Artificial Intelligence Review

https://doi.org/10.1007/s10462-026-11682-8

Article in Press

Towards safe and trustworthy agentic AI: foundations, taxonomy, technologies, applications, and future directions

Received: 26 January 2026

Accepted: 10 August 2026

Published online: 24 August 2026

Cite this article as: Gohil V., Shah S.B., Rauniyar K. *et al.* Towards safe and trustworthy agentic AI: foundations, taxonomy, technologies, applications, and future directions. *Artif Intell Rev* (2026). https://doi.org/10.1007/s10462-026-11682-8

Vijayrajsinh Gohil, Siddhant Bikram Shah, Kritesh Rauniyar, Shuvam Shiwakoti, Surendrabikram Thapa, Diwei Sheng, Laxmi Thapa, Kristina T. Johnson & Usman Naseem

Towards safe and trustworthy agentic AI: foundations, taxonomy, technologies, applications, and future directions

Towards safe and trustworthy agentic AI: Foundations, taxonomy, technologies, applications, and future directions

Vijayrajsinh Gohil[1†], Siddhant Bikram Shah[2†], Kritesh Rauniyar[6], Shuvam Shiwakoti[3], Surendrabikram Thapa[3], Diwei Sheng[1], Laxmi Thapa[4], Kristina T. Johnson[2], Usman Naseem[6*]

[1]New York University, New York City, 10012, New York, USA.
[2]Northeastern University, Boston, 02115, Massachusetts, USA.
[3]Virginia Tech, Blacksburg, 24060, Virginia, USA.
[4] O.P. Jindal Global University , Sonipat, 131001, Haryana, India.
[5]School of Computing, Edinburgh Napier University, Edinburgh, EH10 5DT, Scotland, UK.
[6*]School of Computing, Macquarie University, Sydney, 2109, NSW, Australia.

*Corresponding author(s). E-mail(s): usman.naseem@mq.edu.au;
†These authors contributed equally to this work.

Abstract

Agentic Artificial Intelligence (AI) represents a paradigm shift from static, task-specific systems to autonomous, goal-directed agents capable of reasoning, planning, learning, and acting with minimal human oversight. This paper synthesizes perspectives from philosophy, cognitive science, and AI to define agency, outline its key properties, and situate it in relation to existing paradigms such as reinforcement learning, symbolic reasoning, Belief–Desire–Intention (BDI) architectures, and embodied cognition. We present a taxonomy of agentic systems along dimensions of autonomy, cognitive capability, modality, and environmental interaction, highlighting current capabilities and limitations, and we critically delimit where such a taxonomy is informative and where a functional, closed-loop analysis of agent behavior must take over. The enabling technologies, including large language models, memory architectures, planning frameworks, tool-use mechanisms, and multimodal embodiment, are reviewed alongside diverse application domains ranging from autonomous research and creative systems to web

automation and human–AI collaboration. We analyze safety, alignment, evaluation challenges, and emergent risks, dedicate a section to trustworthiness, privacy, and sustainability by bridging from trustworthy machine learning, verified autonomy, and privacy-preserving learning, and compile a comparative review of prominent benchmarks for assessing agentic behavior. Finally, we outline future research directions, including compositional and modular architectures, cooperative AI, simulation-based safe exploration, data-efficient and developmentally inspired learning, hybrid symbolic–neural systems, and strategies for robust alignment. By providing a comprehensive framework and critical analysis, this work aims to guide the development of agentic AI systems that are not only capable but also safe, trustworthy, and aligned with human values.

Keywords: Agentic AI, Autonomy, Multi-agent systems, Large language models

1 Introduction

Artificial intelligence (AI) and Machine Learning (ML) are undergoing a paradigm shift, with systems transitioning from tools that execute predefined tasks to autonomous agents capable of independent reasoning, planning, and action [1–3]. This shift signals the emergence of a new class of AI systems that can pursue objectives, adapt to changing circumstances, and make decisions with minimal human oversight: Agentic AI. In classical AI paradigms, systems were either rule-based expert systems, which applied predetermined logical inferences [4], or statistical models, which mapped inputs to outputs based on learned correlations [5]. While powerful, these systems cannot deliberate about future states or revise objectives dynamically. In contrast, an agentic AI system treats decision-making as a planning process that actively weighs alternative courses of action, anticipates downstream effects, and can recover from unexpected events by replanning [6]. This allows these systems to tackle complex challenges such as managing complex workflows [7], obstacle avoidance [8], and conducting multi-step scientific experiments [9].

