2024; Hong et al. 2024). Sapkota et al. (2026) argue that treating these properties as ordinary LLM usage creates conceptual confusion, and they separate bounded "AI agents" from orchestrated "agentic AI". Section 3 builds on and refines that distinction.

1.3 Why long-horizon decision-making is the test case

The strongest case for agentic AI lies in work that consists of many dependent steps: fixing real software defects (Jimenez et al. 2024; Yang J et al. 2024), completing web and desktop workflows (Zhou et al. 2024; Xie et al. 2024) and supporting science, from automated chemical experimentation (Boiko et al. 2023) to new mathematical constructions found by program search (Romera-Paredes et al. 2024). Yet long horizons are exactly where current systems are weakest. On realistic web tasks, the best GPT-4-based agent reported with WebArena completed 14.41% of tasks end to end (Zhou et al. 2024). On classical planning problems, only about 12% of plans produced by the strongest model tested were both executable and goal-reaching (Kambhampati et al. 2024). Open-ended computer-use benchmarks show success falling as tasks lengthen (Xie et al. 2024). Measuring the length of task that an agent can complete reliably has itself become a research programme: Kwa et al. (2025) report that the 50% task-completion time horizon of frontier models has doubled roughly every seven months since 2019, which signals rapid progress but also underlines how far typical horizons remain from the multi-day processes found in practice. The distance between what agents promise and what they reliably deliver is the reason this review takes a critical rather than a purely descriptive stance.

1.4 Position relative to existing surveys and novelty

Surveys of LLM agents are already numerous (Table 1). Broad reviews organise agents into component frameworks (Wang L et al. 2024) or into perception, reasoning and action (Xi et al. 2025). Focused surveys address planning (Huang X et al. 2024), tool learning (Qu et al. 2025), memory (Zhang et al. 2024), multi-agent systems (Guo et al. 2024) and agent security and privacy (He et al. 2025). Conceptual treatments distinguish AI agents from agentic AI (Sapkota et al. 2026), organise agentic LLMs around reasoning, acting and interacting (Plaat et al. 2025) and contrast agentic AI with rule-based systems (Acharya et al. 2025).

This journal alone has published six closely related reviews in the past year, so differentiation from them matters most:

- Abou Ali et al. (2025) synthesised 90 studies through a PRISMA-guided process, identified a governance deficit and called for hybrid neuro-symbolic designs.
- Chowa et al. (2026) reviewed LLMs as autonomous agents and tool users. They drew on papers from 2023–2025 in highly ranked venues and analysed 68 public datasets.
- Alva and Pandey (2026) proposed a modular agentic architecture oriented towards cloud-scale deployment.

- Kehkashan et al. (2026) examined agent evaluation. They analysed 15 benchmarks and argued that benchmark scores and deployment viability are disconnected.
- Xu et al. (2026) surveyed LLM-agent security, covering both threats to agents and the use of agents for cybersecurity.
- Khadka and Maple (2026) showed that AI assurance frameworks name the right goals but rarely specify how those goals become auditable once a system changes after deployment. This review was published after our 30 June 2026 cut-off and is cited for context only (Sect. 2.1).

Against this background, the present review makes four claims of novelty. Each is tied to a concrete artefact, so that a reader can check it.

N1. Evidence weighting instead of description. Each headline conclusion is graded with an explicit rule by the strength and consistency of its support, and set against the evidence that qualifies or contradicts it. The result is a claim–evidence map (Sect. 11) that explains apparent contradictions through identified moderators. The reviews above describe architectures, datasets, benchmarks or threats within their own scope. None of them grades cross-capability claims in this way.

N2. An operational reference framework that is tested, not only proposed. The closed-loop framework is defined formally, stage by stage, with measurable pass criteria. It is then applied, using explicit coding rules, to ten published systems (Sects. 3.7 and 3.8). The architectures proposed by Abou Ali et al. (2025) and Alva and Pandey (2026) are not evaluated against existing systems in this way.

N3. Integration of capability, evaluation and assurance. Evaluation validity (Kehkashan et al. 2026) and security (Xu et al. 2026) have each been reviewed separately. Here they are connected to each capability through evaluation hooks (Sect. 9.2), readiness criteria (Sect. 10.1) and agent-specific re-assurance triggers (Sect. 8.5). The triggers extend the lifecycle perspective of Khadka and Maple (2026) to systems whose behaviour changes through memory, tools and policies.

N4. A reproducible and audited evidence base with measured differentiation. The search strings were re-executed and their recall against the corpus was measured (Sect. 2.3). Every reference was checked against arXiv or Crossref. The coded matrix of all 98 sources is released (Online Resource 2). Section 2 reports the limits of the search without hiding them. Differentiation from the recent reviews in this journal is also measured, not only asserted, as described next.

Measured overlap with recent reviews in this journal. To test differentiation objectively, we retrieved the reference lists that five of the six reviews listed above have deposited with Crossref (912 references in total). The sixth, Khadka and Maple (2026), had not yet deposited its list at the time of writing. We matched these lists against our corpus by DOI, arXiv identifier and exact normalised title (Fig. 1).

The overlap is small:

- The reviews cite between 2 and 30 of our 98 sources each.
- 44 of our 98 sources (45%) are cited by none of them.
- Of the 33 sources that carry the graded claims of Sect. 11, 15 (45%) are cited by none of the five reviews. Among them are the negative and contradictory results on self-correction, planning and benchmark validity.

We also ran the check in the reverse direction. The five reviews share only 31 works that at least two of them cite. Of these, 21 are in our corpus, including all three works cited by three or more reviews. Four of them were added as a result of this check (Ma et al. 2024; Yang W et al. 2024; Zhuge et al. 2024; Huang et al. 2025). The remaining ten were excluded for two reasons. Five are task datasets or evaluations of base models rather than studies of agents. The other five would not change the grade of any claim.

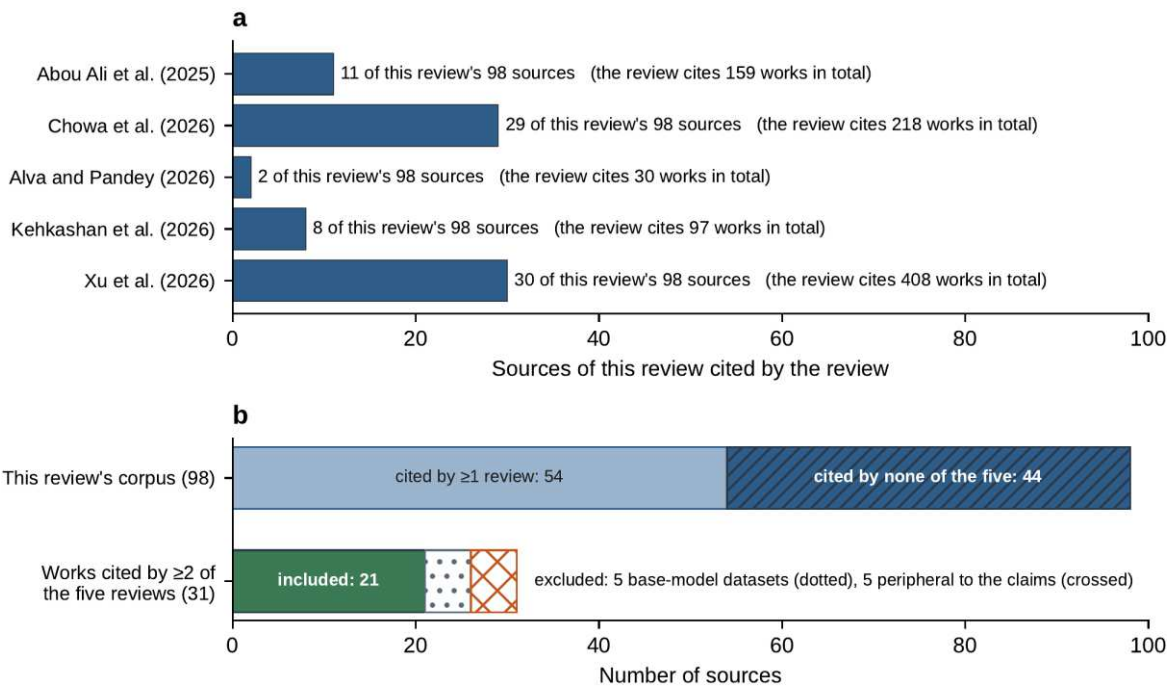


Fig. 1 Measured overlap between this review and five recent reviews in Artificial Intelligence Review, based on the reference lists deposited with Crossref (912 references). a Number of this review's sources cited by each review. b Top: sources in this review's corpus cited by at least one of the reviews versus by none. Bottom: the 31 works cited

by at least two of the reviews, split into those included in this review and those excluded, with the reason for exclusion

Table 1 Positioning of this review relative to the closest prior reviews, including the six most recent related reviews in Artificial Intelligence Review (AIRE). Scope and evidence base are taken from each review's stated aims and abstract; the last column states what the present review adds. Asterisks mark preprints

Review (venue)	Primary scope	Stated evidence base	What this review adds
Abou Ali et al. 2025 (AIRE)	Architectures, applications and future directions of agentic AI	90 studies, PRISMA-guided	Graded claim–evidence map; framework tested against ten systems (N1, N2)
Chowa et al. 2026 (AIRE)	LLMs as autonomous agents and tool users; reasoning, planning, memory	2023–2025 papers in A*/A venues and Q1 journals; 68 datasets	Contradictory evidence reconciled through moderators; readiness criteria (N1, N3)
Alva and Pandey 2026 (AIRE)	Modular agentic architecture for cloud-scale generative systems	Literature synthesis and framework proposal	Operationally defined framework with safety gate, coded against published systems (N2)
Kekhshian et al. 2026 (AIRE)	Evaluation of agentic AI; benchmark–deployment disconnect	15 benchmarks; software development case study	Capability-by-capability evaluation hooks linked to claims and readiness (N3)
Xu et al. 2026 (AIRE)	Security of LLM agents and agents for cybersecurity	Survey of threats and defences	Security threats linked to action gating and re-assurance triggers (N3)
Khadka and Maple 2026 (AIRE)	Operational commitments for AI assurance (not agent-specific)	118-source structured review	Agent-specific re-assurance triggers for memory, tool, model and policy change (N3)
Wang L et al. 2024 (Front Comput Sci)	Construction, application and evaluation of LLM-based autonomous agents	Broad survey	Evidence-weighted assessment of each component (N1)
Xi et al. 2025 (Sci China Inf Sci)	Brain–perception–action framing; agent societies	Broad survey	Verification and gate stages made explicit and testable (N2)
Huang X et al. 2024* (arXiv)	Planning of LLM agents	Focused survey	Planning compared on cost, horizon and safety; claim C1 graded (N1)
Qu et al. 2025 (Front Comput Sci)	Tool learning with LLMs	Focused survey	Tool use treated as a pipeline with permission, validation and rollback gaps
Zhang et al. 2024* (arXiv)	Memory mechanisms of LLM-based agents	Focused survey	Memory consistency and poisoning linked to re-assurance triggers; claim C6 graded
Guo et al. 2024 (IJCAI)	LLM-based multi-agent systems	Focused survey	Equal-compute comparison and failure modes; claim C3 graded
He et al. 2025 (ACM Comput Surv)	Security and privacy of LLM agents	Survey with case studies	Threat-to-residual-risk chain connected to lifecycle assurance
Sapkota et al. 2026 (Inf Fusion)	AI agents versus agentic AI	Conceptual taxonomy	Criteria-based four-level autonomy scale that places both on one axis
Plaat et al. 2025 (J Artif Intell Res)	Agentic LLMs: reasoning, acting, interacting	Survey	Retrospective instantiation of a reference framework
Acharya et al. 2025 (IEEE Access)	Agentic AI versus traditional and rule-based AI	Survey	Criteria-based readiness levels by application domain

1.5 Research questions

Seven research questions (RQs) structure the review.

RQ1 (definitions and architecture). How do the terms AI agent, autonomous agent, agentic AI and multi-agent system differ, and can a single architectural taxonomy accommodate all of them?

RQ2 (planning and reasoning). How far can LLM-based agents plan and verify their own plans for long-horizon decisions, and how do prompting strategies compare with approaches that place a verifier in the loop once computation and validity are controlled?

RQ3 (memory). Which memory architectures do agents use, how do they behave as horizons grow, and what trade-offs arise among recall, cost and reliability?

RQ4 (tool use). How do agents learn and orchestrate external tools, and what determines reliability, generalisation and the safety of an expanding action surface?

RQ5 (coordination). Under what conditions do multi-agent systems outperform single agents at equal computation, and what are their dominant failure modes?

RQ6 (reflection and self-correction). When does self-correction help, when does it fail, and what role does external, verifiable feedback play?

RQ7 (safety, evaluation and deployment). Which safety and control mechanisms exist, how valid and reproducible are current benchmarks, and what follows for dependable deployment, including assurance after deployment?

1.6 Contributions

The review makes seven contributions:

1. A reconciled set of definitions that places conventional LLM applications, tool-augmented agents, autonomous agents and multi-agent systems on a four-level autonomy spectrum. The spectrum is defined by explicit, observable criteria (Sect. 3.1–3.3).
2. A unified taxonomy with seven dimensions. The taxonomy states the different roles of safety and evaluation within it (Sect. 3.6).
3. A formally specified closed-loop reference framework with operational definitions for each stage, and a rule-based retrospective mapping of ten representative systems onto it (Sect. 3.7–3.8).

4. A critical comparison of methods for each capability (Sect. 4–7). A cross-cutting synthesis follows, grading eight headline claims by the strength of their evidence and explaining the principal contradictions through moderators (Sect. 11).
5. Evaluation hooks for each capability, criteria-based readiness levels for application domains, and agent-specific re-assurance triggers for systems that change after deployment (Sects. 8.5, 9 and 10).
6. An eighteen-item research roadmap in which every direction is linked to graded evidence and ranked by an explicit priority rule (Sect. 12).
7. A reproducible evidence base: a review protocol, an audit of search recall, verified bibliographic metadata and a coded evidence matrix covering all 98 sources (Sect. 2; Online Resources 1 and 2).

1.7 Organisation

Section 2 describes the review method and the composition of the corpus. Section 3 sets out conceptual foundations, the taxonomy, the reference framework and its retrospective instantiation. Sections 4 to 7 analyse planning and reflection, memory, tool use and coordination. Section 8 covers safety, governance and assurance, Sect. 9 evaluation and Sect. 10 applications. Section 11 draws the capability analyses together into a graded claim–evidence map. Section 12 develops the prioritised roadmap, and Sect. 13 concludes.

2 Review methodology

2.1 Review design and methodological identity

This article is a systematic, taxonomy-driven critical review. It is systematic in its identification and reporting. It uses predefined search strings, stated eligibility criteria and two-stage screening. It grades evidence by rule, codes every source in a released matrix, and audits the recall of its searches. It follows the PRISMA 2020 reporting guideline (Page et al. 2021) wherever the guideline's items apply to a narrative, qualitative synthesis. It is critical in its synthesis: rather than pooling effect sizes, it compares, contests and reorganises a heterogeneous literature of methods, benchmarks, position papers and governance instruments. No meta-analytic statistical pooling is performed, because the reported outcomes are too heterogeneous in task, metric and model generation to be combined (Sect. 2.8).

The PRISMA 2020 items that do not apply, or were not met, are reported openly rather than left implicit:

- The review protocol was not registered in advance.
- Screening and coding were not performed independently in duplicate. All decisions are recorded in the evidence matrix.
- Record-level counts from the original database searches were not retained, because identification proceeded iteratively. In their place, the searches were re-executed and audited on a fixed date (Sect. 2.3), and the resulting flow is reported in Fig. 2.
- Risk-of-bias assessment in the clinical sense is replaced by the rule-based evidence grading of Sect. 2.5.

Online Resource 1 maps each PRISMA 2020 item to the place in this article where it is reported, or states the deviation. The coded matrix is provided as Online Resource 2.

Review window. The corpus covers sources published from January 2021 to 30 June 2026, together with earlier anchor works. Sources that appeared after the cut-off are not part of the corpus. They are not counted in any corpus statistic or used to grade any claim. One such source, Khadka and Maple (2026), published in September 2026, is cited only as context for the lifecycle-assurance perspective in Sects. 1.4 and 8.5. The search audit and the coverage check were procedures carried out on 30 September 2026. They verified and tested the corpus; they did not extend the window.

2.2 Sources, search period and search strings

Searches covered the venues in which agentic-AI research appears: arXiv (cs.AI, cs.CL, cs.LG and cs.MA), DBLP, Semantic Scholar, the ACL Anthology, OpenReview (ICLR, NeurIPS, ICML and COLM), IEEE Xplore, the ACM Digital Library and SpringerLink. Google Scholar was used as a recall check. The primary window runs from January 2021 to June 2026. Earlier work was admitted only as a marked anchor for a definition or a contrast.

Each capability query combined an agent block ("agentic AI", "LLM agent(s)", "language agent(s)", "autonomous agent(s)", "LLM-based agent(s)") with a capability block. The capability blocks were as follows:

- planning: "planning", "chain-of-thought", "ReAct", "tree of thoughts", "PDDL" and "reasoning model";
- memory: "long-term memory", "episodic memory", "agent memory", "retrieval-augmented" and "memory consolidation";
- tool use: "tool learning", "tool use", "function calling", "code execution", "API" and "Model Context Protocol";

- coordination: "multi-agent", "agent collaboration", "debate", "orchestration" and "role-playing";
- reflection: "self-correction", "self-refine", "reflection", "verification" and "critic";
- safety and governance: "prompt injection", "jailbreak", "guardrail", "oversight", "sandbox", "assurance" and "governance";
- evaluation: "benchmark", "success rate", "contamination", "reproducibility" and "evaluation".

Backward and forward snowballing from key works and from the closest competing surveys complemented the database queries. Online Resource 1 gives the exact strings.

2.3 Search audit: volume and recall

To make identification reproducible, and to expose its limits honestly, we re-executed the capability queries on 30 September 2026 against OpenAlex, a multidisciplinary index that covers arXiv, conference proceedings and journals. The queries were restricted to titles and abstracts and to the review window. We then measured how many corpus sources the same strings retrieve. For this we used the titles and abstracts of the 78 in-window corpus sources that have arXiv records. The results are given in Table 2.

Two findings matter. First, the queries return tens of thousands of records (22,835 for the union of capability blocks). Screening against the eligibility criteria is therefore unavoidable, and a small corpus can only be a purposive, critically selected sample, not a census.

Second, the strict agent-term strings retrieve only 22 of the 78 testable in-window sources (28%). Many foundational agent papers, such as ReAct (Yao et al. 2023b), the self-correction studies of Madaan et al. (2023) and Huang J et al. (2024), and SWE-bench (Jimenez et al. 2024), describe themselves as work on "large language models" rather than "agents". When the agent block is broadened to include "agent(s)", "LLM(s)" and "(large) language model", and the capability vocabulary is broadened accordingly, recall rises to 75 of 78 (96%). The three sources still not retrieved (Bai et al. 2022; Hou et al. 2025; Kwa et al. 2025) entered the corpus through snowballing.

The identification route of every source is recorded in Online Resource 2, and Fig. 2 summarises the whole identification process. For replication we recommend the broadened strings. This audit was carried out after the corpus was assembled. It characterises the protocol; it was not used to select sources.

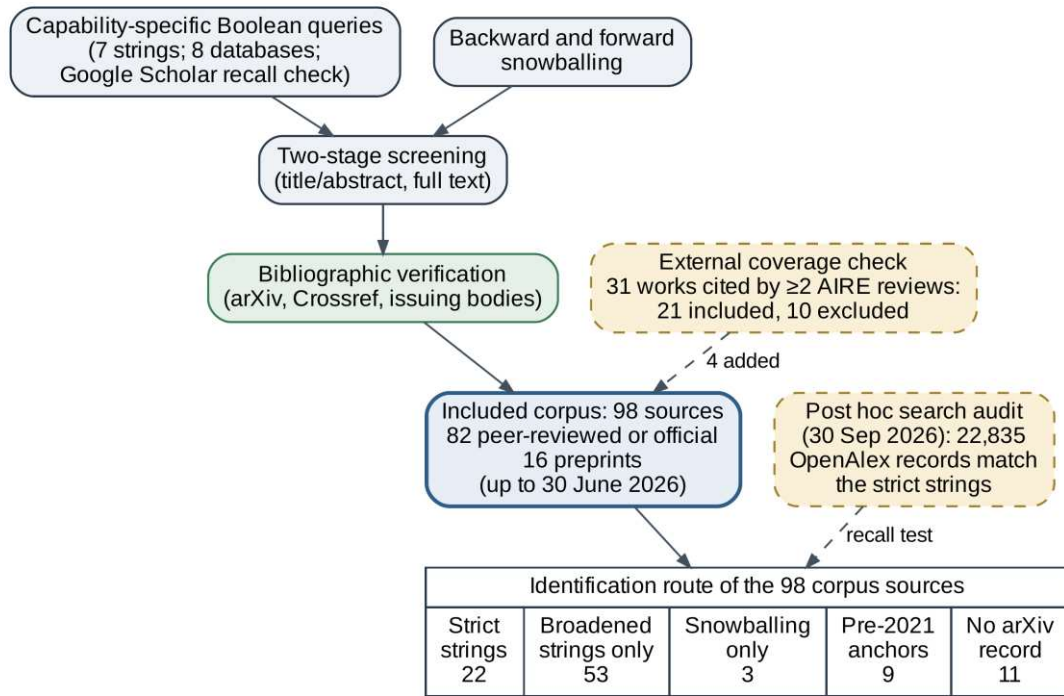


Fig. 2 PRISMA 2020-informed flow of identification, screening and verification of the corpus (sources up to 30 June 2026), with the post hoc search audit and the external coverage check. Record-level counts of the original database searches were not retained (Sect. 2.1); the audit counts are reported instead. The record at the bottom gives the identification route of each of the 98 corpus sources. Dashed boxes mark checks carried out after the corpus was assembled

Table 2 Search audit executed on 30 September 2026. Hits are OpenAlex title-and-abstract matches for the agent block combined with each capability block, restricted to 1 January 2021–30 June 2026. Recall is measured on the in-window corpus sources that have arXiv records, whose titles and abstracts were tested against the same strings

Query	Hits (OpenAlex)	Corpus sources retrieved
Agent block only	30,635	–
Agent block AND planning	5,266	6
Agent block AND memory	1,803	2
Agent block AND tool use	3,182	5
Agent block AND coordination	8,860	6
Agent block AND reflection	4,202	4
Agent block AND safety and governance	6,125	4
Agent block AND evaluation	13,085	15
Union of capability queries (strict strings)	22,835	22 of 78 (28%)
Broadened strings (recommended for replication)	not computed	75 of 78 (96%)
Sources entering only through snowballing	–	56 of 78 under strict strings; 3 under broadened strings

2.4 Eligibility criteria

A source was included when all six of the following held:

- it concerned LLM or foundation-model agents, or was a load-bearing precursor;

- it addressed at least one of the seven capabilities;
- it was a method, benchmark, survey or position paper, empirical study, or governance instrument;
- it fell within the window or was a marked anchor;
- it had a stable identifier;
- it was written in English.

A source was excluded in any of the following cases:

- it had no agentic component, unless it anchored a definition;
- it reported an application with no transferable methodological or safety insight;
- it duplicated a contribution already captured, in which case the version of record was kept;
- it was non-archival grey literature, unless it was the only record of a widely used system or an official governance instrument;
- it could not be verified.

2.5 Screening, evidence grading and coding

Screening ran in two stages, first on titles and abstracts and then on full texts, and was followed by snowballing. Each source was assigned an evidence grade by a fixed rule, so that the weight given to a source in the synthesis is transparent:

- Grade A: peer-reviewed empirical work, benchmarks and methods that report experiments.
- Grade B: peer-reviewed surveys, position papers, official standards and governance instruments.
- Grade C: preprints.

Foundational anchors are graded as A or B by the same rule.

Coding combined deductive and inductive codes. The deductive codes were the seven capabilities plus a definition-and-taxonomy code. The inductive codes were negative result, contradiction, replication or contamination concern, definitional disagreement and open gap. The evidence matrix records, for each source, its identifier, contribution type, publication status, evidence grade, identification route, the sections that cite it, its capability codes and its critical codes (Online Resource 2).

The grading rules for the headline claims (Sect. 11) and for the readiness levels (Sect. 10) are stated where they are used, so that every judgement in the synthesis can be traced to a rule and to identifiable sources.

2.6 Evidence policy and reference verification

Peer-reviewed work is preferred and is required to anchor each headline contradiction. Preprints are admitted because much agent research appears first on arXiv, but they are marked with an asterisk in every table. They cannot, on their own, raise a claim above low confidence (Sect. 11).

Every reference was verified before submission, and the results are reported for each record in Online Resource 2:

- 82 records were compared against arXiv, covering title, complete author list and date.
- 33 publisher DOIs, venues, volumes and pages were checked against Crossref or the proceedings publisher.
- Two governance documents were checked against the issuing body.
- Three standard books and proceedings (Rao and Georgeff 1995; Sutton and Barto 2018; Russell and Norvig 2021) were checked manually against their bibliographic records.

Where an arXiv paper had since appeared at a peer-reviewed venue, the version of record is cited. The verification changed 11 bibliographic records relative to the first draft of this manuscript. The changes corrected first authors, author lists or author order, updated titles, and updated venues for papers that have since been published. The check also corrected one quoted performance figure: the first draft had stated WebArena success as 10.6%, whereas the version of record reports 14.41%.

2.7 Composition of the corpus

The final coded corpus contains 98 sources (Fig. 3). The reference list has 100 entries: the 98 corpus sources, one methodological reference that is not part of the evidence base (Page et al. 2021), and one post-cut-off context source (Khadka and Maple 2026; Sect. 2.1). The evidence matrix in Online Resource 2 lists all 98 corpus sources, with the context source flagged separately. Of these, 82 (84%) are peer-reviewed publications or official instruments, and 16 (16%) are preprints. Nine sources predate 2021 and serve as anchors. By contribution type, the corpus contains 36 methods or systems, 18 surveys and position papers, 17 benchmarks and evaluation studies, 12 empirical analyses of agent behaviour and failure, 10 foundational anchors and 5 governance or standards documents.

Coverage across the capability sections is uneven (Fig. 3c), and the unevenness is itself a finding. Safety and governance (30 sources), planning and reflection (28) and tool use (28) are well represented. Memory (16) and multi-agent coordination (14) draw on markedly fewer sources, which matches the gaps identified in Sects. 5.5 and 7.4.

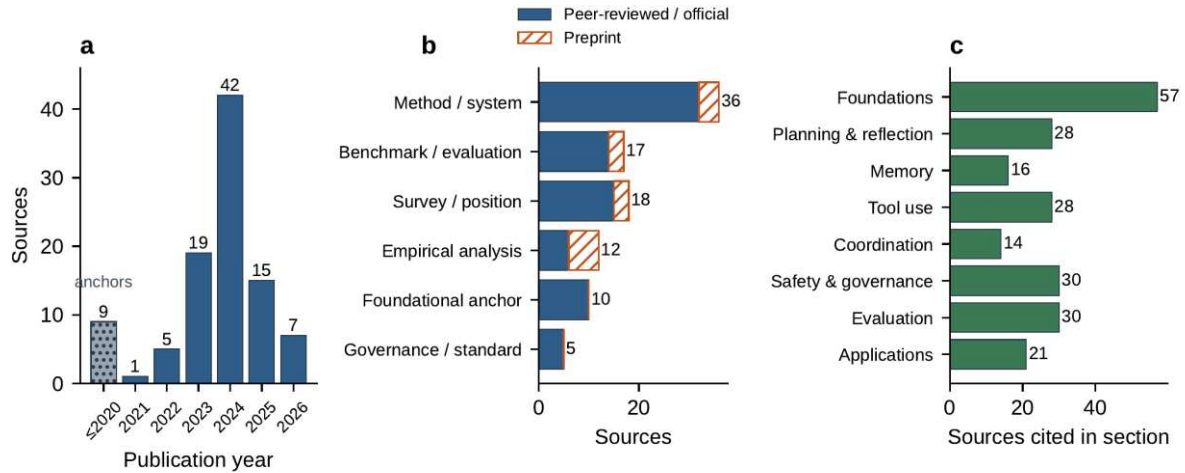


Fig. 3 Composition of the coded corpus (n = 98). a Publication year; the dotted bar aggregates pre-2021 anchor sources. b Contribution type, split into peer-reviewed or official sources (solid) and preprints (hatched). c Number of sources cited in each capability section. Counts are computed from the evidence matrix in Online Resource 2

2.8 Limitations of the review

The review has the following limitations:

- Screening and coding were not performed independently in duplicate, so selection involves reviewer judgement.
- The strict search strings have limited recall, as reported in Sect. 2.3.
- The differentiation analysis relies on reference lists deposited with Crossref, which were available for five of the six recent reviews in this journal.
- Conclusions reflect the literature up to the June 2026 cut-off.
- Coverage is centred on English-language computer-science venues.
- Several critical signals come from preprints.
- Reported performance figures are heterogeneous and, for some benchmarks, affected by contamination, so no numerical pooling is attempted.

2.9 Use of AI-based tools

In line with the journal's policy on large language models, we report that an LLM-based assistant (Claude, Anthropic) was used during manuscript preparation, for the following purposes:

- language editing and restructuring to the journal's format;

- drafting revised and additional text from outlines and source material supplied by the authors, including the framework formalisation (Sect. 3.7), the system mapping and coverage indices (Sect. 3.8), the reliability model (Sect. 3.9), the re-assurance triggers (Sect. 8.5), the readiness criteria (Sect. 10), the claim–evidence map (Sect. 11) and the priority scoring of the roadmap (Sect. 12);
- executing the search audit, the reference-overlap analysis and the bibliographic verification against OpenAlex, arXiv and Crossref, using the strings and rules specified by the authors;
- suggesting a small number of additional recent sources;
- redrawing figures from diagrams and data specified by the authors.

The authors checked every factual claim against the cited sources, decided which sources to include and how to grade and interpret them, and approved the final text. They take full responsibility for the content of the article.

3 Conceptual foundations, taxonomy and reference framework

3.1 Defining agentic AI

Classical agent theory describes an agent as a computational entity situated in an environment and characterised by autonomy, reactivity, pro-activeness and social ability (Wooldridge and Jennings 1995). Textbook treatments add that a rational agent chooses actions expected to maximise its performance measure given its percept history (Russell and Norvig 2021). Present-day agentic AI inherits this vocabulary but replaces hand-engineered control with a policy implemented by a foundation model. Drawing on cognitive-architecture accounts (Sumers et al. 2024) and component-based surveys (Wang L et al. 2024), we use the following working definition:

Agentic AI refers to systems in which one or more foundation-model policies pursue goals over multiple steps by interacting with an environment through a perceive–decide–act loop, using persistent memory and external tools or actions, taking feedback into account, and operating with more autonomy than single-turn response generation requires.

Three properties set such systems apart from ordinary LLM use: goals that extend across time, state that persists from one step to the next, and the ability to take external actions with real consequences. Each property comes in degrees. We therefore treat autonomy as a spectrum (Sect. 3.3), and we read the debate between bounded "AI agents" and orchestrated "agentic AI" (Sapkota et al. 2026) as a question of where a system sits on that spectrum rather than as a categorical divide.

3.2 A spectrum of agent types

The familiar agent types are best treated as regions of one design space rather than as separate species. Reactive agents map percepts to actions with little deliberation (Wooldridge and Jennings 1995). Deliberative agents construct explicit plans; examples range from the belief–desire–intention (BDI) model (Rao and Georgeff 1995) to reasoning-and-acting loops (Yao et al. 2023b), deliberate search over intermediate thoughts (Yao et al. 2023a) and delegation to sound external planners (Liu et al. 2023). Autonomous agents sustain goal-directed behaviour over many steps (Wang L et al. 2024; Park et al. 2023). Tool-augmented agents enlarge their action space by calling external resources (Schick et al. 2023; Qin et al. 2024; Patil et al. 2024; Wang X et al. 2024). Self-improving agents revise their behaviour from feedback (Shinn et al. 2023; Madaan et al. 2023) and accumulate reusable skills (Wang G et al. 2024). Embodied agents act in grounded or simulated worlds (Wang G et al. 2024; Yao et al. 2022). Multi-agent systems divide a task among communicating agents (Wu et al. 2024; Li et al. 2023; Hong et al. 2024; Qian et al. 2024; Du et al. 2024). Because a single system usually combines several of these traits, the taxonomy in Sect. 3.6 classifies systems by their capability profile rather than by a single label.

3.3 Degrees of autonomy

We define an operational four-level autonomy spectrum that is used throughout the review. Existing proposals use autonomy in two incompatible ways. Classical agent theory treats it as a defining property: an agent either has control over its own actions and internal state or it does not (Wooldridge and Jennings 1995). Recent conceptual work instead separates bounded AI agents from orchestrated agentic AI (Sapkota et al. 2026). Neither provides criteria that a reader can apply to a specific system. We therefore define the levels by three observable properties, each of which can be checked from a system's published description:

- C1, control-flow authority: who decides the next step, the developer's fixed program or the model;
- C2, external-effect scope: whether the system can change the state of anything outside its own context;
- C3, temporal and organisational extent: whether it operates across a single task episode, or across sessions and multiple agents.

These properties are chosen because they track the two sources of risk identified in this review. The first is decisions that are not independently checked (C1). The second is consequences that cannot be undone or that accumulate (C2 and C3). The internal–external action split of cognitive architectures (Sumers et al. 2024) makes C2 concrete.

Each level adds exactly one property to the level below it (Table 3). The spectrum is therefore ordinal and cumulative, and a system is assigned the highest level whose criteria it meets. On this reading, Sapkota et al.'s "AI agents" correspond mainly to L3 and their "agentic AI" to L4. We treat their distinction as a boundary on one scale rather than as a difference in kind. Each step up the spectrum widens what the system can do and, at the same time, the scope for errors to go unchecked and for harm to accumulate. Sections 3.7 and 8 return to this trade-off.

Table 3 Criteria for the four-level autonomy spectrum. C1 control-flow authority; C2 external-effect scope; C3 temporal and organisational extent. Each level adds one property to the level below it; a system is assigned the highest level whose criteria it meets

Level	C1: who decides the next step	C2: external effects	C3: extent	Diagnostic question	Examples
L1 Assisted response	Developer (single call)	None	Single turn	Does the system produce one output and stop?	Single-turn LLM use; chain-of-thought
L2 Scaffolded multi-step	Developer-fixed pipeline	None or fixed	One episode	Is the sequence of steps fixed in code?	Self-consistency; tree-of-thoughts search
L3 Self-directed tool use	Model chooses actions	Yes, via tools or APIs	One episode	Does the model decide which external action to take next?	ReAct; Reflexion; Toolformer; SWE-agent
L4 Long-horizon or multi-agent	Model or orchestrating agents	Yes	Across sessions or several agents	Does behaviour persist across sessions or span several agents?	Voyager; AutoGen; MetaGPT; Generative Agents

3.4 Agent–environment interaction

Agentic behaviour forms a closed loop between agent and environment, a generalisation of the perception–action cycle of sequential decision-making (Sutton and Barto 2018). Cognitive-architecture accounts divide the action space into internal actions on memory (reading, writing and reflecting) and external actions on the environment (tool calls, messages and physical effects), both governed by a decision procedure (Sumers et al. 2024). The distinction matters for safety. Internal actions are cheap and can usually be undone, whereas external actions may be consequential and irreversible. For that reason the framework in Sect. 3.7 places a control point between the decision and any external action.

3.5 Core capabilities

Six capabilities serve as the primary analytical dimensions, and evaluation cuts across all of them as a seventh.

Planning turns a goal into steps. Methods range from chain-of-thought prompting (Wei et al. 2022) and self-consistency (Wang X et al. 2023) to explicit search (Yao et al. 2023a) and delegation to symbolic planners (Liu et al. 2023). The evidence that autonomous plan generation is unreliable (Kambhampati et al. 2024) qualifies all of them.

Memory preserves state beyond the context window. It does so through retrieval (Lewis et al. 2020), managed long-term stores (Packer et al. 2024; Zhong et al. 2024) and memory streams (Park et al. 2023).

Tool use enlarges the action space (Schick et al. 2023; Qin et al. 2024; Wang X et al. 2024).

Feedback and reflection let agents revise their behaviour (Shinn et al. 2023; Madaan et al. 2023). Intrinsic self-correction, however, is not dependable (Huang J et al. 2024).

Coordination spreads work across agents (Wu et al. 2024; Du et al. 2024), with benefits that depend on the conditions (Cemri et al. 2025).

Safety concerns the prevention of harmful actions (Sect. 8). Alignment methods shape model behaviour (Ouyang et al. 2022; Bai et al. 2022), but agentic settings add risks tied to action: indirect prompt injection (Greshake et al. 2023; Zhan et al. 2024), tool misuse (Ruan et al. 2024) and harmful behaviour induced by jailbreaks (Andriushchenko et al. 2025). He et al. (2025) survey these risks in depth. Safety is therefore treated here as a capability built into the decision loop, not as a filter added downstream.

3.6 A unified taxonomy

The capabilities above form the axes of a unified taxonomy that classifies systems by capability profile along seven dimensions: planning, memory, tool use, coordination, reflection, safety and evaluation. Each dimension has sub-categories drawn from the surveyed literature. Table 4 lists the dimensions, their sub-categories and representative systems, and Fig. 4 presents the same structure as a hierarchy. For each dimension a system is assigned the sub-category that best describes its dominant mechanism. Because systems increasingly combine mechanisms (for example, episodic reflection over a retrieval-augmented store), the sub-categories are not mutually exclusive. The full description of a system is therefore its profile across all seven dimensions, not any single position.

Two dimensions play a different role from the other five, and we state the difference explicitly to avoid double counting.

Safety appears both as a taxonomy dimension and as a stage of the reference framework, and the two uses answer different questions. As a taxonomy dimension it is descriptive: it records which safety mechanism a system actually has, for example alignment training, sandboxing or a policy constitution. As a framework stage (Sect. 3.7) it is normative: it states where safety must be enforced, namely as a gate on the path from decision to external action, whatever mechanism implements it. A system can therefore have a safety mechanism in the taxonomic sense, such as

alignment training, and still lack the framework's gate. The mapping in Sect. 3.8 shows that this is the common case.

Evaluation is a meta-dimension. It does not describe a mechanism inside the agent. It describes how, and how validly, the agent's other six capabilities are measured. It is therefore the only dimension that applies to all the others. Section 9 treats it in this way: it specifies what must be measured for each capability and where current benchmarks fall short (Sect. 9.2).

Table 4 Unified taxonomy of agentic AI systems. Dimensions are the seven analytical axes; representative systems illustrate each sub-category and are not exhaustive. Asterisks mark preprints

Dimension	Sub-category	Representative systems
Planning	Linear prompting	Chain-of-thought (Wei et al. 2022); self-consistency (Wang X et al. 2023)
	Reason-and-act loop	ReAct (Yao et al. 2023b)
	Deliberate search	Tree of Thoughts (Yao et al. 2023a); world-model search (Hao et al. 2023)
	Neuro-symbolic / verifier-in-loop	LLM+P* (Liu et al. 2023); LLM-Modulo (Kambhampati et al. 2024)
Memory	Working / in-context	ReAct (Yao et al. 2023b)
	Managed long-term	MemGPT* (Packer et al. 2024); MemoryBank (Zhong et al. 2024)
	Episodic and reflective	Generative Agents (Park et al. 2023); Reflexion (Shinn et al. 2023)
	Retrieval-augmented / scalable	RAG (Lewis et al. 2020); A-MEM (Xu et al. 2025); Mem0* (Chhikara et al. 2025)
Tool use	Self-supervised API use	Toolformer (Schick et al. 2023)
	Large-scale API mastery	ToolLLM (Qin et al. 2024)
	Retrieval-aware calling	Gorilla (Patil et al. 2024)
	Code-as-action	CodeAct (Wang X et al. 2024); HuggingGPT (Shen et al. 2023)
	Protocol-mediated tools	Model Context Protocol ecosystem* (Hou et al. 2025)
Coordination	Single agent	ReAct (Yao et al. 2023b)
	Conversational MAS	AutoGen (Wu et al. 2024); CAMEL (Li et al. 2023)
	Role- or SOP-structured MAS	MetaGPT (Hong et al. 2024); ChatDev (Qian et al. 2024)
	Debate / deliberation	Multi-agent debate (Du et al. 2024)
Reflection	None / implicit	Chain-of-thought (Wei et al. 2022)
	Self-feedback refinement	Self-Refine (Madaan et al. 2023)
	Verbal RL / episodic reflection	Reflexion (Shinn et al. 2023)
	Tool-grounded / external verification	CRITIC (Gou et al. 2024); LLM-Modulo (Kambhampati et al. 2024)
Safety	Alignment-level	InstructGPT (Ouyang et al. 2022); Constitutional AI* (Bai et al. 2022)
	Action gating / sandboxing	ToolEmu (Ruan et al. 2024)
	Threat-specific defence and evaluation	InjecAgent (Zhan et al. 2024); AgentDojo (Debenedetti et al. 2024); AgentHarm (Andriushchenko et al. 2025)
	Constitution / policy for agents	TrustAgent (Hua et al. 2024)
Evaluation	Capability suites	AgentBench (Liu et al. 2024)
	Web / computer use	WebArena (Zhou et al. 2024); OSWorld (Xie et al. 2024); Mind2Web (Deng et al. 2023)

Dimension	Sub-category	Representative systems
	Assistant / tool interaction	GAIA (Mialon et al. 2024); τ -bench* (Yao et al. 2024)
	Software engineering and workplace	SWE-bench (Jimenez et al. 2024); SWE-agent (Yang J et al. 2024); TheAgentCompany* (Xu et al. 2024)

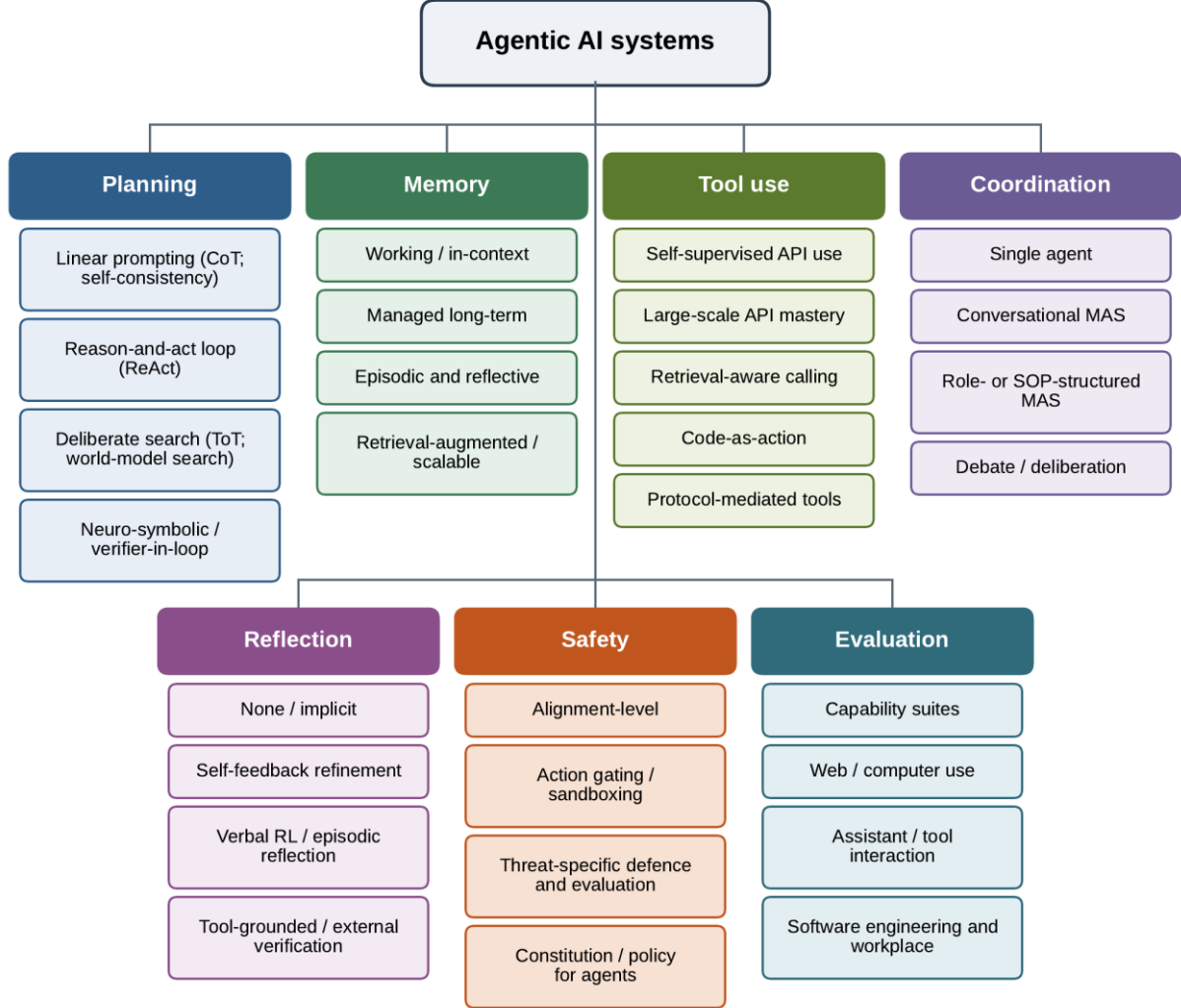


Fig. 4 Unified taxonomy of agentic AI systems across seven analytical dimensions, with the sub-categories used in Table 2

3.7 A closed-loop reference framework

We combine the capabilities into a single closed-loop reference framework for safe agentic decision-making (Fig. 5). The framework keeps the backbone of cognitive-architecture accounts: modular memory, an action space split into internal and external actions, and a decision procedure (Sumers et al. 2024). It then adds two elements that the evidence reviewed in later sections indicates are needed.