The current surge of interest in agentic AI is propelled by several converging advancements and an increasing demand for more sophisticated autonomous capabilities. The most significant contributor has been the emergence of Large Language Models (LLMs) [10]. LLMs have provided an unprecedented foundation for cognitive tasks, offering agents the ability to understand nuanced instructions, reason about the world, generate creative solutions, and even produce code to interact with other software tools and APIs [11]. Simultaneously, advances in Reinforcement Learning (RL) have enabled systems to learn optimal behaviors through interaction with their environments [12]. The synthesis of these technologies has given rise to agents like Auto-GPT [13] and BabyAGI[1] that can autonomously decompose high-level objectives into subtasks, retrieve and synthesize information, and generate code to orchestrate further actions. These agents can browse the web, call external APIs, and spawn

[1] https://github.com/yoheinakajima/babyagi

subprocesses, blurring the line between language comprehension and autonomous reasoning. Figure 1 illustrates the canonical workflow of such an LLM-based agent, in which a reasoning engine coordinates perception, memory, planning, and tool use within an environmental feedback loop; we use this workflow as a reference point throughout the survey.

Beyond software-based agents, the principles of agentic AI are central to the progression of robotics. For robots to operate effectively in unstructured human environments like hospitals and factories, they require the ability to understand abstract commands, formulate and execute sequential multi-step plans, adapt intelligently to their surroundings, and learn from new experiences [14]. The increasing complexity of tasks humans wish to automate necessitates systems that can deliberate, strategize, and demonstrate genuine problem-solving initiative. The combination of symbolic planning with probabilistic reasoning [15], differentiable planning [16], and Monte Carlo Tree Search allows systems to handle environments that exhibit partial observability, stochastic transitions, and continuous action spaces. Decades of robotics research on simultaneous localization and mapping (SLAM) [17], task-and-motion planning [18], and planning under uncertainty supply mature machinery for exactly these conditions, and we draw on this literature throughout rather than treating embodied agency as a recent invention. Further, the availability of large-scale computational resources has enabled these planners to scale to real-world complexities that were previously intractable. While the traditional robotics pipeline of sensing, planning, and acting [19, 20] remains conceptually intact, Agentic AI provides the reasoning capabilities to greatly enhance the sophistication of each component.

However, the pursuit of increasingly autonomous AI also makes the alignment problem increasingly urgent. As agentic systems gain more freedom to act and make decisions, ensuring that their objectives remain aligned with human values becomes critically important and technically challenging [21]. The development of agentic AI is therefore intrinsically linked to research in AI safety and ethics, with the goal of creating systems that are not only competent but also trustworthy. The potential for autonomous systems to misunderstand objectives, exhibit unintended behaviors, or be misused underscores the urgency of addressing alignment proactively. The central thesis organizing this survey is that capability and trustworthiness in agentic AI are not separable concerns to be addressed in sequence, but properties that must be co-designed: the same affordances that make an agent useful—persistent goal pursuit, tool use, long-horizon planning, and self-directed action—are precisely those that make it hazardous when its objective is underspecified or its oversight is incomplete.

This paper aims to provide a comprehensive overview of the current landscape of Agentic AI, organized as a single narrative thread that moves from what agency is, to how agentic systems can be classified (and where classification stops being the right tool), to the technologies that realize them, the applications they enable, the risks they pose, the trustworthiness machinery needed to deploy them responsibly, and the future research required to close the remaining gaps. Section 2 **Foundations of Agentic AI** discusses the philosophical underpinnings of agency, exploring concepts like intentionality, motivational systems, free will, and Kantian autonomy. It then