The first is an explicit verification stage between the decision and any external action. Autonomous plan generation is unreliable and improves when paired with external verifiers

(Kambhampati et al. 2024), and reflection helps mainly when it is grounded in trustworthy external feedback (Huang J et al. 2024; Gou et al. 2024).

The second is a safety-and-authorisation layer that gates external actions. External actions can be irreversible, and tool use exposes agents to untrusted content and injected instructions (Greshake et al. 2023; Zhan et al. 2024). The layer enforces sandboxing and policy checks (Ruan et al. 2024; Hua et al. 2024) at the internal–external boundary described in Sect. 3.4.

Operational definition. To make the framework testable rather than merely illustrative, we define one pass through the loop at step t . The agent receives an observation o_t and retrieves a memory context $r_t = R(M_t, o_t)$ from its memory store M_t . The policy proposes an action $a_t = \pi(o_t, r_t)$. The action carries a type, internal or external, and a reversibility label. The verifier returns $v_t = V(a_t, o_t, r_t) \in \{\text{pass}, \text{revise}\}$, together with a critique. The decisive requirement is that V draws on a signal independent of π , such as a test, a tool result, a formal checker or a separate model with access to ground truth. For every external action, the gate returns $g_t = G(a_t, P, \rho_t) \in \{\text{admit}, \text{block}, \text{confirm}\}$, where P is an explicit permission policy and ρ_t is a risk estimate. If $g_t = \text{confirm}$, a human decision is required. Only admitted actions reach the environment. The reflection step then writes an outcome record to memory, $M_{t+1} = W(M_t, o_t, a_t, o_{t+1})$, with provenance attached.

Table 5 gives, for each of the eight stages, its operational definition, the artefact that shows it exists, a minimal test of whether it works, and a metric. The same definitions are used as coding rules in Sect. 3.8.

The framework serves two purposes. Looking back, it describes existing systems, most of which implement only some of the stages (Sect. 3.8). Looking forward, it indicates where gains in reliability and safety are most likely to come from: the stages that are usually missing, namely verification and the safety gate. The framework is an architectural hypothesis derived from the evidence. It has not itself been validated as a whole in an end-to-end empirical study. Such a study is included in the roadmap as part of the safe tool-grounded decisions direction (Sect. 12).

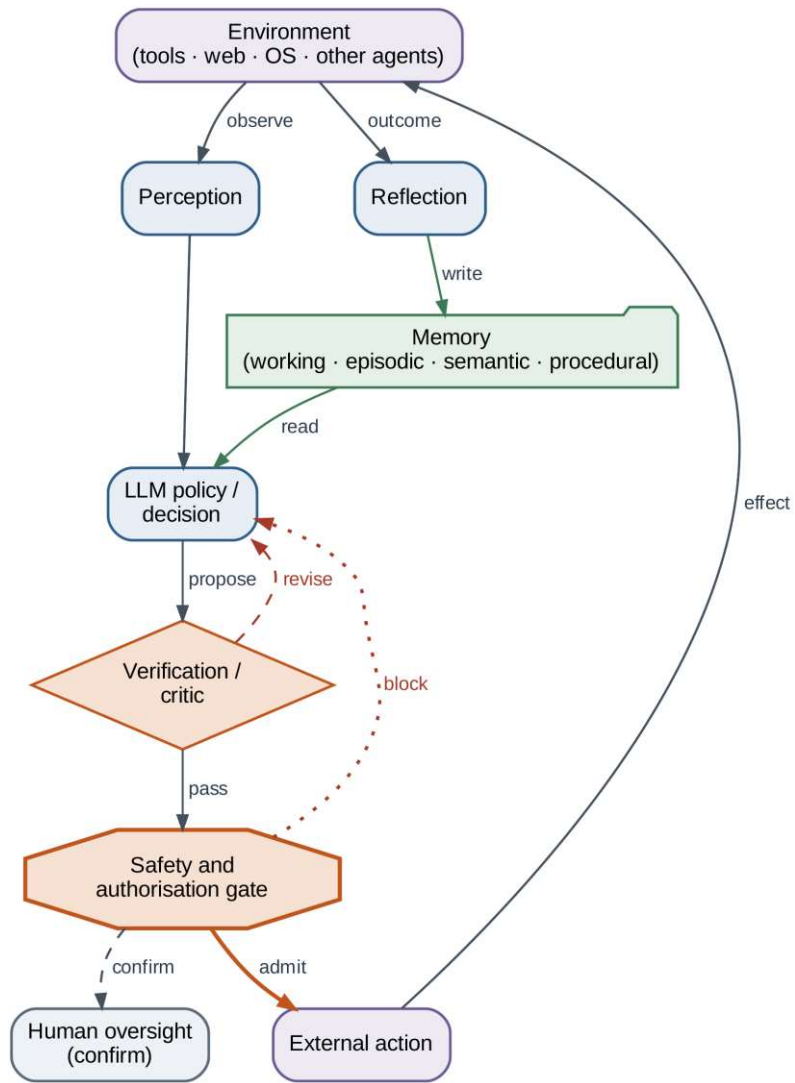


Fig. 5 Closed-loop reference framework for safe agentic decision-making. An LLM policy reads modular memory and proposes an action; an explicit verification stage and a safety-and-authorisation gate sit between the decision and any external action. The gate admits the action, returns it for revision, blocks it or requests human confirmation; outcomes are reflected into memory. Solid arrows show the main flow, dashed arrows revision or confirmation, and dotted arrows blocking

Table 5 Operational definitions of the eight stages of the reference framework (Fig. 5). The definitions also serve as coding rules for Table 6

Stage	Operational definition	Evidence that the stage exists	Minimal test	Metric
Perception	Maps raw inputs to observation o_t , with the source recorded	Observation parser; provenance tags	Injected content in inputs is labelled as untrusted	Provenance coverage
Memory read/write	Retrieves $r_t = R(M_t, o_t)$; writes with provenance	Explicit memory store and retrieval policy	Contradictory facts are detected on retrieval	Consistency rate; staleness
LLM policy	Proposes $a_t = \pi(o_t, r_t)$ with action type and reversibility label	Action schema with type labels	External actions are labelled correctly	Label accuracy
Verification	$V(a_t) \in \{\text{pass, revise}\}$ using a signal independent of π	Tests, tools, formal checker or separate critic	Seeded faulty actions are caught before execution	Fault-detection rate; false-revise rate
Safety and authorisation gate	$G(a_t, P, \rho_t) \in \{\text{admit, block, confirm}\}$ for every external action	Permission policy P ; risk estimate; confirmation path	Out-of-policy and injected actions are blocked	Attack success rate; utility under attack
External action	Executes only admitted actions, in a sandbox where possible	Sandbox or least-privilege execution	A blocked action produces no side effect	Unauthorised side-effect count
Observation return	Returns o_{t+1} and the execution status	Structured result channel	Failures are reported, not silently dropped	Unreported-failure rate
Reflection to memory	Writes $M_{t+1} = W(M_t, o_t, a_t, o_{t+1})$ with outcome and provenance	Reflection or outcome log	The outcome of a failed action is recorded and reused	Error-recurrence rate

3.8 Retrospective instantiation of the framework

To test whether the framework describes real systems rather than an ideal, we mapped ten representative systems onto its eight stages (Table 6). The systems were chosen to span the main families of Table 4: reason-and-act, reflective and verifier-in-the-loop planners; managed and skill-library memory; code-as-action and tool-grounded critique; conversational and role-structured multi-agent frameworks; and a policy-based safety system.

Each cell was coded from the original publication using the definitions in Table 5, with three rules:

- "Yes": the paper describes a dedicated component that performs the stage's function by default.
- "Partial": the function exists only implicitly, only for some action types, or only as an optional configuration.
- "No": the paper describes no such function.

In the mapping, every system implements perception and a policy, and every system has at least a partial form of external action. Memory and reflection are common. Only two systems (LLM-Modulo and CRITIC), both designed around verification, have a verification stage coded "Yes". Only one (TrustAgent) has a safety gate coded "Yes". General-purpose frameworks such as AutoGen and SWE-agent provide partial controls, such as human-input modes, containerised execution or edit-time checks, but leave their configuration to the developer. None of the ten systems has both control stages coded "Yes".

To summarise the mapping numerically, each system receives two scores. The coverage index is the mean score over the seven coded stages, with Yes = 1, Partial = 0.5 and No = 0. The control index is the mean over the two control stages, verification and the safety gate.

Coverage indices range from 0.50 (ReAct) to 0.79 (Reflexion, Voyager), with a mean of 0.66. Control indices range from 0.00 (ReAct, MemGPT) to 0.75 (LLM-Modulo, TrustAgent), with a mean of 0.38. Across all ten systems, the five "doing" stages (perception, memory, policy, external action and reflection) score 0.78 on average, while the two "checking" stages score 0.38. The gap between doing and checking is therefore about 0.40 on a 0–1 scale.

The sample is small and purposive, so the mapping does not estimate the prevalence of these stages in the field. It does show that the pattern the framework predicts, namely strong machinery for doing and sparse machinery for checking, holds across the main families of systems.

Table 6 Retrospective instantiation of the reference framework (Fig. 5) for ten representative systems, based on the mechanisms described in the original publications. Yes, explicit dedicated component; Partial, partial, implicit or optional component; No, component absent. Asterisks mark preprints. Coverage index = mean over the seven coded stages (Yes = 1, Partial = 0.5, No = 0); control index = mean over verification and the safety gate

System	Perception	Memory read/write	LLM policy	Verification	Safety and authorisation gate	External action	Reflection to memory	Coverage index	Control index	Note
ReAct (Yao et al. 2023b)	Yes	Partial	Yes	No	No	Yes	No	0.50	0.00	Context-only memory; no check before acting
Reflexion (Shinn et al. 2023)	Yes	Yes	Yes	Partial	No	Yes	Yes	0.79	0.25	Evaluator signal (e.g. tests) drives episodic reflection
MemGPT* (Packer et al. 2024)	Yes	Yes	Yes	No	No	Partial	Partial	0.57	0.00	Model-managed paging between context and storage
Voyager (Wang G et al. 2024)	Yes	Yes	Yes	Partial	No	Yes	Yes	0.79	0.25	Self-verification plus environment feedback; skill library
MetaGPT (Hong et al. 2024)	Yes	Yes	Yes	Partial	No	Yes	Partial	0.71	0.25	Role hierarchy with review steps and shared messages
AutoGen (Wu et al. 2024)	Yes	Partial	Yes	Partial	Partial	Yes	No	0.64	0.50	Optional human input and containerised code execution
SWE-agent (Yang J et al. 2024)	Yes	Partial	Yes	Partial	Partial	Yes	No	0.64	0.50	Edit-time linting; sandboxed repository environment
CRITIC (Gou et al. 2024)	Yes	No	Yes	Yes	No	Partial	Partial	0.57	0.50	Tool-grounded critique before final answer
LLM-Modulo (Kambhampati et al. 2024)	Yes	Partial	Yes	Yes	Partial	Partial	Partial	0.71	0.75	External critics verify plans; back-prompting
TrustAgent (Hua et al. 2024)	Yes	Partial	Yes	Partial	Yes	Yes	No	0.71	0.75	Agent constitution applied before, during and after planning

3.9 A quantitative model of verification and gating over long horizons

To show why the two control stages matter, and why independence of verification matters in particular, we add a minimal reliability model to the framework. It is deliberately simple, and each parameter corresponds to a measurable quantity in Table 5.

Consider an episode with H consequential external actions. At each action the policy proposes an erroneous or unsafe action with probability e . The verifier detects such an action with nominal probability d . However, a verifier that shares the policy's blind spots, as in self-verification, detects only a fraction of what it would detect independently. We capture this with a correlation parameter $\rho \in [0, 1]$, so that the effective detection rate is $d(1 - \rho)$. The gate then blocks a fraction g of the unsafe actions that remain. The residual per-action failure rate is

$$p = e \cdot [1 - d(1 - \rho)] \cdot (1 - g),$$

and, if failures are independent across actions, the probability that an episode completes with no failed action is $S(H) = (1 - p)^H$.

Two consequences follow (Fig. 6).

First, reliability requirements tighten sharply with the horizon. Reaching $S = 0.9$ over $H = 50$ actions requires $p \leq 1 - 0.9^{(1/50)} \approx 0.0021$, more than twenty times lower than an unaided per-action error rate of $e = 0.05$.

Second, the value of verification depends on its independence. With illustrative values $e = 0.05$ and $d = 0.8$, a self-verifier with $\rho = 0.9$ lowers p only from 0.050 to 0.046, and $S(50)$ rises only from 0.08 to 0.10. An independent verifier with $\rho = 0.1$ lowers p to 0.014 ($S(50) = 0.49$). Adding a gate with $g = 0.5$ lowers p to 0.007 ($S(50) = 0.70$).

The parameter values are illustrative, not estimates, and the independence assumption across steps is optimistic, because errors in real agents compound. The model is nevertheless consistent with the empirical pattern in Sects. 4 and 9. Intrinsic self-correction adds little (Huang J et al. 2024; Stechly et al. 2024), externally grounded critique helps (Gou et al. 2024; Kambhampati et al. 2024), and success falls with task length and with repeated trials (Xie et al. 2024; Yao et al. 2024). The model also yields testable predictions for the end-to-end evaluation proposed in Sect. 12. Success should fall roughly geometrically with the number of consequential actions. Ablating the verifier's independence should change that slope more than ablating its nominal accuracy.

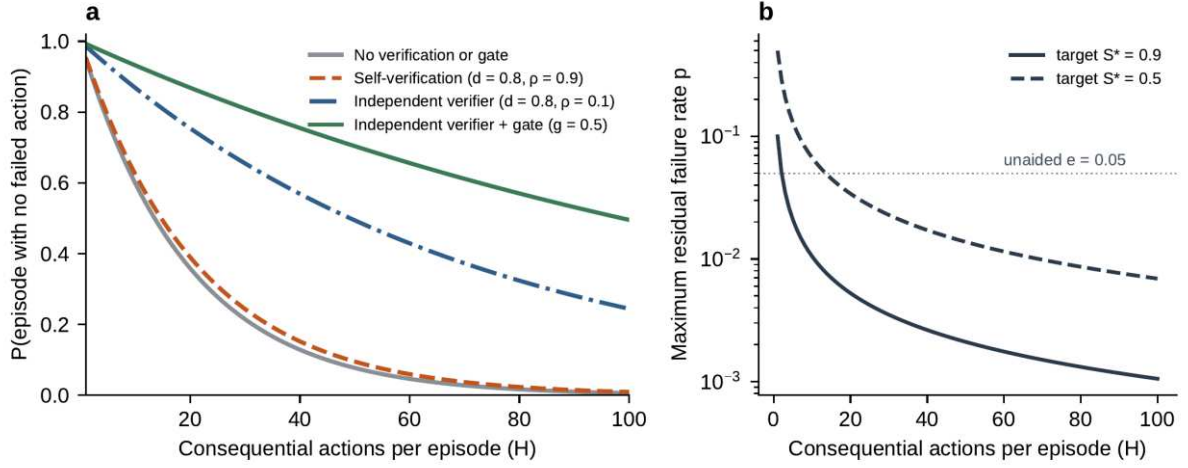


Fig. 6 Reliability model of the reference framework (Sect. 3.9). a Probability that an episode of H consequential actions contains no failed action, for an unaided policy (per-action error $e = 0.05$), self-verification ($d = 0.8$, correlation $\rho = 0.9$), an independent verifier ($d = 0.8, \rho = 0.1$), and an independent verifier plus a safety gate ($g = 0.5$). b Maximum residual per-action failure rate p that allows a target episode success S^* over H actions; the dotted line marks the unaided error rate. Parameter values are illustrative, not estimates

4 Planning, reasoning and reflection

4.1 The planning problem

Planning converts a goal into a course of action and revises that course as the world responds. In agentic settings it cannot be separated from reasoning and reflection. This section compares the main planning families along eight dimensions: mechanism, computational assumption, strength, weakness, suitable tasks, scalability to long horizons, dependence on external feedback and safety implication (Table 7). It locates these families in a plan–execute–reflect–replan pipeline that shows where verification and safety gating intervene (Fig. 7), and it closes by explaining why planning remains unreliable for open-ended autonomy. A dedicated survey breaks agent planning into task decomposition, plan selection, external modules, reflection and memory (Huang X et al. 2024). We examine each of these critically. Methods are described in terms of technique and reported outcomes only; no model’s private reasoning traces are reproduced.

4.2 Goal formulation and decomposition

An under-specified goal must first be divided into subgoals. Plan-and-solve prompting asks the model to write a plan and then carry it out, which improves zero-shot multi-step reasoning (Wang L et al. 2023). Role-structured pipelines move decomposition out of a single model and across several agents (Hong et al. 2024). Decomposition turns one hard problem into several easier ones and exposes structure that can later be checked. Its errors, however, propagate: the model that decomposes is the same fallible policy that executes the steps, so decomposition on its own provides no independent check.

4.3 Prompt-based reasoning and reasoning models

Chain-of-thought prompting improves multi-step reasoning by eliciting intermediate steps (Wei et al. 2022). Self-consistency samples several reasoning paths and keeps the most frequent answer; with a 540-billion-parameter model it raised accuracy on the GSM8K arithmetic benchmark by 17.9 percentage points (Wang X et al. 2023). Neither method needs training, tools or an environment. Both, however, are open-loop: nothing outside the model checks the steps, so plausible errors pass unnoticed, and the cost of self-consistency grows with the number of samples. They work well for short, self-contained problems and scale poorly to long horizons.

A more recent line of work trains reasoning behaviour directly with reinforcement learning. DeepSeek-R1 showed that large-scale reinforcement learning with verifiable rewards can produce long, self-reflective reasoning traces and strong results in mathematics and programming (Guo et al. 2025). Such reasoning models raise the ceiling for single-step reasoning, but they do not remove the limits discussed below. Their training reward is available precisely because the tasks are verifiable. Accuracy of large reasoning models can also collapse beyond a threshold of problem complexity (Shojaee et al. 2025). For open-ended agent tasks without a reliable verifier, the core difficulty therefore remains.

4.4 Reasoning and acting

Interleaving reasoning with actions lets an agent collect information and adapt. The original ReAct study reported absolute improvements of about 34 and 10 percentage points on two interactive benchmarks, ALFWorld and WebShop, over imitation and reinforcement learning baselines (Yao et al. 2023b). Grounding reasoning in observations reduces hallucination that the environment could have corrected. A sensitivity analysis (a preprint), however, finds that much of the benefit comes from the similarity between exemplars and queries rather than from interleaving itself, and that performance is fragile under changes to prompts and exemplars (Verma et al. 2024). ReAct-style agents consume environmental feedback but do not verify their own actions. Without a gate, instructions injected into an observation can redirect them (Greshake et al. 2023).

4.5 Search-based and model-predictive planning

Search-based methods explore several paths instead of committing to one. Tree-of-thoughts planning searches over evaluated intermediate states and raised success on the Game of 24 from 4% with chain-of-thought prompting to 74% (Yao et al. 2023a). Graph-structured variants generalise the topology, but within the verified corpus the tree formulation is the concrete representative. Model-predictive planning simulates the consequences of actions with a world model and uses the simulation to guide search (Hao et al. 2023). It is the LLM-era counterpart of

model-based planning (Sutton and Barto 2018) and of systems that combine search with learning (Silver et al. 2016). Search delivers large look-ahead gains but multiplies model calls. It is limited by the quality of the state evaluator, which Sect. 4.7 shows to be unreliable. Model-predictive planning is limited in turn by how faithful the world model is.

4.6 Hierarchical, symbolic, neural and neuro-symbolic planning

Hierarchical approaches plan at several levels of abstraction, from simple two-level prompting (Wang L et al. 2023) to deep role hierarchies (Hong et al. 2024). Abstraction keeps the combinatorial explosion in check but generalises poorly when a task does not fit the assumed structure.

Three paradigms lie behind these families. Symbolic planning hands the problem to a sound external solver, which guarantees correctness but requires a formal domain model (Liu et al. 2023; Russell and Norvig 2021). Neural planning relies on the model end to end: it is flexible but offers no soundness. Neuro-symbolic hybrids pair a flexible generator with external verification. The most developed critical position holds that autoregressive models cannot plan or verify their own plans reliably. In that study only about 12% of the plans generated by the strongest model were executable and reached the goal, and the authors propose an "LLM-Modulo" arrangement in which a generator is surrounded by external critics (Kambhampati et al. 2024). On current evidence, the hybrid paradigm is the most promising route to reliability.

4.7 Reflection, verification and replanning

Across repeated trials, storing verbal reflections in an episodic buffer improves performance (Shinn et al. 2023). Iterative self-feedback within a single model improves outputs by about 20% on average across seven tasks (Madaan et al. 2023). Both results need strong qualification. Without external feedback, self-correction often fails to improve reasoning and can degrade it (Huang J et al. 2024). Models are also poor at verifying their own reasoning and plans (Stechly et al. 2024, preprint). Correction grounded in something external does better: critiquing outputs with tools improves results (Gou et al. 2024), in line with the verifier-in-the-loop position (Kambhampati et al. 2024).

The reconciling interpretation is that reflection helps to the extent that it is anchored to a reliable external signal, and that it can entrench mistakes when it refers only to itself. This is the central design lesson of the section. Replanning helps within an episode (Shinn et al. 2023; Yao et al. 2023b), but it does not solve the long-horizon problem. Performance on open-ended computer tasks falls steeply as tasks lengthen (Xie et al. 2024), only a minority of realistic web tasks are

solved (Zhou et al. 2024), and the pass^k metric shows that the probability of succeeding on all k repeated attempts at the same task drops quickly as k increases (Yao et al. 2024).

4.8 Cost, error accumulation, false progress and failure modes

Four cross-cutting liabilities affect every planning family.

Cost. Search and sampling multiply model calls (Yao et al. 2023a; Wang X et al. 2023), and evaluations often omit cost, which distorts comparisons (Kapoor et al. 2024).