defines the key properties of an agentic system—autonomy, goal-directedness, perception, memory, planning, and feedback loops. It also examines how agentic AI relates to existing concepts like reinforcement learning agents, symbolic and BDI architectures, embodied cognition, and open-ended learning, and traces its historical trajectory from early systems like SHRDLU to modern LLM-based agents. Following this, in Section 3 **Taxonomy of Agentic AI Systems**, we present a detailed taxonomy, categorizing systems by their level of autonomy (from reactive to fully deliberative), cognitive capabilities (planning, memory, and learning), and environment interaction (closed- vs. open-world); critically, we close the section by distinguishing descriptive taxonomy from functional modeling, arguing that taxonomic coverage does not imply behavioral, safety, or alignment completeness. Section 4 **Technologies Behind Agentic AI Systems** discusses the core technologies that enable agentic behavior: LLMs as reasoning engines, memory systems ranging from short-term context windows to long-term retrieval-augmented generation (RAG), planning frameworks like Monte Carlo Tree Search and PDDL, tool-use mechanisms including function-calling APIs and plugins, and the orchestration frameworks that assemble these primitives into deployable systems. We further explore how multimodal and embodied agents extend agency into physical environments. In Section 5 **Applications of Agentic AI**, we highlight real-world use cases of Agentic AI, like autonomous research and scientific reasoning. We further discuss creative systems for game design and narrative generation, web-based agents that automate search and coding tasks, and collaborative human–AI copilots in education, software development, and healthcare. Section 6 **Challenges and Risks** examines safety and alignment risks like misaligned goals, reward hacking, and instrumental convergence; evaluation challenges like emergent misbehavior, self-looping, and hallucinations; and ethical or legal concerns surrounding autonomy and accountability, surveying existing benchmarks like WebArena, SWE-bench, and SciBench. Building directly on the survey's thesis, Section 7 **Toward Trustworthy Agentic AI** consolidates safety, robustness, privacy, and sustainability into a dedicated treatment, bridging from trustworthy machine learning, verified autonomy, and privacy-preserving learning, and proposing a concrete research agenda.

Finally, in Section 8 **Future Directions**, we explore upcoming research frontiers—including compositional and modular architectures, social agency and cooperative AI, agentic alignment research, safe exploration via simulation environments, data-efficient and developmentally inspired learning, and hybrid symbolic-neural systems. We discuss strategies for super-alignment—the project of ensuring that AI systems whose capabilities may eventually exceed those of their human supervisors nevertheless remain safely and beneficially aligned with human values—that combine scalable human feedback, formal verification, and adversarial testing. We reiterate the rising influence of Agentic AI and envision what safe, useful, and ethically aligned systems should look like. Through this paper, we aim to highlight both the opportunities and responsibilities of agentic AI, advocating for systems that are not only capable but also aligned with societal welfare.

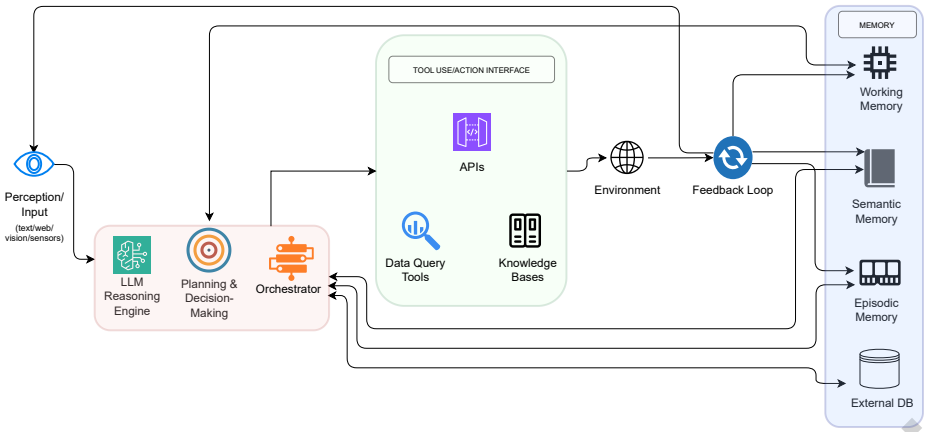

Fig. 1 LLM Agent workflow overview. Perceptual inputs are processed by an LLM reasoning engine and planning module, while an orchestrator coordinates interactions with external tools, APIs, knowledge bases, and the environment. Feedback from tool execution updates multiple memory components, including memory, enabling iterative reasoning, decision-making, and adaptive behavior.