Error accumulation. Small per-step errors compound over long horizons (Xie et al. 2024), and the accuracy of reasoning models can collapse beyond a threshold of problem complexity (Shojaee et al. 2025).

Reward hacking and false progress. Optimising a proxy signal, such as a self-generated critique, a weak test or an exploitable benchmark check, produces confident but false progress. The evidence for this is indirect but consistent: failures of task verification in multi-agent systems (Cemri et al. 2025), unreliable self-verification (Stechly et al. 2024, preprint) and benchmark checks that can be satisfied through leaked solutions (Aleithan et al. 2024, preprint; Kapoor et al. 2024).

Documented failure modes. These include the brittleness of the reasoning-and-acting pattern (Verma et al. 2024, preprint), unreliable autonomous planning (Kambhampati et al. 2024), self-correction that makes answers worse (Huang J et al. 2024) and the failure taxonomy for multi-agent systems (Cemri et al. 2025).

Table 7 Comparison of planning and reasoning methods for agentic AI. Representative systems are illustrative; preprints among the supporting critiques are identified in the text

Method family	How it works	Computational assumption	Key strength	Key weakness	Suitable tasks	Long-horizon scalability	External-feedback dependence	Safety implication
Chain-of-thought (Wei et al. 2022)	Elicits intermediate steps in one pass	Useful structure can be surfaced within one context	Cheap; needs no tools or training	Open-loop; steps unverified	Short, self-contained reasoning	Low	None	Low action risk alone; unsafe once connected to tools
Self-consistency (Wang X et al. 2023)	Samples several paths and keeps the majority answer	More samples raise accuracy	+17.9 points on GSM8K	Cost grows with samples	Reasoning with checkable answers	Low	None	As above
RL-trained reasoning models (Guo et al. 2025)	Long reasoning traces learned with verifiable rewards	Reliable reward signal during training	Strong maths and coding results	Depends on verifiable tasks; complexity collapse	Verifiable reasoning	Low–moderate	High during training	Stronger reasoning does not replace an action gate
ReAct-style (Yao et al. 2023b)	Interleaves reasoning with actions	Observations are informative	Grounded and adaptive	Brittle; gains partly exemplar-driven	Interactive, observation-rich tasks	Moderate	Uses feedback but does not verify	Emits external actions; needs a gate
Tree or graph search (Yao et al. 2023a)	Explores and scores partial solutions	A usable state evaluator exists	Strong look-ahead (4% to 74%)	Costly; bounded by self-evaluation	Puzzles with checkable states	Low	High (internal evaluator)	Mostly internal until outputs act
Model-predictive (Hao et al. 2023)	Simulates outcomes to guide search	World model predicts adequately	Avoids dead ends	Limited by model fidelity	Domains with predictable dynamics	Low–moderate	Model-internal prediction	Can pre-screen hazards the model captures
Hierarchical (Wang L et al. 2023; Hong et al. 2024)	Plans across abstraction levels	Task fits the assumed structure	Controls combinatorial growth	Poor generalisation outside the structure	Well-structured workflows	Good within domain	Varies	Level boundaries aid gating
Symbolic delegation (Liu et al. 2023)	Delegates to a sound planner	A formal domain model exists	Soundness guarantees	Formal model is costly to build	Formalisable planning	High within domain	External solver	Verifiable plans improve safety
Neuro-symbolic / verifier-in-loop (Kambhampati et al. 2024)	Model generates; critics verify	A reliable external verifier exists	Reliability through an independent check	Verifier coverage limits scope	Correctness-critical planning	Best available route	High by design	Verification on the critical path
Self-reflection (Shinn et al. 2023; Madaan et al. 2023)	Revises from feedback	Feedback is informative	Gains across trials (~20% average)	Intrinsic correction can degrade reasoning	Retryable tasks with feedback	Moderate	High; helps only if feedback is reliable	Self-referential correction can entrench errors
Tool-grounded critique (Gou et al. 2024)	Checks outputs against tools	Suitable verifying tools exist	Grounded correction works	Depends on tool availability	Tool-checkable outputs	Moderate	High (external)	External check reduces unsafe outputs

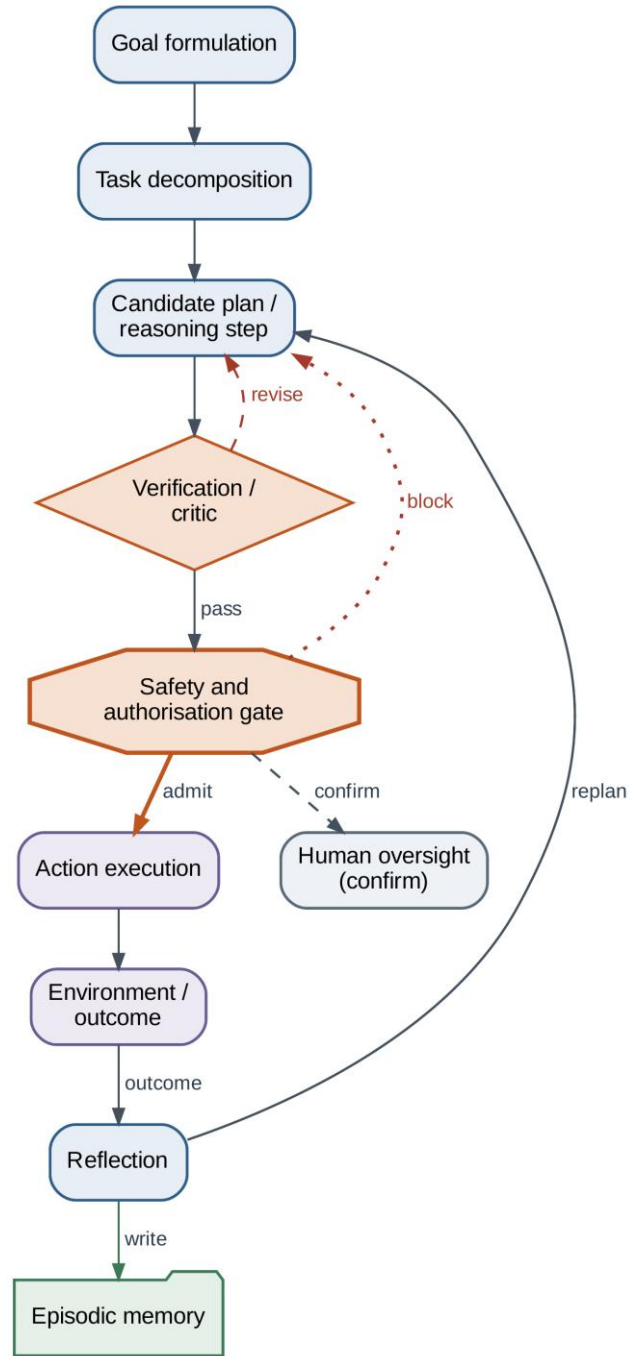


Fig. 7 Planning, execution, reflection and replanning pipeline, showing where verification and safety gating intervene before external actions. Line styles as in Fig. 5

4.9 Why planning remains unreliable for long-horizon autonomy

Taken together, the reviewed literature is consistent with a specific conclusion (claim C1 in Sect. 11): the methods that do most to improve short-horizon reasoning do not, on the available evidence, secure long-horizon reliability. Four structural reasons are offered in the literature.

First, the dominant paradigm is open-loop at the level that matters. Prompt-based reasoning, and even reasoning-and-acting, has no independent check on whether each step is correct (Wei et al. 2022; Yao et al. 2023b; Verma et al. 2024).

Second, the obvious remedy, self-correction, is least dependable exactly where it is most needed. Intrinsic self-verification does not work consistently and can harm reasoning (Huang J et al. 2024; Stechly et al. 2024).

Third, autonomous plan generation is weak in itself: only about one plan in eight was valid in the planning study cited above. This suggests that greater model capability will not close the gap on its own (Kambhampati et al. 2024).

Fourth, errors accumulate over long horizons and can be hidden by apparent progress against flawed success signals (Xie et al. 2024; Cemri et al. 2025; Kapoor et al. 2024).

The constructive implication recurs across these studies, although it rests mainly on planning and reasoning benchmarks rather than on deployed agents. Reliability improves when generation is separated from verification and the verifier draws on something outside the generating model (Liu et al. 2023; Kambhampati et al. 2024; Gou et al. 2024). It does not improve when the model grades its own work. This motivates, although it does not by itself prove, an architecture in which every consequential action passes an independent check and a safety gate, which is the hypothesis encoded in the framework of Sect. 3.7.

5 Memory architectures

5.1 Memory as a first-class capability

What an agent can do over time is limited by what it can remember. Cognitive-architecture accounts describe memory as modular: a working memory plus long-term episodic, semantic and procedural stores, accessed through internal read and write actions (Sumers et al. 2024). This section reviews the substrates on which memory is built, the policies that manage them, and the reliability and safety problems that remain open (Table 8; Fig. 8).

5.2 Substrates

The simplest form of memory is the model's own context, which holds information for the duration of a task (Yao et al. 2023b). Operating-system-style management moves information between the context window and external storage under the model's control (Packer et al. 2024). Long-term stores keep knowledge across sessions, either with principled updating and decay (Zhong et al. 2024) or as memory streams that support reflection and planning (Park et al. 2023).

The cognitive-architecture taxonomy distinguishes three kinds of long-term memory (Sumers et al. 2024). Episodic memory records experiences; the reflection buffer that supports learning across trials is an example (Shinn et al. 2023). Semantic memory holds general facts. Procedural memory holds reusable skills (Wang G et al. 2024).

The dominant durable substrate is a vector store queried by similarity. It is the agentic form of retrieval-augmented generation (Lewis et al. 2020) and of its precursors and variants (Guu et al. 2020; Borgeaud et al. 2022; Gao et al. 2023). Agentic memory systems add selective retrieval and dynamic organisation. A-MEM reports token savings of 85–93% relative to baselines (Xu et al. 2025), and Mem0 reports about 90% fewer tokens and 91% lower 95th-percentile latency than full-context methods (Chhikara et al. 2025, preprint). Knowledge-graph memory stores entities and relations instead of opaque vectors. It aligns naturally with semantic memory, but on the verified evidence its systematic comparison against vector stores in agentic settings is still missing.

5.3 Policies

Memory operations are explicit reads and writes (Sumers et al. 2024). They are implemented as function calls issued by the model (Packer et al. 2024) or as automatic note creation and linking (Xu et al. 2025), and the quality of retrieval largely determines their value (Lewis et al. 2020). The Generative Agents architecture ranks memories for retrieval by recency, relevance and an explicit importance score (Park et al. 2023). Summarisation and compression make memories fit a finite context (Packer et al. 2024; Chhikara et al. 2025; Xu et al. 2025), but they lose detail. Deliberate forgetting keeps stores current, for instance through a decay curve modelled on Ebbinghaus's forgetting curve (Zhong et al. 2024). A heuristic that forgets the wrong item, however, erodes the very competence that memory is meant to provide.

5.4 Reliability and safety problems

As memory accumulates, it gathers contradictions, outdated entries and inconsistencies over time. Current agent memory is mostly additive. Principled conflict resolution, detection of staleness and maintenance of temporal consistency are much less developed than storage and retrieval (Zhang et al. 2024). Consolidation-based systems have begun to merge and update entries instead of only appending them (Chhikara et al. 2025). No standard mechanism yet exists for detecting a conflict, deciding which version supersedes another, or reconciling timestamps. This is a first-order reliability gap, because an agent acting on a stale or contradicted memory fails with full confidence.

Memory is also an attack surface. Adversarial content written into memory can steer later behaviour (He et al. 2025; Greshake et al. 2023). Durable memory and interaction logs retain user data, which increases exposure of personal information (He et al. 2025). Moving knowledge out of model weights and into memory avoids catastrophic forgetting at the weight level (Wang G et al. 2024; Packer et al. 2024). It does so, however, by shifting the problem to retrieval and consistency, neither of which is solved. How durable memories could be safely consolidated into model parameters remains an open question.

5.5 Unresolved gaps in memory

Four gaps stand out.

Persistence semantics. Durable semantic knowledge and short-lived episodic knowledge are rarely distinguished (Sumers et al. 2024; Zhang et al. 2024).

Conflict and temporal consistency. There is no standard detection or resolution mechanism (Zhang et al. 2024; Chhikara et al. 2025).

Structured versus unstructured memory. Knowledge-graph memory lacks head-to-head evaluation in agentic settings.

Secure and private memory. Defences against poisoning and principled handling of personal data are at an early stage (He et al. 2025; Greshake et al. 2023).

Table 8 Comparison of memory mechanisms in agentic AI. Representative systems are illustrative; asterisks mark preprints

Memory mechanism	Representation	Write policy	Retrieval policy	Key strength	Key weakness or gap	Cost profile	Safety and privacy consideration
Context / working (Yao et al. 2023b)	In-context tokens	Kept in context	Attention over context	Fast; fully available	Bounded; lost when the task ends	Low	Transient; low retention risk
Managed long-term (Packer et al. 2024*)	Context plus external store	Model-issued paging	Paged recall	Extends the effective window	Heuristic paging; retrieval-bound	Moderate	External store persists data
Episodic and reflective (Park et al. 2023; Shinn et al. 2023)	Experience and reflection records	Append records	Recency, relevance and importance	Learning across trials	Heuristic importance scoring	Grows with experience	Accumulates user and context data
Semantic (Sumers et al. 2024)	Facts and knowledge	Curated or extracted	Fact or entity lookup	Stable general knowledge	Persistence semantics rarely modelled	Varies	Risk of knowledge poisoning
Procedural / skill library (Wang G et al. 2024)	Reusable skills	Add on success	Match skill to task	Reuse without re-derivation	Skill validity over time	Grows with skills	Executable skills widen the action surface
Vector store / retrieval-augmented (Lewis et al. 2020; Xu et al. 2025; Chhikara et al. 2025*)	Embeddings plus text	Append or consolidate	Similarity top-k	Scalable; token savings of about 85–93%	Retrieval caps output quality; weak conflict handling	Efficient at query time	Poisoning via corpus; privacy of stored text
Knowledge graph (structured)	Entities and relations	Relation insertion	Graph or relational query	Relational queries; contradiction checks	No systematic agentic evaluation (gap)	Construction and maintenance cost	Structure supports audit and privacy control

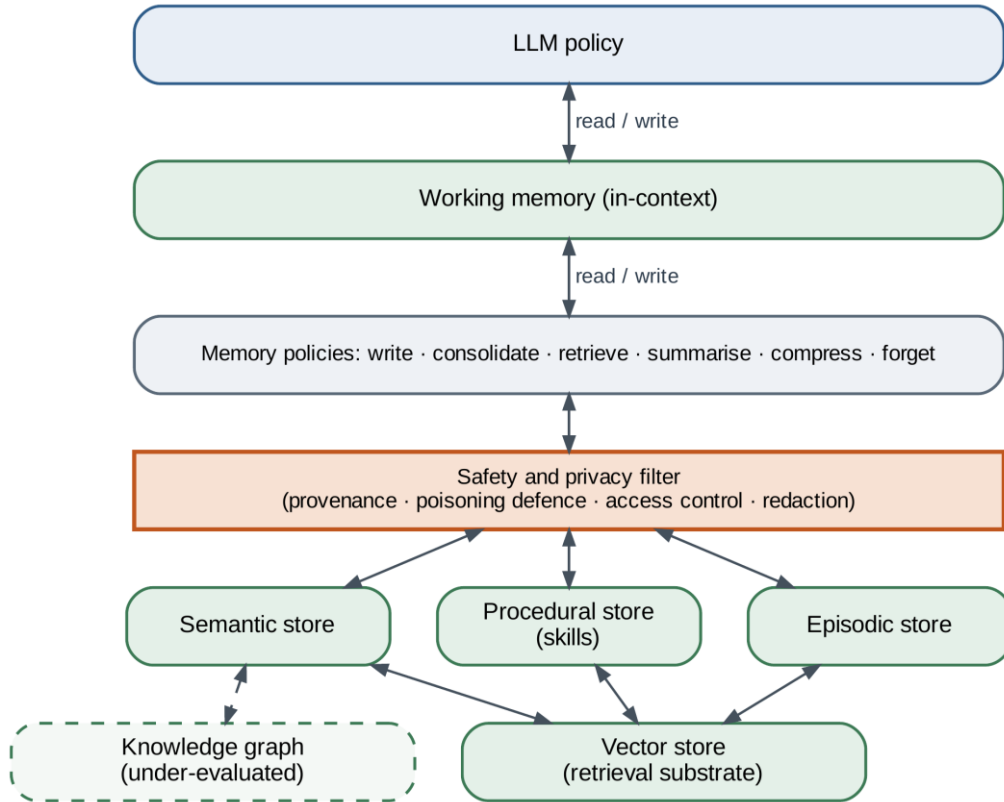


Fig. 8 Layered agent-memory architecture: working memory, a band of memory policies, a safety-and-privacy filter, and long-term episodic, semantic and procedural stores over a retrieval substrate. The dashed knowledge-graph store marks a substrate that has not been systematically evaluated in agentic settings

6 Tool use and external action

6.1 From reasoning to consequence

Tool use is the point at which an agent's decisions start to have consequences. It extends the action space to functions, APIs, browsers, databases, interpreters, search engines, scientific instruments and embodied effectors. In doing so it turns a reasoning system into an actor whose errors and vulnerabilities affect the outside world (Table 9; Fig. 9).

6.2 Mechanisms of external action

Function calling. The model emits a structured call, either learned in a self-supervised way (Schick et al. 2023) or trained across more than 16,000 real-world APIs (Qin et al. 2024).

API orchestration. A controller model composes several tools (Shen et al. 2023), helped by retrieval-aware tool selection (Patil et al. 2024). A survey of tool learning divides the process into task planning, tool selection, tool calling and response generation (Qu et al. 2025).

Standardised tool protocols. The interface between agents and tools is itself becoming standardised. The Model Context Protocol (MCP) is an open standard for connecting models to external tools and data sources, with bi-directional communication and dynamic discovery of capabilities. A systematic study of MCP maps the lifecycle of an MCP server into creation, deployment, operation and maintenance phases, and identifies sixteen threat scenarios spread across malicious developers, external attackers, malicious users and security flaws (Hou et al. 2025, preprint). Standardisation lowers integration costs, but it also concentrates risk: one compromised or poorly specified tool server can affect every agent that connects to it.

Browser use. Agents navigate and manipulate web pages. They have been evaluated on functional task completion (Zhou et al. 2024), generalist navigation across real websites (Deng et al. 2023) and grounded online shopping (Yao et al. 2022).

Database access. Querying and updating structured stores is a separate skill with its own evaluation environment (Liu et al. 2024).

Code execution. Code gives the agent a general and expressive action interface (Wang X et al. 2024). Software-engineering agents rely on it (Yang J et al. 2024; Jimenez et al. 2024).

Search tools. Retrieval of information is among the most dependable tool calls (Schick et al. 2023; Yao et al. 2023b).

Scientific tools. Agents drive instruments and solvers (Boiko et al. 2023; Romera-Paredes et al. 2024).

Embodied action. Decisions are linked to effectors, so far mainly in simulated worlds (Wang G et al. 2024; Yao et al. 2022). Physical robotics lies largely outside the verified LLM-agent corpus and is treated here as an adjacent frontier.

6.3 The tool-use pipeline

Reliable tool use is a pipeline, and each stage has its own characteristic failures.

Tool discovery becomes a retrieval problem once the number of tools is large (Qin et al. 2024; Patil et al. 2024). Tool selection chooses among the candidates (Qu et al. 2025). Argument generation must produce well-formed parameters (Patil et al. 2024).

Output validation is the stage most often missing. It checks results before they are used, either by critiquing them with tools (Gou et al. 2024) or by verifying them independently before acting (Kambhampati et al. 2024).

Recovery after a failed call relies on replanning (Shinn et al. 2023; Wang X et al. 2024). Measurements of reliability across repeated trials show that recovery is far from robust (Yao et al. 2024, preprint).

Transaction rollback, meaning the clean reversal of a failed or unsafe action, is a systems-level requirement that the verified corpus does not standardise. This is why the safety gate must screen actions before they execute, rather than relying on undoing them afterwards.

6.4 Permission, sandboxing and prevention of unsafe actions

Because external actions may be irreversible and may be steered by untrusted content, tool use cannot be separated from safety.

Permission control limits which actions are allowed and requires confirmation for high-stakes ones (Hua et al. 2024). Sandboxing contains the effects of execution (Ruan et al. 2024; Wang X et al. 2024).

Prevention of unsafe actions must address a well-documented threat surface. Indirect prompt injection was first demonstrated against LLM-integrated applications (Greshake et al. 2023) and has since been quantified for tool-using agents (Zhan et al. 2024) and evaluated in dynamic environments (Debenedetti et al. 2024). Jailbreaks can induce harmful behaviour in tool-using agents (Andriushchenko et al. 2025). A security survey catalogues the full range of threats (He et al. 2025), and tool protocols add supply-chain threats of their own (Hou et al. 2025).

The consistent lesson is that safety must be enforced at the boundary between internal and external actions defined in Sect. 3.4.

6.5 Unresolved gaps in tool use

Output validation is under-provided (Gou et al. 2024; Kambhampati et al. 2024). Recovery and rollback are immature, so irreversible actions must be prevented rather than undone (Shinn et al. 2023; Wang X et al. 2024). Tool selection and argument generation degrade as tool sets grow and change (Qin et al. 2024). Defences lag behind attacks, including attacks on the tool supply chain (Zhan et al. 2024; Debenedetti et al. 2024; Andriushchenko et al. 2025; He et al. 2025; Hou et al. 2025).

Table 9 Tool-use and external-action mechanisms. Representative systems are illustrative; asterisks mark preprints

Mechanism	How it works	Representative systems or evaluations	Key strength	Key weakness or gap	Safety and permission consideration
Function calling	Structured call to a function	Toolformer (Schick et al. 2023); ToolLLM (Qin et al. 2024)	Extends the action space cheaply	Malformed or hallucinated calls	Each call is a potential external effect
API orchestration	Plans, selects and composes tools	HuggingGPT (Shen et al. 2023); Gorilla (Patil et al. 2024); survey (Qu et al. 2025)	Solves composite tasks	Brittle routing from plan to tool	Effects compound across calls
Standardised tool protocols	Common interface for discovering and invoking tools and data	Model Context Protocol analysis (Hou et al. 2025*)	Interoperability; lower integration cost	Server lifecycle and supply-chain threats	A compromised server affects every connected agent
Browser use	Navigates and manipulates pages	WebArena (Zhou et al. 2024); Mind2Web (Deng et al. 2023); WebShop (Yao et al. 2022)	Access to web tasks	Low success on realistic tasks	Untrusted page content enables injection
Database access	Queries or updates structured stores	AgentBench database environment (Liu et al. 2024)	Structured and checkable	Schema and query errors	Read/write authorisation is critical
Code execution	Runs code as the action	CodeAct (Wang X et al. 2024); SWE-agent (Yang J et al. 2024); SWE-bench (Jimenez et al. 2024)	General and expressive	Large risk surface	Requires sandboxing
Search tools	Retrieves information	Toolformer (Schick et al. 2023); ReAct (Yao et al. 2023b)	Reliable grounding	Result quality varies	Retrieved content can carry injections
Scientific tools	Drives instruments and solvers	Coscientist (Boiko et al. 2023); FunSearch (Romera-Paredes et al. 2024)	Enables discovery workflows	Needs domain oversight	High-stakes physical actions
Embodied actions	Acts through effectors	Voyager (Wang G et al. 2024); WebShop (Yao et al. 2022)	Grounded learning	Mostly simulated; physical robotics out of scope	Effector actions are consequential
Output validation	Checks results before use	CRITIC (Gou et al. 2024); LLM-Modulo (Kambhampati et al. 2024)	Catches errors before action	Under-provided in practice (gap)	Prevents acting on wrong results
Recovery / rollback	Retries or undoes after failure	Reflexion (Shinn et al. 2023); CodeAct (Wang X et al. 2024); τ -bench (Yao et al. 2024*)	Enables recovery	No standard rollback (gap)	Irreversibility must be prevented
Permission / sandbox	Gates and contains actions	ToolEmu (Ruan et al. 2024); TrustAgent (Hua et al. 2024)	Contains unsafe effects	Limited coverage and robustness	Core of unsafe-action prevention

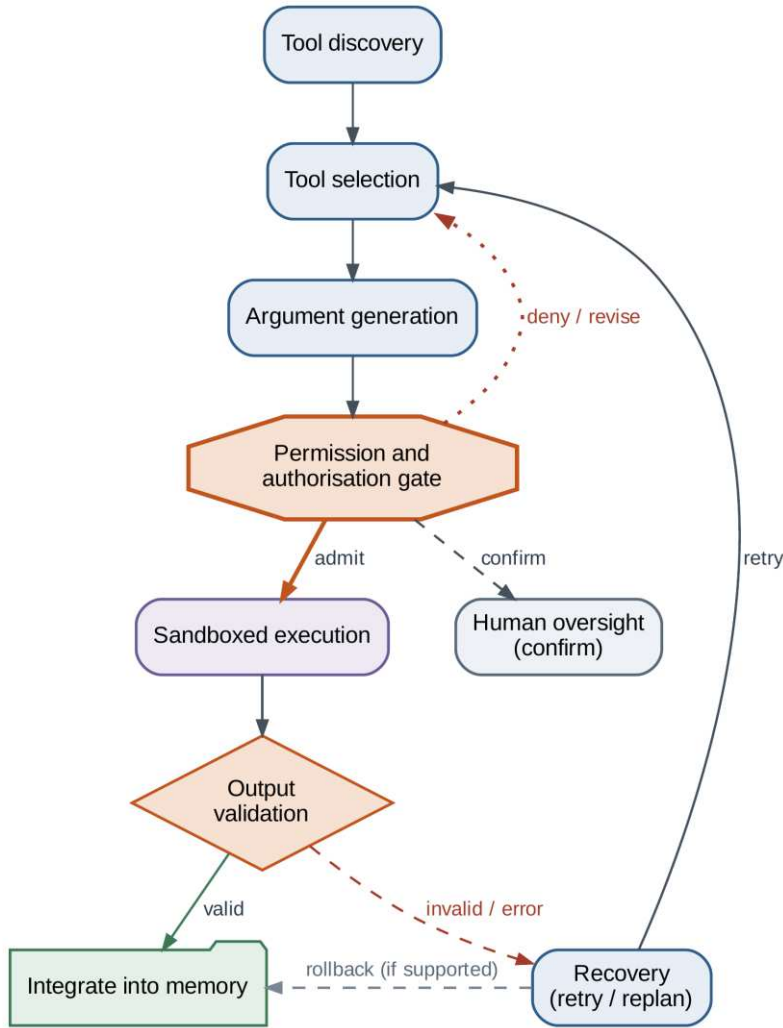


Fig. 9 Tool-use and recovery workflow: tool discovery and selection, argument generation, a permission gate, sandboxed execution and output validation, with explicit paths for human confirmation, retry and (where supported) rollback

6.6 Bridging synthesis

Sections 5 and 6 cover the two capabilities that turn a reasoning core into a situated agent: durable state and consequential action. Both show the same pattern found for planning. The mature machinery is concerned with doing: storing, retrieving, discovering, selecting and calling. The immature machinery is concerned with checking: conflict resolution and staleness detection for memory, output validation and rollback for tools. The documented safety exposures concentrate at these unchecked boundaries (Sect. 8). This convergence is the main empirical motivation for placing verification and a safety-and-authorisation gate on the critical path between a decision and its external effect.

7 Multi-agent coordination

7.1 Why coordinate at all

Multi-agent systems (MAS) divide a task among communicating agents in the expectation that specialisation and interaction will outperform a single, monolithic agent. Whether this expectation holds is contested.

A large-scale failure analysis found that the gains of MAS over single-agent baselines on popular benchmarks are often small. It proposed a taxonomy of fourteen failure modes grouped into three categories: problems of system design, misalignment between agents, and failures of task verification. The taxonomy was validated with strong inter-annotator agreement (Cohen's $\kappa = 0.88$), and the authors argue that many of these failures arise from system design and organisation rather than only from limits of the underlying models (Cemri et al. 2025).

A separate study argues from the data processing inequality that, under a fixed budget of reasoning tokens and with full use of the context, a single agent is more information-efficient than a team. It confirms this experimentally across three model families on multi-hop reasoning, and predicts that multi-agent systems become competitive only when a single agent's effective use of context degrades or when more computation is spent (Tran and Kiela 2026, preprint).

Structure matters as well. When faulty agents are introduced deliberately, the performance of multi-agent teams falls by 5.5% to 23.7% depending on the collaboration structure, with hierarchical structures the most resilient (Huang et al. 2025).

Coordination is therefore a design choice whose benefits depend on conditions. Guo et al. (2024) provide a broader survey of the area.

7.2 Structure and interaction

Most surveyed systems are cooperative. They divide labour among role-playing agents (Li et al. 2023), conversable agents (Wu et al. 2024) or agents following standard operating procedures (Hong et al. 2024; Qian et al. 2024). Adversarial interaction appears mainly as debate (Du et al. 2024). Genuinely competitive or negotiating settings remain little studied in the verified corpus.

Coordination can be centralised or decentralised. Centralised orchestration (Hong et al. 2024) is easier to control but concentrates failure at the orchestrator. Decentralised, conversation-driven patterns (Wu et al. 2024) avoid a single point of failure but are harder to verify. Assigning roles sharpens each agent's focus (Li et al. 2023; Hong et al. 2024; Qian et al. 2024), but every hand-off between roles is a place where information can be lost.

Agents coordinate through communication protocols (Wu et al. 2024; Li et al. 2023) and through shared memory (Hong et al. 2024). Outputs are combined by task allocation, consensus or voting (Wang X et al. 2023; Du et al. 2024). Aggregation cancels errors only when those errors are independent, and shared prompts and shared base models frequently violate that assumption.

7.3 Emergent behaviour and coordination failures

Interaction produces emergent behaviour, some of it desirable (Park et al. 2023; Chen et al. 2024) and some harmful.

Because agents condition on each other's outputs, a single error can spread and be amplified rather than corrected. Cemri et al. (2025) trace this to modes of inter-agent misalignment such as withholding information and derailing the task. When agents share a base model and a prompt, they may reinforce the same mistake, a group-level analogue of hallucination. This helps explain both the small gains of MAS and the single-agent parity result (Cemri et al. 2025; Tran and Kiela 2026).

Coordination also has a price in computation and tokens, which weakens the case for MAS once it is weighed against the benefit (Tran and Kiela 2026; Kapoor et al. 2024).

Several classical pathologies reappear in the documented failures. Breakdowns resembling deadlock show up as premature termination and failure to recognise when a task is finished (Cemri et al. 2025). Collusion is a plausible emergent risk that the verified corpus has not yet studied directly. Attributing responsibility remains unsolved, which complicates both debugging and accountability (Cemri et al. 2025), against a documented governance deficit (Abou Ali et al. 2025).

7.4 Synthesis and gaps in coordination

The literature does not show that more agents reliably produce better outcomes. It shows instead that coordination helps when specialisation is genuine, errors are independent and communication is disciplined, and that it harms otherwise. The open problems are correspondingly specific:

- principled criteria for choosing between a multi-agent system and a single agent under equal computation (Tran and Kiela 2026);
- mechanisms that stop errors and group hallucinations from cascading across hand-offs (Cemri et al. 2025);
- the study of competitive, negotiating and potentially colluding agents;
- governance mechanisms for attributing responsibility (Abou Ali et al. 2025).

These problems carry into the safety analysis, because a multi-agent system multiplies not only capability but also the opportunities for unsafe action.

8 Safety, alignment, robustness, governance and assurance

8.1 The action gap

Safety for an agent differs in kind from safety for a model that only generates text, because an agent acts. Its outputs are actions whose consequences are external and sometimes irreversible. This section sets out the main threats, arranges them in a risk taxonomy (Table 10), presents a defence-in-depth control architecture (Fig. 10), and examines how assurance can remain valid for agents that change after deployment (Table 11). It closes by arguing that conventional model-level safety is necessary but not sufficient once a system can act.

8.2 Threats

Threats to model behaviour. Hallucination becomes consequential once it drives actions; it is mitigated by grounding and external verification rather than by the model's own judgement (Gou et al. 2024; Kambhampati et al. 2024). Goal misalignment arises when an agent optimises the wrong objective. Alignment techniques (Ouyang et al. 2022; Bai et al. 2022) shape behaviour but do not constrain actions. Specification gaming and reward hacking satisfy a proxy instead of the real goal. The evidence for them is indirect but consistent: failures of task verification (Cemri et al. 2025), unreliable self-verification (Huang J et al. 2024; Stechly et al. 2024) and benchmark checks that leaked solutions can satisfy (Kapoor et al. 2024; Aleithan et al. 2024).

Threats through inputs and channels. Prompt injection, and especially indirect prompt injection, hijacks behaviour through content returned by retrieval or tools (Greshake et al. 2023; Zhan et al. 2024; Debenedetti et al. 2024). Jailbreaks manipulate the model into harmful behaviour (Andriushchenko et al. 2025). Backdoors implanted during fine-tuning can manipulate an agent's final output or its intermediate reasoning and actions, and such attacks can take more diverse and covert forms than backdoors in conventional LLMs (Yang W et al. 2024).

Threats through actions and resources. These include tool misuse (Ruan et al. 2024; Andriushchenko et al. 2025), privilege escalation, data leakage, memory poisoning and unsafe autonomous actions, all catalogued in a survey of agent security and privacy (He et al. 2025). Memory poisoning is closely linked to the injection channel (Greshake et al. 2023), and tool supply chains add a further route (Hou et al. 2025).

Organisational threats. Excessive delegation and missing accountability are amplified in multi-agent systems, where responsibility attribution remains unsolved (Cemri et al. 2025) against a

documented governance deficit (Abou Ali et al. 2025). Chan et al. (2023) argue that the harms of increasingly agentic algorithmic systems are often systemic and long-range, which is why they must be anticipated before deployment rather than discovered afterwards.

8.3 A risk taxonomy from threat to residual risk

To make these threats actionable, we trace each one along a seven-stage chain: threat, vulnerability, agent behaviour, consequence, detection, mitigation and residual risk. Table 10 is organised around this chain. Making residual risk explicit is the most important discipline in the taxonomy. Every current mitigation leaves some risk unaddressed, and responsible deployment requires stating what that risk is.

Table 10 Agentic-AI risk taxonomy along the chain threat → vulnerability → agent behaviour → consequence → detection → mitigation → residual risk. Asterisks mark preprints

Threat	Vulnerability	Agent behaviour	Consequence	Detection	Mitigation	Residual risk
Indirect prompt injection (Greshake et al. 2023)	Trust in retrieved or tool-returned content	Follows injected instructions	Hijacked actions; data exfiltration	Injection benchmarks (Zhan et al. 2024; Debenedetti et al. 2024)	Input isolation; provenance; action gating	No general robust defence (He et al. 2025)
Jailbreak / model manipulation (Andriushchenko et al. 2025)	Refusal training can be circumvented	Acts on harmful requests	Harmful external actions	Harm benchmarks (Andriushchenko et al. 2025)	Alignment (Ouyang et al. 2022; Bai et al. 2022*); policy engine	Alignment remains bypassable
Tool misuse (Ruan et al. 2024)	Broad tool access	Invokes tools for unintended ends	Real-world harm; resource abuse	Sandbox risk emulation (Ruan et al. 2024)	Least privilege; sandboxing	Incomplete permission coverage
Tool supply chain (Hou et al. 2025*)	Unvetted or compromised tool servers	Calls a malicious or altered tool	Data leakage; unintended actions at scale	Server provenance and audit	Vetted registries; signed tools; scoped credentials	Protocol-level security immature
Memory poisoning (He et al. 2025)	Writable, unverified memory	Acts on poisoned data	Persistently corrupted behaviour	Provenance and anomaly checks	Verified writes; curation	Detection of subtle poisoning immature
Data leakage / privacy (He et al. 2025)	Retained user data	Exposes sensitive data	Privacy violation	Log audits; monitors	Redaction; access control; retention limits	Private memory at an early stage
Goal misalignment / specification gaming (Huang J et al. 2024; Cemri et al. 2025)	Flawed success signal	Optimises a proxy; false progress	Confidently wrong outcomes	External verification (Kambhampati et al. 2024; Gou et al. 2024)	Verifier in the loop; grounded checks	Self-verification unreliable (Stechly et al. 2024*)
Unsafe autonomous action (Sumers et al. 2024)	Ungated transition from internal to external action	Executes an irreversible action	Irreversible harm	Pre-action verification; monitors (Chan et al. 2024)	Authorisation gate; human confirmation (Hua et al. 2024)	No standard rollback or shutdown
Cascading MAS failure / over-delegation (Cemri et al. 2025)	Uncontrolled hand-offs	Propagates and amplifies errors	Compounded failure	Trace analysis (Cemri et al. 2025)	Orchestration limits; hand-off verification	Attribution unsolved (Abou Ali et al. 2025)
Assurance decay after deployment (Khadka and Maple 2026)	Evidence collected before memory, tool, model or policy changes	Behaviour drifts from the assured baseline	Safety case silently invalid	Re-assurance triggers (Table 11)	Lifecycle-valid assurance case; named stop/go authority	Agent-specific thresholds not standardised

8.4 Controls: defence in depth

The mitigations fit together into a layered architecture (Fig. 10). Guardrails and policy engines encode which actions are permitted (Hua et al. 2024) and go beyond model-level alignment (Ouyang et al. 2022; Bai et al. 2022). Runtime verification checks plans and calls before they take effect (Kambhampati et al. 2024; Gou et al. 2024). Access control enforces least privilege (He et al. 2025; Hua et al. 2024). Sandboxing contains effects (Ruan et al. 2024; Wang X et al. 2024).

Monitoring and logging record actions for audit, although logs are themselves a privacy liability if mishandled (He et al. 2025). Chan et al. (2024) group the measures that make deployed agents visible into three families: agent identifiers, real-time monitoring and activity logging. Each family involves a trade-off between how intrusive and how informative it is.

Human approval checkpoints place a person in the loop for high-stakes or low-confidence actions (Wu et al. 2024; Hua et al. 2024). They are needed because agents cannot reliably judge their own risk (Huang J et al. 2024). Fail-safe shutdown stops an agent that crosses a safety boundary. The verified corpus does not standardise this capability, and we present it both as a required control and as an open engineering problem.

Multi-agent settings also require governance: attribution of responsibility, verification at hand-offs and limits on orchestration (Cemri et al. 2025; Abou Ali et al. 2025).

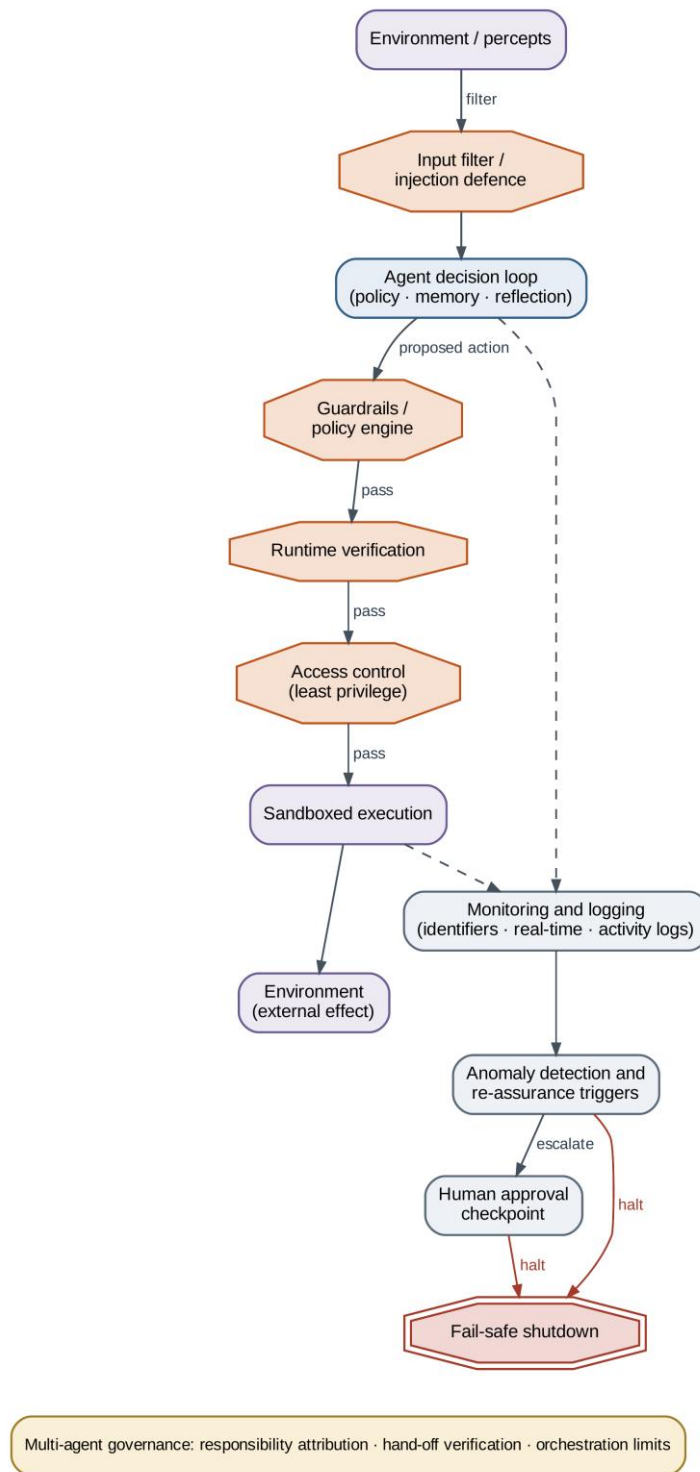


Fig. 10 Defence-in-depth safety architecture around the agent decision loop, from input filtering through guardrails, runtime verification, access control and sandboxing, together with monitoring, re-assurance triggers, human approval and fail-safe shutdown, and a multi-agent governance layer

8.5 Assurance under change

Controls applied at design time are not enough for systems that keep changing. A recent review in this journal, published after our cut-off and cited here as context (Sect. 2.1), argues that AI

assurance should be understood as "a maintained argument under change rather than a one-off statement of compliance" (Khadka and Maple 2026). It evaluates frameworks against five operational commitments: testable claims, evidence obligations, lifecycle validity, accountability and recourse, and interoperability. Agentic systems test the lifecycle-validity commitment in a particularly sharp way. A conventional model changes mainly when it is retrained. An agent's behaviour changes whenever it writes to memory, gains or loses a tool, receives a new system prompt or policy, is moved onto a different base model, or meets a new environment. Evidence of safety collected before deployment can therefore become obsolete without any change to the model's weights.

Existing governance instruments point in this direction but stop short of agent-specific practice. The NIST AI Risk Management Framework organises risk management into four functions (govern, map, measure and manage) and treats measurement as continuing rather than one-off (Tabassi 2023). For high-risk systems, the European Union's AI Act requires a risk-management system, automatic event logging, human oversight and post-market monitoring (European Parliament and Council of the European Union 2024). Singapore's Model AI Governance Framework for Agentic AI is, to our knowledge, the first regulator-issued framework written specifically for agents. It provides guidance for managing the risks of increasingly autonomous AI agents and stresses that human actors remain accountable for their deployment and operation (Infocomm Media Development Authority 2026). Because such frameworks are largely voluntary or principle-level, they do not specify the concrete conditions under which an agent's assurance case lapses and must be reopened (Khadka and Maple 2026).

Table 11 fills this gap. For each type of change that is characteristic of agents, it gives an example monitoring signal and an illustrative re-assurance trigger, and it links each trigger to the stage of the reference framework (Fig. 5) that must be re-verified. It adapts the lifecycle-trigger approach of Khadka and Maple (2026) to the agentic setting, using the failure evidence reviewed in Sects. 4–7. The thresholds are illustrative rather than universal. They should be set in advance for each domain and risk level, and each should be tied to a named person with authority to stop or roll back the agent.

Table 11 Illustrative re-assurance triggers for agentic systems that change after deployment. Adapted to the agentic setting from the lifecycle-trigger approach of Khadka and Maple (2026); thresholds are illustrative and must be set per domain and risk level

Change condition	Example monitoring signal	Illustrative re-assurance trigger	Framework stage to re-verify (Fig. 5)	Supporting evidence
Base-model or version update	Success, pass ^k and harm scores on a frozen regression suite	Any drop beyond the approved tolerance, or any new failure on safety-critical cases	Policy; verification; safety gate	Yao et al. 2024*; Andriushchenko et al. 2025
System prompt, policy or constitution change	Refusal rate; policy-violation rate on red-team prompts	Policy-violation rate rises, or refusals of legitimate tasks rise above tolerance	Safety and authorisation gate	Hua et al. 2024; Bai et al. 2022*
Tool added, removed or updated (including tool-protocol servers)	Tool-call error rate; newly reachable action types; server provenance	New action class with external effect, unverified server, or permission scope widened	Safety gate; external action	Ruan et al. 2024; Hou et al. 2025*
Memory growth or consolidation	Contradiction rate; staleness; share of retrieved items without provenance	Contradictions or unprovenanced writes exceed tolerance; poisoning indicator fires	Memory read/write; reflection	Zhang et al. 2024*; He et al. 2025
Change in environment or data sources	Injection-detection hits; distribution shift in observations	New untrusted content channel, or injection hits above baseline	Perception; safety gate	Greshake et al. 2023; Debenedetti et al. 2024
Change in team composition (multi-agent)	Hand-off failure rate; termination errors; token cost	Failure-mode frequency rises, or cost rises without a gain in success	Coordination; verification	Cemri et al. 2025; Tran and Kiela 2026*
Operational or governance event	Human-intervention rate; incidents; complaints or appeals	Intervention rate or incidents exceed tolerance; any irreversible-action incident	All stages; stop/go decision	Chan et al. 2024; Khadka and Maple 2026

8.6 Why conventional LLM safety is insufficient for systems that act

Conventional LLM safety, meaning alignment training, refusal behaviour and output filtering, is necessary. We argue that, once a system can act, it is insufficient on its own, for five reasons. The first three are supported by direct empirical evidence; the last two rest on analysis of system design and governance (claims C5 and C8 in Sect. 11).

First, alignment governs outputs, not consequences (Ouyang et al. 2022; Bai et al. 2022). Harm arises from what an agent does, so safety has to be enforced at the boundary between internal and external actions (Sumers et al. 2024), which output-level alignment does not reach.

Second, the threat channel bypasses the model's judgement. Indirect injection delivers malicious instructions as data the agent is supposed to act on (Greshake et al. 2023; Zhan et al. 2024; Debenedetti et al. 2024). Refusal training aimed at overtly harmful requests does not catch it.

Third, the model cannot police itself. Intrinsic self-correction and self-verification are unreliable (Huang J et al. 2024; Stechly et al. 2024). Safety that depends on the model judging its own actions inherits this weakness, so independent external verification is required (Kambhampati et al. 2024; Gou et al. 2024).

Fourth, harm compounds across long horizons and across agents. Reviewing responses one at a time does not address errors that accumulate over long tasks. Nor does it address multi-agent failure modes that arise from system design and are therefore unlikely to disappear with stronger base models alone (Cemri et al. 2025; Tran and Kiela 2026).

Fifth, accountability and control are missing at the level of the system. There is a governance deficit (Abou Ali et al. 2025), responsibility attribution is unsolved (Cemri et al. 2025), and assurance lapses when the agent changes (Khadka and Maple 2026). Together with the absence of a standard fail-safe shutdown, this leaves incomplete the means to attribute, halt and reverse harmful actions.

These arguments point to safety that is architectural and enforced at the level of actions: a defence-in-depth stack maintained across the lifecycle, with model alignment as the essential first layer but not the last. How much each additional layer reduces harm in deployed agents has not yet been measured, and this is itself a research priority (Sect. 12).

9 Evaluation frameworks and benchmarks

9.1 What should be measured

Evaluating an agent is harder than evaluating a model. An agent is judged on its behaviour over many steps in an interactive environment, and that behaviour comes with a cost, a level of reliability and some exposure to attack. Current benchmarks capture part of this well and much of it poorly (Sects. 9.2 and 9.3; Fig. 11).

Task success and goal completion are the headline metrics. They are reported as functional success on web tasks (Zhou et al. 2024), the percentage of real software issues resolved (Jimenez et al. 2024) and success rates per environment (Liu et al. 2024). Planning accuracy is measured on formal domains (Valmeekam et al. 2023; Kambhampati et al. 2024). Tool selection and execution are measured with tool suites (Qin et al. 2024; Patil et al. 2024).

Memory retrieval is measured on long-conversation benchmarks (Xu et al. 2025; Chhikara et al. 2025), but measures of memory consistency remain underdeveloped (Zhang et al. 2024). Adaptability is exercised through reflective retries (Shinn et al. 2023). Robustness is tested through perturbation (Verma et al. 2024) and adversarial inputs (Debenedetti et al. 2024), and safety through adversarial suites (Sect. 9.4). Interpretability is rarely evaluated with a dedicated metric.

Latency, token usage and cost are increasingly reported for memory systems (Chhikara et al. 2025; Xu et al. 2025). They are still frequently omitted overall, which distorts comparisons (Kapoor et al. 2024). The rate of human intervention is central to deployment but is seldom reported, even though frameworks support human-in-the-loop operation (Wu et al. 2024).

Newer benchmarks move towards consequential work. TheAgentCompany simulates a small software firm with internal websites and data; its most competitive baseline agent completed 30% of tasks autonomously, and the more difficult long-horizon tasks were still out of reach (Xu et al. 2024, preprint). Long-horizon performance falls sharply (Xie et al. 2024), and reliability across repeated trials declines (Yao et al. 2024). Horizon-based metrics such as the 50% task-completion time horizon express progress in terms of how long the equivalent task takes a human (Kwa et al. 2025). Reproducibility remains weak (Kapoor et al. 2024). Two evaluation designs relate directly to the control stages of the framework. Progress-rate metrics record partial advancement through multi-turn tasks rather than binary success, and so expose where along a trajectory an agent fails (Ma et al. 2024). Agent-as-a-judge evaluation uses an agent with tool access to assess other agents' intermediate steps. In one study (a preprint) it outperformed LLM-as-a-judge baselines and was

as reliable as the study's human evaluation baseline (Zhuge et al. 2024). This supports the framework's requirement that verification draw on a signal independent of the generating policy.

9.2 Evaluation as a cross-cutting dimension

Because evaluation is a meta-dimension of the taxonomy (Sect. 3.6), it has to be specified for each capability and for each stage of the framework, not only at the level of whole-task success. Table 12 links each capability and framework stage to four things: the quantity that must be measured to support a reliability claim, representative benchmarks that measure it, how well it is currently covered, and the metric that is missing.

Three patterns follow from the table. Capabilities that end in an externally checkable outcome, such as tool execution, code and web tasks, are measured best. Capabilities whose failures are silent, such as memory consistency, the independence of verification and calibrated deferral, have no standard metric. And the two control stages the framework adds, verification and the safety gate, are evaluated almost only by adversarial safety suites, and in the benchmarks reviewed here they are not evaluated inside the capability benchmarks on which systems are compared. These gaps feed directly into the claim grading in Sect. 11 and the roadmap in Sect. 12.

Table 12 Evaluation as a cross-cutting dimension: for each capability and framework stage, what must be measured to support a reliability claim, representative benchmarks, current coverage and the missing metric. Asterisks mark preprints

Capability or stage	What must be measured	Representative benchmarks or studies	Coverage	Missing metric
Planning	Plan validity; long-horizon success; repeated-trial reliability	PlanBench; OSWorld; WebArena; τ -bench*	Well covered for validity; weak for horizon	Success as a function of horizon with cost reported
Memory	Retrieval accuracy; consistency; staleness; poisoning robustness	Long-conversation evaluations in A-MEM and Mem0*	Retrieval partial; consistency absent	Contradiction and staleness rates
Tool use	Selection and argument accuracy; execution success; output validation	ToolBench; AgentBench; SWE-bench	Well covered	Validation catch-rate before use
Coordination	Gain over a single agent at equal compute; hand-off failures	Failure taxonomy (Cemri et al. 2025); Tran and Kiela 2026*	Weak	Compute-matched MAS versus single-agent delta
Reflection and verification	Correction gain with versus without external signal; verifier independence	Huang J et al. 2024; CRITIC; LLM-Modulo	Partial (reasoning tasks only)	Verifier fault-detection rate in agent loops
Safety gate	Attack success; utility under attack; harmful-action rate	InjecAgent; AgentDojo; AgentHarm; ToolEmu	Covered, but only in separate suites	Safety reported inside capability benchmarks
Human oversight	Intervention rate; deferral calibration	Frameworks support it (AutoGen), but it is rarely measured	Absent	Intervention rate and calibration error
Lifecycle assurance	Regression after memory, tool, model or policy change	None identified for agents	Absent	Re-assurance trigger coverage (Table 11)

9.3 Comparing the major benchmarks

Table 13 compares the principal benchmarks, and three patterns stand out.

First, coverage of capabilities is uneven. Web and software tasks have well-established metrics, whereas memory consistency, interpretability and the rate of human intervention have no standard benchmarks.

Second, metrics differ from benchmark to benchmark, so comparisons across benchmarks are unsafe.

Third, safety testing forms a separate adversarial family instead of being built into capability testing. A system can therefore score well on capability without ever being tested for safety.

Table 13 Agentic-AI benchmarks and evaluation metrics. "Agent access" denotes what the agent may observe or act on; "Safety testing" denotes whether adversarial or unsafe behaviour is evaluated. Asterisks mark preprints

Benchmark	Evaluated capability	Environment	Task type	Ground truth	Metrics	Agent access	Safety testing	Main limitation
AgentBench (Liu et al. 2024)	Broad capability (8 environments)	Interactive, multi-domain	OS, database, knowledge graph, games, web	Task-specific checkers	Per-environment and overall score	Environment-specific	No	Heterogeneous metrics; 2023-era models
WebArena (Zhou et al. 2024)	Web task completion	Self-hosted realistic sites	E-commerce, forums, development, CMS	Functional correctness	Success rate (best GPT-4 agent 14.41%)	Browser state	No	Fixed sites; English-centric
Mind2Web (Deng et al. 2023)	Generalist web navigation	137 real sites (offline)	Action prediction	Human-annotated actions	Step and action accuracy	Offline snapshots	No	Static offline traces
OSWorld (Xie et al. 2024)	Open-ended computer use	Real OS and applications	Multi-application tasks	Execution checks	Success rate	Real desktop control	No	Setup cost; long-horizon collapse
SWE-bench (Jimenez et al. 2024)	Software issue resolution	Real GitHub repositories	Bug and feature resolution	Repository tests	% resolved	Repository and code tools	No	Leakage documented (Aleithan et al. 2024*)
TheAgentCompany (Xu et al. 2024*)	Consequential workplace tasks	Simulated software company	Browsing, coding, communication	Checkpoint-based evaluators	Full and partial completion (best 30%)	Internal sites, code, chat	No	Single simulated organisation
ToolLLM / ToolBench (Qin et al. 2024)	Tool use over 16,000+ APIs	API sandbox	Multi-tool completion	Reference or win rate	Pass and win rate	API interface	No	API drift; realism
PlanBench (Valmeekam et al. 2023)	Formal planning	PDDL domains	Plan generation and verification	Sound validator	Plan validity	Symbolic specification	No	Classical-domain focus
τ -bench (Yao et al. 2024*)	Tool-agent-user interaction	Domain simulators	Multi-turn tool tasks	Policy and state checks	pass ^k	Tools and simulated user	Partial	Two domains
WebShop (Yao et al. 2022)	Grounded web shopping	Simulated shop	Search and purchase	Reward against target	Task reward and success	Product pages	No	Simulated environment
ToolEmu (Ruan et al. 2024)	Tool-use risk	LM-emulated sandbox	Risky tool scenarios	LM safety evaluator	Risk and failure rate	Emulated tools	Yes (primary)	Emulation fidelity
InjecAgent (Zhan et al. 2024)	Injection robustness	Tool-agent settings	Injection attacks	Attack labels	Attack success rate	Tools and injected content	Yes (primary)	Static attack set
AgentDojo (Debenedetti et al. 2024)	Attack and defence	Dynamic tool environment	Utility versus attack	Utility and attack checks	Utility and attack success	Tools and adversarial content	Yes (primary)	Deployment coverage
AgentHarm (Andriushchenko et al. 2025)	Harmfulness	Tool-agent settings	Harmful requests	Harm and refusal labels	Harm score; refusal	Tool-using agent	Yes (primary)	Scope of harm taxonomy

9.4 Systemic weaknesses

Contamination and leakage inflate apparent success, even on benchmarks scored by execution (Aleithan et al. 2024, preprint). Many results are hard to reproduce and are reported without cost controls (Kapoor et al. 2024).

Environments trade realism against control. Offline traces test action prediction against fixed page snapshots (Deng et al. 2023), and simulated or self-hosted sites diverge from the open web (Yao et al. 2022; Zhou et al. 2024).

Safety evaluation is weak and kept separate: capability benchmarks are largely untested for safety (Ruan et al. 2024; Zhan et al. 2024; Debenedetti et al. 2024; Andriushchenko et al. 2025).

Inconsistent metrics and poor reproducibility undermine comparison. Reliability-aware metrics such as pass^k (Yao et al. 2024) and time-horizon measures (Kwa et al. 2025) are steps towards reporting what deployment actually requires.

9.5 Synthesis for evaluation

Current benchmarks establish, with reasonable confidence, two things: agents are far from reliable on realistic long-horizon tasks, and they are broadly vulnerable under adversarial testing. They establish much less reliably how capable any particular system is, because contamination, artefacts of the environment, inconsistent metrics and missing cost and reliability reporting all bias comparisons.

A defensible evaluation would report capability, cost, reliability over repeated trials and robustness under attack together, on interactive tasks that resist contamination. It would treat success in a single run on a static and possibly contaminated benchmark as weak evidence. That standard has not yet been reached. This review therefore uses reported figures as contextual evidence and does not treat them as a leaderboard.



Fig. 11 Multi-dimensional evaluation framework for agentic systems, showing for each dimension whether current benchmarks measure it well (filled circle), weakly or partially (diamond) or not in a standardised way (open circle). Safety dimensions are measured only by a separate adversarial benchmark family, not by capability benchmarks

10 Application domains

10.1 Readiness criteria

Claims that agents are "ready" for a domain are rarely backed by explicit criteria. We therefore classify each domain by the strongest type of evidence in the corpus, using four criteria:

- E1, task realism: agent behaviour is evaluated on real tasks of the target domain, or on a benchmark built from them.
- E2, independent evaluation: the evidence comes from a public benchmark or has been replicated by others, not only from the system's authors.
- E3, safety validation: safety or oversight has been evaluated specifically for the domain.
- E4, operational use: use in practice is documented, with human oversight of consequential actions.

Four readiness levels follow from these criteria (Table 14):

- R0: no domain-specific evidence of the target autonomous behaviour.
- R1: laboratory demonstration (E1, in a research setting, without E2).
- R2: benchmark evidence (E1 and E2, without E3 or E4).
- R3: supervised deployment (E1–E4).

A domain is assigned the highest level whose criteria are all met by at least one Grade A or B source in the corpus (Sect. 2.5). Readiness levels describe the type of evidence, not how good performance is. A domain can be at R2 with low success rates, and the table reports performance separately.

Table 14 Readiness levels and the evidence criteria that define them. E1 task realism; E2 independent evaluation; E3 domain safety validation; E4 documented operational use with human oversight. A domain is assigned the highest level whose criteria are met by at least one Grade A or B source

Level	E1	E2	E3	E4	Interpretation
R0 No domain evidence	No	No	No	No	No evidence of the target autonomous behaviour in the corpus
R1 Laboratory demonstration	Yes	No	No	No	Shown by the system's authors in a research setting
R2 Benchmark evidence	Yes	Yes	No	No	Independent or public benchmark evidence; no safety validation or deployment
R3 Supervised deployment	Yes	Yes	Yes	Yes	Documented use with human oversight of consequential actions

Applying these criteria tempers the narrative of readiness. No domain reaches R3 in the corpus. The most frequently cited successes are at R1 or R2, and every high-stakes domain is at R0 (Table 15).

10.2 Domains at R1 and R2

Scientific discovery (R1). Agents have planned and executed chemical experiments (Boiko et al. 2023) and have found new mathematical constructions by combining program search with an evaluator (Romera-Paredes et al. 2024). These are genuine laboratory demonstrations carried out under expert supervision, not autonomous practice with validated safety. Autonomous experimentation also carries physical risks that call for oversight and containment (cf. Ruan et al. 2024).

Software engineering (R2). This is the best-evidenced application (Jimenez et al. 2024; Yang J et al. 2024; Hong et al. 2024; Qian et al. 2024). Even here, success on realistic issues is modest and the flagship benchmark is affected by contamination (Aleithan et al. 2024, preprint). The corpus contains no domain-specific safety validation (E3) or documented operational use (E4). The accurate classification is therefore benchmark evidence (R2).

Embodied systems (R1, simulated). These have been demonstrated in simulated worlds (Wang G et al. 2024; Yao et al. 2022), which does not establish competence in the physical world.

Business process automation (R2). Web and computer-use agents target this area, but the low success rate on realistic tasks (14.41%; Zhou et al. 2024), the partial success in a simulated workplace (30%; Xu et al. 2024) and the degradation over long horizons (Xie et al. 2024) show that reliable autonomous automation has not yet been demonstrated.

Finance (R0). The finance work in the corpus consists of simulations of macroeconomic behaviour (Li et al. 2024), not autonomous financial decision-making, so criterion E1 is not met for the target behaviour.

10.3 High-stakes domains at R0

For the domains below, the verified corpus contains no dedicated, safety-validated evidence of deployment. The analysis therefore rests on general limits of capability (Kambhampati et al. 2024; Huang J et al. 2024) and on documented risks (He et al. 2025; Greshake et al. 2023; Andriushchenko et al. 2025).

Healthcare (R0). Healthcare is high-stakes and regulated, with acute privacy and safety constraints. The reliability gaps described above argue against autonomous clinical action, and no validated autonomous healthcare agents were found. Appropriate use at present is bounded, supervised assistance with strict data protection (He et al. 2025).

Education (R0). Agents are proposed for tutoring and administration. Personalisation, however, implies retaining data about minors (He et al. 2025), and the limits on reliability argue against unsupervised assessment. No validated deployment was found.

Cybersecurity (R0). The field is dual-use. Harm benchmarks show that jailbreaks can drive agents to harmful behaviour (Andriushchenko et al. 2025), and the security survey maps a wide threat surface (He et al. 2025). The evidence therefore documents the risk of misuse more clearly than it documents validated defensive deployment.

Physical robotics (R0). This lies largely outside the verified corpus, since embodied results are simulated (Wang G et al. 2024). Because physical actions are often irreversible, autonomous deployment would be premature.

Public services (R0). Deployment raises obligations of accountability and fairness, against a documented governance deficit (Abou Ali et al. 2025), privacy constraints (He et al. 2025) and emerging obligations for oversight and post-market monitoring (European Parliament and Council

of the European Union 2024). No validated deployment was found, and the barriers are as much legal and institutional as technical.

Table 15 Application domains, risks, required safeguards and readiness level (Sect. 10.1). Requirements and risks are inferred from the evidence cited in the text; the table does not assert deployment beyond it. Asterisks mark preprints

Domain	Expected agent behaviour	Technical requirements	Risk level	Required human oversight	Data and privacy constraints	Validation requirements	Deployment barriers	Readiness level (Table 14)
Scientific discovery	Plan and run experiments; search for solutions	Reliable tools; external verification; containment	High	High; expert in the loop	Research-data governance	Independent reproduction; safety review	Physical-world risk	R1 (Boiko et al. 2023; Romera-Paredes et al. 2024)
Software engineering	Resolve issues; write and test code	Code execution; repository tools; verification	Medium–high	Human code review	Source-code and IP confidentiality	Contamination-resistant evaluation	Modest success; leakage (Aleithan et al. 2024*)	R2; resolution rates modest (Jimenez et al. 2024; Yang J et al. 2024)
Embodied systems	Perceive and act in an environment	Grounded action; skill memory	Medium (simulated) to high (physical)	High	Environment and sensor data	Sim-to-real validation	Physical grounding gap	R1, simulated only (Wang G et al. 2024)
Business process automation	Execute digital workflows	Browser and OS control; long-horizon reliability	Medium–high	Human approval of actions	Enterprise data privacy	Reliability and cost reporting	Low success; long-horizon collapse	R2; success 14–30% (Zhou et al. 2024; Xie et al. 2024; Xu et al. 2024*)
Finance	(Currently) simulate economic behaviour	Auditability; reliability; controls	High	High; regulated sign-off	Financial-data regulation	Auditable, reproducible validation	Regulation; reliability gap	R0 (macroeconomic simulation only; Li et al. 2024)
Healthcare	Assist clinicians; manage information	Very high reliability; privacy; verification	Very high	Maximal; clinician in the loop	Patient-data protection	Clinical validation; regulatory approval	Reliability; regulation; liability	R0
Education	Tutor; support administration	Reliable content; private personalisation	High (minors)	High	Protection of student and minor data	Pedagogical and safety validation	Reliability; privacy of minors	R0
Cybersecurity	Assist defence (dual use)	Robustness; strict permissions	Very high (dual use)	High; strict governance	Sensitive security data	Adversarial safety testing	Misuse risk (Andriushchenko et al. 2025)	R0 (misuse risk documented)
Robotics (physical)	Control physical effectors	Real-time control; safety interlocks	Very high (irreversible)	High; hardware interlocks	Sensor and operational data	Physical safety certification	Sim-to-real gap; hardware safety	R0
Public services	Support administration and decisions	Fairness; accountability; auditability	High	High; accountable human authority	Citizen-data protection	Fairness audit; legal review; post-market monitoring	Governance deficit (Abou Ali et al. 2025); legal	R0

10.4 Synthesis for applications

Two conclusions follow, and both counsel restraint.

First, in the evidence reviewed, readiness is lowest where stakes are highest (claim C7, moderate confidence). The domains with the strongest evidence (software engineering, scientific discovery and simulated embodiment) tolerate supervision and allow actions to be reversed. The highest-stakes domains have no domain-specific agent evidence at all. This conclusion is bounded by the corpus, which centres on computer-science venues. Domain literatures, clinical studies for example, were not searched systematically and may contain evidence at R1 or R2.

Second, the binding constraints are the ones traced throughout this review:

- unreliable long-horizon planning (C1);
- unreliable intrinsic self-correction (C2);
- immature evaluation (C4);
- a gap in safety, governance and assurance at the level of actions (C5 and C8).

Presenting current demonstrations as ready for deployment would misrepresent the evidence and, in high-stakes domains, could cause harm.

11 Critical synthesis: tensions and a graded claim–evidence map

11.1 Four tensions and their moderators

Sections 4 to 10 each set positive results against negative ones. Read together, the apparent contradictions reduce to four tensions. In each case the conflict disappears once a moderating condition is made explicit. This is the review's main analytical result, as distinct from description.

Tension 1: self-correction works and fails. Iterative self-feedback and verbal reflection report gains (Madaan et al. 2023; Shinn et al. 2023). Intrinsic self-correction, however, often fails to improve reasoning or makes it worse (Huang J et al. 2024; Stechly et al. 2024). The moderator is the source of the feedback. Gains appear when the correction signal comes from outside the generating model, for example tests, tools, environment rewards or oracle labels (Gou et al. 2024; Kambhampati et al. 2024). They are unreliable when the model grades itself.

Tension 2: multi-agent systems help and do not help. Debate and role-structured teams report improvements (Du et al. 2024; Hong et al. 2024; Qian et al. 2024). Failure analyses find small gains and recurring breakdowns in coordination (Cemri et al. 2025), and single agents match teams when the budget of reasoning tokens is equal (Tran and Kiela 2026). Two moderators

reconcile these findings. One is whether the comparison holds computation constant. The other is whether the agents' errors are independent, which shared base models and prompts undermine.

Tension 3: planning improves and remains unreliable. Search, reasoning models and hybrid planners raise scores substantially (Yao et al. 2023a; Guo et al. 2025; Liu et al. 2023). Autonomous plan generation nevertheless fails on formal planning tasks (Kambhampati et al. 2024; Valmeekam et al. 2023), and long-horizon success stays low (Xie et al. 2024; Zhou et al. 2024). The moderators are task verifiability and horizon length. Gains concentrate where a reliable evaluator exists and the horizon is short.

Tension 4: benchmark progress and deployment readiness diverge. Scores and time horizons are rising (Kwa et al. 2025). Yet contamination, static environments, missing cost reporting and siloed safety testing limit what those scores imply for deployment (Aleithan et al. 2024; Kapoor et al. 2024; Kehkashan et al. 2026). The moderator is the validity of the evaluation: how resistant it is to contamination, how interactive it is, and whether it reports cost and repeated-trial reliability.

11.2 Graded claim–evidence map

Table 16 states the eight headline claims of the review, the evidence for and against each, the moderator that reconciles them, and a confidence grade assigned by an explicit rule:

- High: at least three independent supporting sources of Grade A or B, and any contrary Grade A or B evidence is explained by a stated moderator.
- Moderate: two supporting Grade A or B sources, or a conclusion that is explicitly bounded by the scope of the corpus.
- Low: support rests on preprints (Grade C) or on a single Grade A or B source.

Four claims reach high confidence: C1, C2, C5 and the need-for-controls part of C8. Two are moderate. C3 is supported by two peer-reviewed failure and resilience studies, and C7 is explicitly bounded by the scope of the corpus. Two are low (C4 and C6), because their strongest support is still in preprints. These two should be read as well-motivated hypotheses rather than established findings, and they are where confirmatory studies would add most (Sect. 12). The last column of Table 16 states, for each claim, what evidence would change its grade. This makes the grading falsifiable.

To keep claims and evidence aligned throughout the article, every summary statement that depends on one of these claims carries its claim identifier, in the abstract's findings, in Sect. 10.4 and in Sect. 13. No summary statement is stated more strongly than its grade allows.

Table 16 Graded claim–evidence map. Evidence grades: A peer-reviewed empirical or method work; B peer-reviewed survey, position or governance instrument; C preprint. Confidence rule: High requires at least three independent Grade A or B supporting sources, with any contrary Grade A or B evidence explained by a stated moderator; Moderate requires two such sources or a conclusion explicitly bounded by the corpus; Low applies when support rests on Grade C or on a single A or B source

I D	Claim	Supporting evidence (grade)	Qualifying or contrary evidence (grade)	Moderator	Confidence	What would change the grade
C 1	Autonomous LLM plan generation is unreliable over long horizons without external verification	Valmeekam et al. 2023 (A); Kambhampati et al. 2024 (B); Xie et al. 2024 (A); Zhou et al. 2024 (A); Shojaei et al. 2025 (A)	Yao et al. 2023a (A); Guo et al. 2025 (A)	Task verifiability; horizon length	High	Lower: long-horizon success rising without external verifiers on contamination-resistant tasks
C 2	Intrinsic self-correction is unreliable; correction grounded in an external signal helps	Huang J et al. 2024 (A); Gou et al. 2024 (A); Kambhampati et al. 2024 (B); Stechly et al. 2024 (C)	Madaan et al. 2023 (A); Shinn et al. 2023 (A)	Source of feedback (external or oracle versus intrinsic)	High	Lower: intrinsic self-correction gains replicated without oracle or external signals
C 3	Multi-agent gains are conditional and often small at equal computation	Cemri et al. 2025 (A); Huang et al. 2025 (A); Tran and Kiela 2026 (C); Kapoor et al. 2024 (C)	Du et al. 2024 (A); Hong et al. 2024 (A); Qian et al. 2024 (A); Chen et al. 2024 (A)	Compute matching; error independence; team structure	Moderate	Raise: further peer-reviewed compute-matched comparisons; lower: consistent multi-agent gains at equal compute
C 4	Headline benchmark scores overstate deployable capability	Kehkashan et al. 2026 (B); Aleithan et al. 2024 (C); Kapoor et al. 2024 (C); Yao et al. 2024 (C)	Kwa et al. 2025 (A) (real but bounded progress); Ma et al. 2024 (A) (progress-rate metrics refine rather than refute success rates)	Contamination resistance; interactivity; cost and pass^k reporting	Low	Raise: peer-reviewed replications of contamination or pass^k degradation
C 5	Tool-using agents are broadly vulnerable to injection and misuse that model alignment does not prevent	Greshake et al. 2023 (A); Zhan et al. 2024 (A); Debenedetti et al. 2024 (A); Andriushchenko et al. 2025 (A); Ruan et al. 2024 (A); He et al. 2025 (B); Yang W et al. 2024 (A)	Defences reduce but do not eliminate attacks (Debenedetti et al. 2024)	Presence of action gating and input isolation	High	Lower: defences that reduce attack success to near zero without utility loss
C 6	Agent memory lacks mechanisms for conflict detection and temporal consistency, and is an attack surface	He et al. 2025 (B); Zhang et al. 2024 (C); Chhikara et al. 2025 (C)	Xu et al. 2025 (A) (dynamic organisation, not consistency)	Consolidation and provenance policies	Low	Raise: peer-reviewed measurements of contradiction and staleness in agent memory
C 7	In the corpus, readiness evidence is weakest in the highest-stakes domains	Boiko et al. 2023 (A); Romera-Paredes et al. 2024 (A); Jimenez et al. 2024 (A); Yang J et al. 2024 (A)	Domain literatures not searched systematically	Scope of corpus (computer-science venues)	Moderate (bounded)	Change: domain literatures (e.g. clinical) showing R1–R3 evidence
C 8	Action-level controls and lifecycle assurance are needed beyond model alignment	Hua et al. 2024 (A); Ruan et al. 2024 (A); Chan et al. 2023 (B); Chan et al. 2024 (B); Tabassi 2023 (B); European Parliament and Council of the European Union 2024 (B)	Effectiveness of layered controls in deployment not yet measured	Deployment evidence	High (need); Low (effect size)	Raise effect-size grade: deployment studies measuring harm reduction from layered controls

12 Prioritised research roadmap

12.1 Structure and prioritisation rule

The graded evidence converges on one conclusion. What mainly limits agentic AI at present appears to be the lack of architecture and infrastructure for verification, safety, assurance and accountability, rather than model capability alone. The roadmap is organised by time horizon (Table 17; Fig. 12). Every direction is linked to the claims in Table 16 that document its gap.

To make the prioritisation explicit, each direction receives a score from 0 to 5 made up of three parts:

- S (0 or 1): whether the direction directly reduces the risk of unsafe external action.
- E (0, 1 or 2): the confidence of the linked claim; Low scores 0, Moderate 1 and High 2.
- D (0, 1 or 2): the number of other directions that depend on it in Fig. 12, capped at 2.

Directions scoring 4 or 5 are rated high priority, 2 or 3 medium, and 0 or 1 lower priority. Assigning S and the claim links involves author judgement, and both are shown in Table 17 so that readers can re-score.

Three immediate directions score as high priority: grounded verification, safe tool-grounded decisions, and uncertainty estimation for calibrated deferral. In the medium term, dynamic adaptation, explainability and human–agent collaboration also score as high priority. The roadmap contains exactly eighteen directions (Table 17), and the end-to-end evaluation of the reference framework is not a nineteenth. It is part of the safe tool-grounded decisions direction, whose proposed work includes ablating the verification stage and the safety gate to measure how much each contributes to task success and to avoiding harm. The model of Sect. 3.9 states the predictions that this ablation should test. Several directions receive a lower score, among them continual learning, cross-domain generalisation and cost efficiency. This does not mean they are unimportant. It means that the specific gaps they address are either less directly linked to unsafe action or documented mainly in preprints.

12.2 Immediate directions

Grounded verification. Generation should be separated from verification, and verification should rest on external checks. The reasons are that intrinsic self-correction is unreliable (Huang J et al. 2024; Stechly et al. 2024, preprint), while grounded verification helps (Kambhampati et al. 2024; Gou et al. 2024).

Safe tool-grounded decisions. These require permission gating, sandboxing, provenance tracking and runtime verification. Rollback is not standardised, and agents are broadly susceptible to injection and misuse, including through tool protocols (Greshake et al. 2023; Zhan et al. 2024; Ruan et al. 2024; Andriushchenko et al. 2025; Hou et al. 2025).

Trustworthy memory. Memory needs defined persistence semantics, detection of contradictions and verified writes (Zhang et al. 2024; Chhikara et al. 2025; He et al. 2025).

Uncertainty estimation. Calibrated confidence and the ability to abstain are needed because an agent may not know when it is wrong (Huang J et al. 2024; Stechly et al. 2024, preprint). Calibration is also the basis for handing a decision safely to a human.

Evaluation standards. Evaluations should report capability, cost, reliability over repeated trials and robustness under attack together, on tasks that resist contamination.

Reproducibility. Standard evaluation harnesses and released artefacts are needed (Kapoor et al. 2024; Aleithan et al. 2024, preprint; Yao et al. 2024; Kwa et al. 2025).

12.3 Medium-term directions

Long-horizon autonomy. The available evidence does not show that scaling alone closes the reliability gap over long horizons (Xie et al. 2024; Kambhampati et al. 2024; Yao et al. 2024). Hierarchical architectures with per-step external verification and consistent memory are needed.

Temporal and causal reasoning. Agent memory has a documented weakness in temporal consistency (Zhang et al. 2024). Explicit temporal, causal and world models are needed, building on work that treats reasoning as planning with a world model (Hao et al. 2023).

Dynamic adaptation. Policies that adapt robustly online, and benchmarks that are continually refreshed, are needed to counter brittleness (Verma et al. 2024, preprint; Deng et al. 2023).

Continual learning. Durable memories need to be consolidated into model parameters safely and without catastrophic forgetting (Wang G et al. 2024; Packer et al. 2024).

Explainability. Decision records must be faithful and auditable, because reasoning traces may not reflect the actual basis of a decision (Verma et al. 2024, preprint).

Multi-agent governance. Accountable orchestration and mechanisms for attributing failures are required (Cemri et al. 2025; Abou Ali et al. 2025).

Human-agent collaboration. The level at which an agent defers to a human should be calibrated and should be measured through the rate of intervention (Wu et al. 2024; Hua et al. 2024; Huang J et al. 2024).

Cross-domain generalisation. Capabilities should be validated on held-out domains (Liu et al. 2024; Mialon et al. 2024).

Energy and cost efficiency. Cost-aware methods and efficient memory are required (Yao et al. 2023a; Wang X et al. 2023; Cemri et al. 2025; Chhikara et al. 2025; Xu et al. 2025; Kapoor et al. 2024).

12.4 Long-term directions

Embodied safety. This requires verified control, hardware interlocks and validated transfer from simulation to reality, since current embodied results are simulated (Wang G et al. 2024).

Regulatory accountability. This requires standards for auditability and accountability, and lifecycle-valid assurance cases. They must answer a governance deficit and unsolved attribution (Abou Ali et al. 2025; Cemri et al. 2025; He et al. 2025), and they must translate emerging regulation into agent-specific evidence (European Parliament and Council of the European Union 2024; Infocomm Media Development Authority 2026; Khadka and Maple 2026).

Social and ethical implications. Privacy, dual-use misuse, effects on labour and fairness of access require interdisciplinary governance that no single technical control can provide (He et al. 2025; Abou Ali et al. 2025; Andriushchenko et al. 2025; Chan et al. 2023).

Hybrid versus scaled architectures. Several of these questions share a foundational one: can reliable autonomous planning and self-assessment be reached by scaling current models, or does it require a hybrid architecture that couples generation to external verification? The current evidence (C1 and C2) leans towards the hybrid answer (Kambhampati et al. 2024; Huang J et al. 2024), although no controlled comparison at scale has yet settled the question.

Table 17 Prioritised research roadmap. Linked claims refer to Table 16. Score = S (direct reduction of unsafe-action risk, 0–1) + E (confidence of linked claim: Low 0, Moderate 1, High 2) + D (dependent directions in Fig. 12, capped at 2). Priority: High 4–5; Medium 2–3; Lower 0–1

Tier	Direction	Linked claims	Proposed direction	Evaluation criteria	S	E	D	Score	Priority
Immediate	Grounded verification (reliable self-correction)	C2	Independent verifier on the action path; no self-grading	Correction gain with versus without an external signal; verifier fault-detection rate	1	2	1	4	High
Immediate	Safe tool-grounded decisions	C5, C8	Permission gate, sandbox, provenance, runtime verification; end-to-end ablation of the framework stages	Utility under attack reported alongside capability	1	2	2	5	High
Immediate	Uncertainty estimation for deferral	C2	Calibrated confidence with abstention and deferral	Calibration error; quality of deferral	1	2	2	5	High
Immediate	Trustworthy memory	C6	Persistence semantics; contradiction detection; verified writes	Consistency and staleness rates; robustness to poisoning	1	0	2	3	Medium
Immediate	Evaluation standardisation	C4	Joint capability, cost, reliability and safety reporting on clean tasks	Adoption; pass ^k ; cost-controlled comparison	0	0	2	2	Medium
Immediate	Reproducibility	C4	Standard harnesses; released code, seeds and prompts	Independent replication	0	0	1	1	Lower
Medium	Dynamic adaptation	C5	Robust online adaptation; continually refreshed benchmarks	Distribution-shift and live-environment tests	1	2	1	4	High
Medium	Explainability	C8	Faithful, auditable decision records	Faithfulness metrics; audit exercises	1	2	1	4	High
Medium	Human–agent collaboration	C2	Calibrated deferral and interaction design	Intervention rate; success under oversight	1	2	1	4	High
Medium	Long-horizon autonomy	C1	Hierarchical decomposition with per-step verification and consistent memory	Success as a function of horizon; time horizon; pass ^k	0	2	0	2	Medium
Medium	Multi-agent governance	C3	Accountable orchestration; hand-off verification	Accuracy of failure attribution; compute-matched gains	1	1	1	3	Medium
Medium	Temporal and causal reasoning	C6	Explicit temporal, causal and world models	Temporal-order and causal-intervention tasks	0	0	0	0	Lower
Medium	Continual learning	C6	Memory-to-parameter consolidation with retention guarantees	Retention and interference tests	0	0	0	0	Lower
Medium	Cross-domain generalisation	C4	Validation on held-out domains	Generalist suites with held-out tasks	0	0	0	0	Lower
Medium	Energy and cost efficiency	C3	Cost-aware methods; efficient memory	Cost-controlled comparison	0	1	0	1	Lower
Long-term	Regulatory accountability and assurance	C8	Auditability standards; lifecycle-valid assurance cases with re-assurance triggers	Compliance and audit criteria; trigger coverage	1	2	0	3	Medium
Long-term	Embodied safety	C7	Verified control; interlocks; sim-to-real safety	Physical safety certification	1	1	0	2	Medium
Long-term	Social and ethical implications	C7	Interdisciplinary governance; impact assessment	Societal-impact and audit frameworks	1	1	0	2	Medium

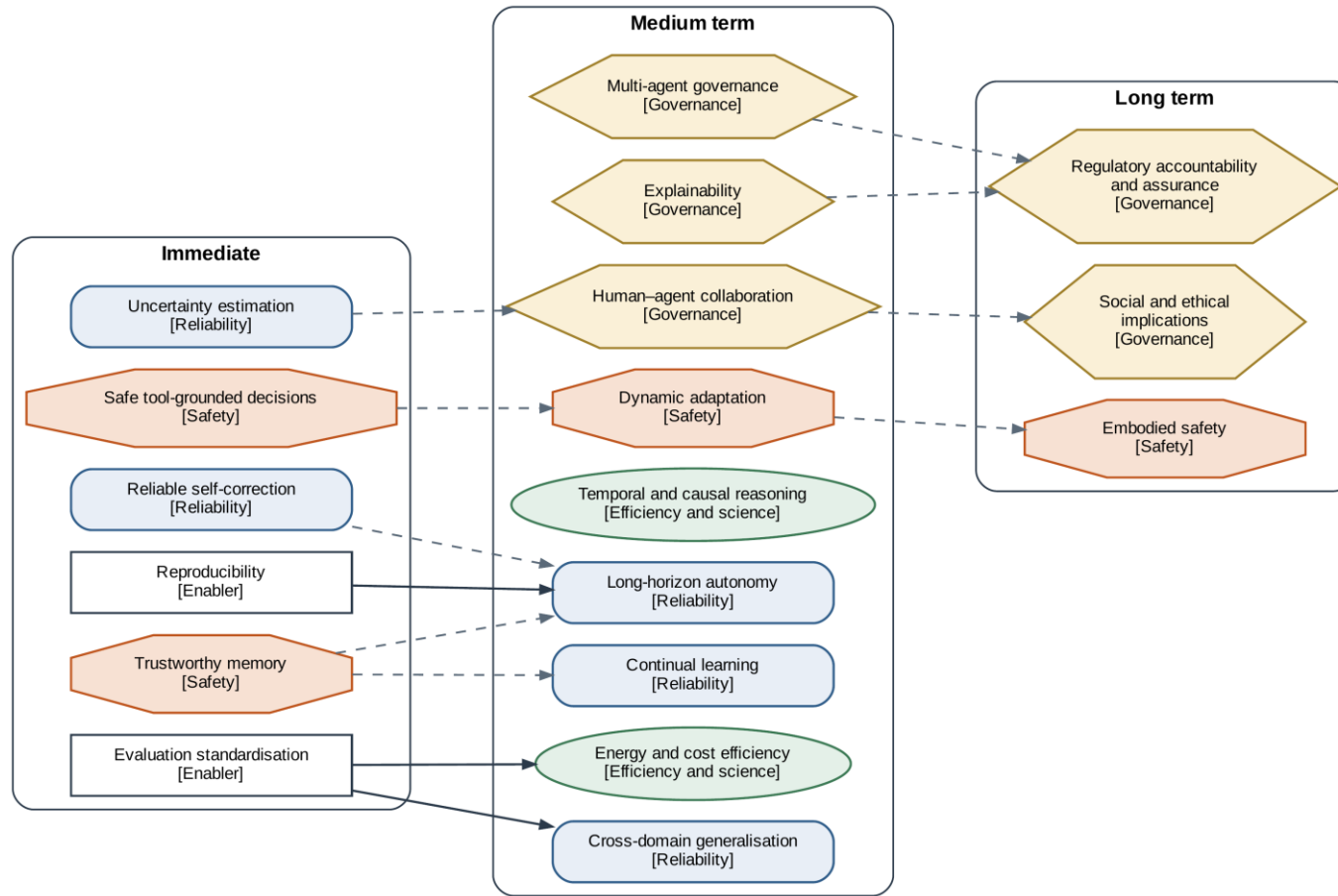


Fig. 12 Research roadmap positioning eighteen directions by time horizon. Node shape and bracketed labels give the thematic cluster (reliability, safety, governance, efficiency and science, enabler). Dashed arrows mark prerequisite dependencies and solid arrows show the influence of the evaluation and reproducibility enablers. The number of dependent directions enters the priority score in Table 17

13 Conclusion

Agentic AI marks a genuine shift from models that generate text to systems that pursue goals over time, remember what they observe, use tools, coordinate with other agents and act in the world. This review has argued that the promise of that shift is real. It has also argued that the present reliability and safety of agentic systems are frequently overstated, and that much of the literature describes systems rather than weighing the evidence on whether they work.

Graded against explicit rules, the evidence supports the following conclusions, with the confidence shown:

- Autonomous plan generation is unreliable over long horizons, and gains concentrate where tasks are short and verifiable (C1, high confidence).
- Self-correction helps when it is grounded in external, verifiable feedback and is unreliable when purely intrinsic (C2, high).
- Tool-using agents are broadly vulnerable to injection and misuse, which model-level alignment does not prevent (C5, high).
- Controls at the level of actions, together with assurance that remains valid over the lifecycle, are needed beyond alignment (C8, high for the need). How effective such controls are in deployment has not yet been measured (low).
- In the corpus reviewed, readiness is lowest where stakes are highest (C7, moderate).
- Multi-agent gains are conditional on compute matching, error independence and team structure (C3, moderate).
- Two further claims are plausible but rest mainly on preprints and remain hypotheses: that headline benchmark scores overstate deployable capability (C4), and that agent memory lacks mechanisms for maintaining consistency (C6).

The unifying lesson is architectural. Current evidence does not indicate that reliability and safety emerge from a larger policy alone. They appear to depend on coupling the policy to external verification and to controls on action, and on keeping those controls valid as the system changes. The formally specified reference framework (Sect. 3.7) encodes this lesson as a testable hypothesis. The retrospective mapping (Sect. 3.8) shows that none of ten representative systems implements both of its control stages in full: across the ten, the checking stages score 0.38 against 0.78 for the doing stages. The reliability model (Sect. 3.9) shows why this gap matters more as horizons lengthen.

On this basis we propose six conditions for agentic AI to become reliable, safe and deployable:

1. Generation is separated from verification, and verification is grounded in a signal independent of the generating model.
2. Every consequential action passes a safety-and-authorisation gate backed by least-privilege permissions and sandboxing. Because rollback is not standardised, the gate relies on prevention rather than on undoing actions.
3. Memory is trustworthy: consistent, aware of conflicts, and protected against poisoning and leakage.
4. The agent estimates its uncertainty well enough to defer to calibrated human oversight, and the rate of human intervention is measured.
5. Evaluation reports capability, cost, reliability over repeated trials and robustness under attack together, on interactive and reproducible tasks that resist contamination, with the hooks for each capability given in Table 12.
6. Governance and assurance remain valid after deployment through attribution of responsibility, auditability, and re-assurance triggers that reopen the safety case whenever memory, tools, policies or models change (Table 11).

These conditions imply a principle of proportionality: the autonomy granted to an agent should not exceed the level at which its reliability and safety have been demonstrated, measured on the readiness scale of Sect. 10. The highest-stakes domains, currently at R0, should therefore be the last to receive autonomous deployment, not the first. Agentic AI for complex decision-making is best understood today as a promising capability that still awaits the infrastructure for verification, safety and assurance that would make it trustworthy. The prioritised roadmap in Sect. 12 sets out where building that infrastructure should begin.

Statements and Declarations

Funding

The authors did not receive support from any organisation for the submitted work.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Author contributions

Mohan Babu A: conceptualisation, literature search and screening, formal analysis, , writing – original draft. Faranak Nejatia: visualisation, Draft Quality and Validation. Madhan E S:

conceptualisation, methodology, supervision, writing – review and editing. Both authors read and approved the final manuscript.

Ethics approval

Not applicable. This article reviews published literature and involves no human participants, human data or animals.

Consent to participate and consent for publication

Not applicable.

Data availability

No new primary data were generated. The review protocol, including the PRISMA 2020 item mapping, the full and broadened search strings, the search audit, eligibility criteria, grading rules and coding scheme, is provided as Online Resource 1. The coded evidence-extraction matrix for all 98 corpus sources, with the post-cut-off context source flagged separately, is provided as Online Resource 2 and underlies Figs. 2 and 3.

Use of generative AI

The use of an LLM-based assistant during manuscript preparation is documented in Sect. 2.9.

Supplementary Information

Online Resource 1 (ESM_1.pdf): review protocol with PRISMA 2020 item mapping. Online Resource 2 (ESM_2.xlsx): coded evidence-extraction matrix for the 98 corpus sources, including evidence grades, identification routes and verification sources.

References

- Abou Ali M, Dornaika F, Charafeddine J (2025) Agentic AI: a comprehensive survey of architectures, applications, and future directions. *Artif Intell Rev* 59:11. <https://doi.org/10.1007/s10462-025-11422-4>
- Acharya DB, Kuppan K, Divya B (2025) Agentic AI: autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access* 13:18912–18936. <https://doi.org/10.1109/ACCESS.2025.3532853>
- Aleithan R, Xue H, Mohajer MM, Nnorom E, Uddin G, Wang S (2024) SWE-Bench+: enhanced coding benchmark for LLMs. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2410.06992>
- Alva LR, Pandey B (2026) Agentic AI systems in the age of generative models: architectures, cloud scalability, and real-world applications. *Artif Intell Rev* 59:88. <https://doi.org/10.1007/s10462-025-11458-6>
- Andriushchenko M, Souly A, Dziemian M et al (2025) AgentHarm: a benchmark for measuring harmfulness of LLM agents. In: *Proceedings of the 13th International Conference on Learning Representations (ICLR 2025)*. <https://doi.org/10.48550/arXiv.2410.09024>
- Bai Y, Kadavath S, Kundu S et al (2022) Constitutional AI: harmlessness from AI feedback. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2212.08073>
- Boiko DA, MacKnight R, Kline B, Gomes G (2023) Autonomous chemical research with large language models. *Nature* 624:570–578. <https://doi.org/10.1038/s41586-023-06792-0>
- Borgeaud S, Mensch A, Hoffmann J et al (2022) Improving language models by retrieving from trillions of tokens. In: *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*. <https://doi.org/10.48550/arXiv.2112.04426>
- Brown TB, Mann B, Ryder N et al (2020) Language models are few-shot learners. In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. <https://doi.org/10.48550/arXiv.2005.14165>
- Cemri M, Pan MZ, Yang S, Agrawal LA, Chopra B, Tiwari R, Keutzer K, Parameswaran A, Klein D, Ramchandran K, Zaharia M, Gonzalez JE, Stoica I (2025) Why do multi-agent LLM systems fail? In: *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp 135676–135729. <https://doi.org/10.52202/085713-4082>
- Chan A, Salganik R, Markelius A et al (2023) Harms from increasingly agentic algorithmic systems. In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23)*, pp 651–666. <https://doi.org/10.1145/3593013.3594033>
- Chan A, Ezell C, Kaufmann M, Wei K, Hammond L, Bradley H, Bluemke E, Rajkumar N, Krueger D, Kolt N, Heim L, Anderljung M (2024) Visibility into AI agents. In: *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24)*, pp 958–973. <https://doi.org/10.1145/3630106.3658948>
- Chen W, Su Y, Zuo J, Yang C, Yuan C, Chan C-M, Yu H, Lu Y, Hung Y-H, Qian C, Qin Y, Cong X, Xie R, Liu Z, Sun M, Zhou J (2024) AgentVerse: facilitating multi-agent collaboration and exploring emergent behaviors. In: *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*. <https://doi.org/10.48550/arXiv.2308.10848>
- Chhikara P, Khant D, Aryan S, Singh T, Yadav D (2025) Mem0: building production-ready AI agents with scalable long-term memory. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2504.19413>

- Chowa SS, Alvi R, Rahman SS, Rahman MA, Raiaan MAK, Islam MR, Hussain M, Azam S (2026) From language to action: a review of large language models as autonomous agents and tool users. *Artif Intell Rev* 59:71. <https://doi.org/10.1007/s10462-025-11471-9>
- Debenedetti E, Zhang J, Balunović M, Beurer-Kellner L, Fischer M, Tramèr F (2024) AgentDojo: a dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*. <https://doi.org/10.48550/arXiv.2406.13352>
- Deng X, Gu Y, Zheng B, Chen S, Stevens S, Wang B, Sun H, Su Y (2023) Mind2Web: towards a generalist agent for the web. In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*. <https://doi.org/10.48550/arXiv.2306.06070>
- Du Y, Li S, Torralba A, Tenenbaum JB, Mordatch I (2024) Improving factuality and reasoning in language models through multiagent debate. In: *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*. <https://doi.org/10.48550/arXiv.2305.14325>
- European Parliament and Council of the European Union (2024) Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union L*, 12 July 2024. <http://data.europa.eu/eli/reg/2024/1689/oj>
- Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, Dai Y, Sun J, Wang M, Wang H (2023) Retrieval-augmented generation for large language models: a survey. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2312.10997>
- Gou Z, Shao Z, Gong Y, Shen Y, Yang Y, Duan N, Chen W (2024) CRITIC: large language models can self-correct with tool-interactive critiquing. In: *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*. <https://doi.org/10.48550/arXiv.2305.11738>
- Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M (2023) Not what you've signed up for: compromising real-world LLM-integrated applications with indirect prompt injection. In: *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*, pp 79–90. <https://doi.org/10.1145/3605764.3623985>
- Guo T, Chen X, Wang Y, Chang R, Pei S, Chawla NV, Wiest O, Zhang X (2024) Large language model based multi-agents: a survey of progress and challenges. In: *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*, pp 8048–8057. <https://doi.org/10.24963/ijcai.2024/890>
- Guo D, Yang D, Zhang H et al (2025) DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* 645:633–638. <https://doi.org/10.1038/s41586-025-09422-z>
- Guu K, Lee K, Tung Z, Pasupat P, Chang M-W (2020) Retrieval augmented language model pre-training. In: *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*. <https://doi.org/10.48550/arXiv.2002.08909>
- Hao S, Gu Y, Ma H, Hong JJ, Wang Z, Wang DZ, Hu Z (2023) Reasoning with language model is planning with world model. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pp 8154–8173. <https://doi.org/10.18653/v1/2023.emnlp-main.507>
- He F, Zhu T, Ye D, Liu B, Zhou W, Yu PS (2025) The emerged security and privacy of LLM agent: a survey with case studies. *ACM Comput Surv* 58(6):1–36. <https://doi.org/10.1145/3773080>
- Hong S, Zhuge M, Chen J, Zheng X, Cheng Y, Zhang C, Wang J, Wang Z, Yau SKS, Lin Z, Zhou L, Ran C, Xiao L, Wu C, Schmidhuber J (2024) MetaGPT: meta programming for a multi-agent collaborative framework. In: *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*. <https://doi.org/10.48550/arXiv.2308.00352>

- Hou X, Zhao Y, Wang S, Wang H (2025) Model Context Protocol (MCP): landscape, security threats, and future research directions. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2503.23278>
- Hua W, Yang X, Jin M, Li Z, Cheng W, Tang R, Zhang Y (2024) TrustAgent: towards safe and trustworthy LLM-based agents. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp 10000–10016. <https://doi.org/10.18653/v1/2024.findings-emnlp.585>
- Huang J, Chen X, Mishra S, Zheng HS, Yu AW, Song X, Zhou D (2024) Large language models cannot self-correct reasoning yet. In: *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*. <https://doi.org/10.48550/arXiv.2310.01798>
- Huang X, Liu W, Chen X, Wang X, Wang H, Lian D, Wang Y, Tang R, Chen E (2024) Understanding the planning of LLM agents: a survey. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2402.02716>
- Huang J-t, Zhou J, Jin T, Zhou X, Chen Z, Wang W, Yuan Y, Lyu M, Sap M (2025) On the resilience of LLM-based multi-agent collaboration with faulty agents. In: *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, PMLR 267:26202–26226. <https://doi.org/10.48550/arXiv.2408.00989>
- Infocomm Media Development Authority (2026) Model AI governance framework for agentic AI. Government of Singapore. <https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf>. Accessed 30 September 2026
- Jimenez CE, Yang J, Wettig A, Yao S, Pei K, Press O, Narasimhan K (2024) SWE-bench: can language models resolve real-world GitHub issues? In: *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*. <https://doi.org/10.48550/arXiv.2310.06770>
- Kambhampati S, Valmeekam K, Guan L, Verma M, Stechly K, Bhambri S, Saldyt L, Murthy A (2024) Position: LLMs can't plan, but can help planning in LLM-Modulo frameworks. In: *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*. <https://doi.org/10.48550/arXiv.2402.01817>
- Kapoor S, Stroebl B, Siegel ZS, Nadgir N, Narayanan A (2024) AI agents that matter. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2407.01502>
- Kehkashan T, Abdullah M, Al-Shamayleh AS, Ivković N, Ismail NA, Ahmad SSS, Rehman A, Akhonzada A (2026) From benchmarks to deployment: a comprehensive review of agentic AI evaluation. *Artif Intell Rev* 59:167. <https://doi.org/10.1007/s10462-026-11571-0>
- Khadka A, Maple C (2026) Artificial intelligence (AI) assurance: challenges, gaps, and the path forward. *Artif Intell Rev*. <https://doi.org/10.1007/s10462-026-11719-y>
- Kwa T, West B, Becker J et al (2025) Measuring AI ability to complete long software tasks. In: *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*. <https://doi.org/10.48550/arXiv.2503.14499>
- Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W, Rocktäschel T, Riedel S, Kiela D (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. <https://doi.org/10.48550/arXiv.2005.11401>
- Li G, Hammoud HAAK, Itani H, Khizbullin D, Ghanem B (2023) CAMEL: communicative agents for "mind" exploration of large language model society. In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. <https://doi.org/10.48550/arXiv.2303.17760>
- Li N, Gao C, Li M, Li Y, Liao Q (2024) EconAgent: large language model-empowered agents for simulating macroeconomic activities. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, pp 15523–15536. <https://doi.org/10.18653/v1/2024.acl-long.829>

- Liu B, Jiang Y, Zhang X, Liu Q, Zhang S, Biswas J, Stone P (2023) LLM+P: empowering large language models with optimal planning proficiency. arXiv preprint. <https://doi.org/10.48550/arXiv.2304.11477>
- 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, Huang M, Dong Y, Tang J (2024) AgentBench: evaluating LLMs as agents. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024). <https://doi.org/10.48550/arXiv.2308.03688>
- Ma C, Zhang J, Zhu Z, Yang C, Yang Y, Jin Y, Lan Z, Kong L, He J (2024) AgentBoard: an analytical evaluation board of multi-turn LLM agents. In: Advances in Neural Information Processing Systems 37 (NeurIPS 2024). <https://doi.org/10.48550/arXiv.2401.13178>
- Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegrefe S, Alon U, Dziri N, Prabhume S, Yang Y, Gupta S, Majumder BP, Hermann K, Welleck S, Yazdanbakhsh A, Clark P (2023) Self-Refine: iterative refinement with self-feedback. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023). <https://doi.org/10.48550/arXiv.2303.17651>
- Mialon G, Fourrier C, Swift C, Wolf T, LeCun Y, Scialom T (2024) GAIA: a benchmark for general AI assistants. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024). <https://doi.org/10.48550/arXiv.2311.12983>
- Mnih V, Kavukcuoglu K, Silver D et al (2015) Human-level control through deep reinforcement learning. *Nature* 518:529–533. <https://doi.org/10.1038/nature14236>
- Ouyang L, Wu J, Jiang X et al (2022) Training language models to follow instructions with human feedback. In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022). <https://doi.org/10.48550/arXiv.2203.02155>
- Packer C, Wooders S, Lin K, Fang V, Patil SG, Stoica I, Gonzalez JE (2024) MemGPT: towards LLMs as operating systems. arXiv preprint. <https://doi.org/10.48550/arXiv.2310.08560>
- Page MJ, McKenzie JE, Bossuyt PM et al (2021) The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* 372:n71. <https://doi.org/10.1136/bmj.n71>
- Park JS, O'Brien JC, Cai CJ, Morris MR, Liang P, Bernstein MS (2023) Generative agents: interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), pp 1–22. <https://doi.org/10.1145/3586183.3606763>
- Patil SG, Zhang T, Wang X, Gonzalez JE (2024) Gorilla: large language model connected with massive APIs. In: Advances in Neural Information Processing Systems 37 (NeurIPS 2024). <https://doi.org/10.48550/arXiv.2305.15334>
- Plaat A, van Duijn M, van Stein N, Preuss M, van der Putten P, Batenburg KJ (2025) Agentic large language models, a survey. *J Artif Intell Res* 84:29. <https://doi.org/10.48550/arXiv.2503.23037>
- Qian C, Liu W, Liu H, Chen N, Dang Y, Li J, Yang C, Chen W, Su Y, Cong X, Xu J, Li D, Liu Z, Sun M (2024) ChatDev: communicative agents for software development. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), pp 15174–15186. <https://doi.org/10.18653/v1/2024.acl-long.810>
- Qin Y, Liang S, Ye Y, Zhu K, Yan L, Lu Y, Lin Y, Cong X, Tang X, Qian B, Zhao S, Hong L, Tian R, Xie R, Zhou J, Gerstein M, Li D, Liu Z, Sun M (2024) ToolLLM: facilitating large language models to master 16000+ real-world APIs. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024). <https://doi.org/10.48550/arXiv.2307.16789>
- Qu C, Dai S, Wei X, Cai H, Wang S, Yin D, Xu J, Wen J-R (2025) Tool learning with large language models: a survey. *Front Comput Sci* 19(8):198343. <https://doi.org/10.1007/s11704-024-40678-2>

- Rao AS, Georgeff MP (1995) BDI agents: from theory to practice. In: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). AAAI Press, Menlo Park, pp 312–319
- Romera-Paredes B, Barekatin M, Novikov A et al (2024) Mathematical discoveries from program search with large language models. *Nature* 625:468–475. <https://doi.org/10.1038/s41586-023-06924-6>
- Ruan Y, Dong H, Wang A, Pitis S, Zhou Y, Ba J, Dubois Y, Maddison CJ, Hashimoto T (2024) Identifying the risks of LM agents with an LM-emulated sandbox. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024). <https://doi.org/10.48550/arXiv.2309.15817>
- Russell SJ, Norvig P (2021) Artificial intelligence: a modern approach, 4th edn. Pearson, Hoboken
- Sapkota R, Roumeliotis KI, Karkee M (2026) AI agents vs. agentic AI: a conceptual taxonomy, applications and challenges. *Inf Fusion* 126:103599. <https://doi.org/10.1016/j.inffus.2025.103599>
- Schick T, Dwivedi-Yu J, Dessi R, Raileanu R, Lomeli M, Zettlemoyer L, Cancedda N, Scialom T (2023) Toolformer: language models can teach themselves to use tools. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023). <https://doi.org/10.48550/arXiv.2302.04761>
- Shen Y, Song K, Tan X, Li D, Lu W, Zhuang Y (2023) HuggingGPT: solving AI tasks with ChatGPT and its friends in Hugging Face. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023). <https://doi.org/10.48550/arXiv.2303.17580>
- Shinn N, Cassano F, Berman E, Gopinath A, Narasimhan K, Yao S (2023) Reflexion: language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023). <https://doi.org/10.48550/arXiv.2303.11366>
- Shojaee P, Mirzadeh I, Alizadeh K, Horton M, Bengio S, Farajtabar M (2025) The illusion of thinking: understanding the strengths and limitations of reasoning models via the lens of problem complexity. In: Advances in Neural Information Processing Systems 38 (NeurIPS 2025). <https://doi.org/10.48550/arXiv.2506.06941>
- Silver D, Huang A, Maddison CJ et al (2016) Mastering the game of Go with deep neural networks and tree search. *Nature* 529:484–489. <https://doi.org/10.1038/nature16961>
- Stechly K, Valmeekam K, Kambhampati S (2024) On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2402.08115>
- Sumers TR, Yao S, Narasimhan K, Griffiths TL (2024) Cognitive architectures for language agents. *Trans Mach Learn Res*. <https://doi.org/10.48550/arXiv.2309.02427>
- Sutton RS, Barto AG (2018) Reinforcement learning: an introduction, 2nd edn. MIT Press, Cambridge
- Tabassi E (2023) Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1. National Institute of Standards and Technology, Gaithersburg. <https://doi.org/10.6028/NIST.AI.100-1>
- Tran D, Kiela D (2026) Single-agent LLMs outperform multi-agent systems on multi-hop reasoning under equal thinking token budgets. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2604.02460>
- Valmeekam K, Marquez M, Olmo A, Sreedharan S, Kambhampati S (2023) PlanBench: an extensible benchmark for evaluating large language models on planning and reasoning about change. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track. <https://doi.org/10.48550/arXiv.2206.10498>

- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. <https://doi.org/10.48550/arXiv.1706.03762>
- Verma M, Bhambri S, Kambhampati S (2024) On the brittle foundations of ReAct prompting for agentic large language models. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2405.13966>
- Wang L, Xu W, Lan Y, Hu Z, Lan Y, Lee RK-W, Lim E-P (2023) Plan-and-solve prompting: improving zero-shot chain-of-thought reasoning by large language models. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pp 2609–2634. <https://doi.org/10.18653/v1/2023.acl-long.147>
- Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, Chowdhery A, Zhou D (2023) Self-consistency improves chain of thought reasoning in language models. In: *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. <https://doi.org/10.48550/arXiv.2203.11171>
- Wang G, Xie Y, Jiang Y, Mandlekar A, Xiao C, Zhu Y, Fan L, Anandkumar A (2024) Voyager: an open-ended embodied agent with large language models. *Trans Mach Learn Res*. <https://doi.org/10.48550/arXiv.2305.16291>
- Wang L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, Chen Z, Tang J, Chen X, Lin Y, Zhao WX, Wei Z, Wen J-R (2024) A survey on large language model based autonomous agents. *Front Comput Sci* 18(6):186345. <https://doi.org/10.1007/s11704-024-40231-1>
- Wang X, Chen Y, Yuan L, Zhang Y, Li Y, Peng H, Ji H (2024) Executable code actions elicit better LLM agents. In: *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*. <https://doi.org/10.48550/arXiv.2402.01030>
- Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi E, Le Q, Zhou D (2022) Chain-of-thought prompting elicits reasoning in large language models. In: *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. <https://doi.org/10.48550/arXiv.2201.11903>
- Wooldridge M, Jennings NR (1995) Intelligent agents: theory and practice. *Knowl Eng Rev* 10(2):115–152. <https://doi.org/10.1017/S0269888900008122>
- Wu Q, Bansal G, Zhang J, Wu Y, Li B, Zhu E, Jiang L, Zhang X, Zhang S, Liu J, Awadallah AH, White RW, Burger D, Wang C (2024) AutoGen: enabling next-gen LLM applications via multi-agent conversation. In: *Proceedings of the First Conference on Language Modeling (COLM 2024)*. <https://doi.org/10.48550/arXiv.2308.08155>
- Xi Z, Chen W, Guo X et al (2025) The rise and potential of large language model based agents: a survey. *Sci China Inf Sci* 68(2):121101. <https://doi.org/10.1007/s11432-024-4222-0>
- Xie T, Zhang D, Chen J, Li X, Zhao S, Cao R, Hua TJ, Cheng Z, Shin D, Lei F et al (2024) OSWorld: benchmarking multimodal agents for open-ended tasks in real computer environments. In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, Datasets and Benchmarks Track. <https://doi.org/10.48550/arXiv.2404.07972>
- Xu FF, Song Y, Li B, Tang Y et al (2024) TheAgentCompany: benchmarking LLM agents on consequential real world tasks. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2412.14161>
- Xu W, Liang Z, Mei K, Gao H, Tan J, Zhang Y (2025) A-MEM: agentic memory for LLM agents. In: *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*. <https://doi.org/10.48550/arXiv.2502.12110>
- Xu Y, Zhuang Y, Liu X, Zhang T, Xiao B, Xu X, Jiang D, Wang J, Hu H (2026) LLM agents security duality: a comprehensive survey of self-security and empowered cybersecurity. *Artif Intell Rev* 59:174. <https://doi.org/10.1007/s10462-026-11563-0>

- Yang J, Jimenez CE, Wettig A, Lieret K, Yao S, Narasimhan K, Press O (2024) SWE-agent: agent-computer interfaces enable automated software engineering. In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. <https://doi.org/10.48550/arXiv.2405.15793>
- Yang W, Bi X, Lin Y, Chen S, Zhou J, Sun X (2024) Watch out for your agents! Investigating backdoor threats to LLM-based agents. In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. <https://doi.org/10.48550/arXiv.2402.11208>
- Yao S, Chen H, Yang J, Narasimhan K (2022) WebShop: towards scalable real-world web interaction with grounded language agents. In: *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. <https://doi.org/10.48550/arXiv.2207.01206>
- Yao S, Yu D, Zhao J, Shafran I, Griffiths TL, Cao Y, Narasimhan K (2023a) Tree of thoughts: deliberate problem solving with large language models. In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. <https://doi.org/10.48550/arXiv.2305.10601>
- Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y (2023b) ReAct: synergizing reasoning and acting in language models. In: *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. <https://doi.org/10.48550/arXiv.2210.03629>
- Yao S, Shinn N, Razavi P, Narasimhan K (2024) τ -bench: a benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2406.12045>
- 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>
- Zhang Z, Bo X, Ma C, Li R, Chen X, Dai Q, Zhu J, Dong Z, Wen J-R (2024) A survey on the memory mechanism of large language model based agents. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2404.13501>
- Zhong W, Guo L, Gao Q, Ye H, Wang Y (2024) MemoryBank: enhancing large language models with long-term memory. In: *Proceedings of the AAAI Conference on Artificial Intelligence 38(17)*:19724–19731. <https://doi.org/10.1609/aaai.v38i17.29946>
- Zhou S, Xu FF, 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: *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*. <https://doi.org/10.48550/arXiv.2307.13854>
- Zhuge M, Zhao C, Ashley D, Wang W, Khizbullin D, Xiong Y, Liu Z, Chang E, Krishnamoorthi R, Tian Y et al (2024) Agent-as-a-judge: evaluate agents with agents. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2410.10934>