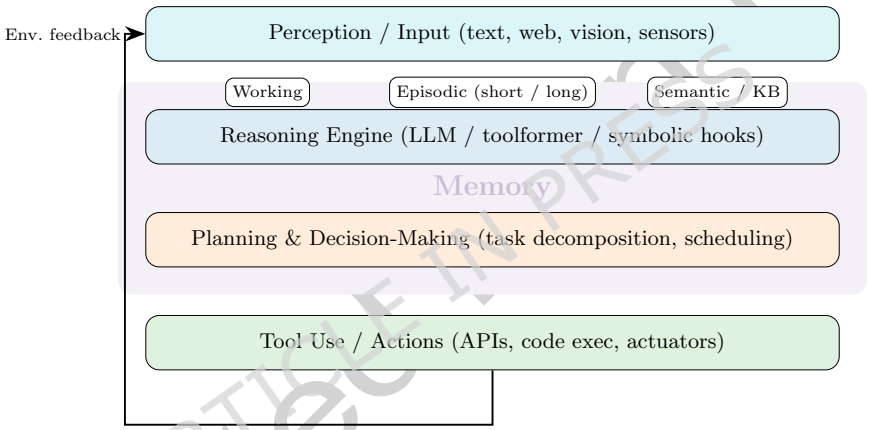

Fig. 2 Layered agent architecture. (Tables 3, 4). The agent integrates perception, memory, reasoning, planning, and tool-use modules within a closed feedback loop, enabling it to process multimodal inputs, maintain context, make decisions, and interact with external environments through actions and tools.

2 Foundations of Agentic AI

Before agentic AI can be classified, engineered, or governed, it must be grounded conceptually. We begin with the philosophical question of what agency is and what would license attributing it to an artificial system (Section 2.1); we use that analysis to motivate a minimal set of operational properties that any agentic system must exhibit, including the perceptual grounding that anchors the rest (Section 2.2); we then

situate these properties against the established AI paradigms—reinforcement learning, symbolic and BDI systems, embodied cognition, and open-ended learning—that each capture some but not all of them (Section 2.3); and we close with a historical trajectory showing how these strands converged on the present generation of agents (Section 2.4). The connecting thread is that contemporary agentic AI is best understood not as a rupture but as a synthesis whose components, and whose unresolved tensions, are inherited directly from these traditions.

2.1 Philosophical Foundations of Agency

A critical starting point for understanding agentic AI is to ground it in the rich history of philosophical inquiry into agency. By examining how agency has been conceptualized across major philosophical traditions, we gain valuable frameworks to assess and interrogate the capabilities and limitations of agentic AI systems. An agent is fundamentally understood as a being with the capacity to act, and agency is the exercise of this capacity [22]. Philosophical explorations delve into what distinguishes an intentional action from a mere event, a crucial distinction for evaluating the behavior of AI [23].

2.1.1 Intentionality: The Aboutness of Action

A cornerstone of agency is the concept of intentionality, most notably formulated by Franz Brentano in *Psychology from an Empirical Standpoint* [24]. Intentionality refers to the "aboutness" of mental states–the property of thoughts, beliefs, and desires being directed toward objects or states of affairs [22]. In the philosophy of action, behavior is considered an "action" when it is caused by the agent's mental states, such as their beliefs and desires [23].

In the context of agentic AI, intentionality is often simulated rather than possessed. An LLM-based agent might exhibit behaviors that appear goal-directed, but this is typically a result of internal prompt-conditioning or environmental feedback. Whether these behaviors stem from genuine intentional states or are a sophisticated mimicry remains an open philosophical challenge [22].

Intentionality concerns what mental states are about and where the direction of action originates—i.e., the agent's motivational system. Philosophical accounts of desire and conation have direct computational counterparts in work on artificial motivation. Extrinsic, task-defined objectives (reward functions, goal specifications) capture only one mode of motivation; a substantial body of research argues that genuinely open-ended, self-directed agency requires intrinsic motivation—internally generated drives such as novelty, curiosity, learning progress, competence, and empowerment that lead an agent to set its own subgoals in the absence of external reward [25–27]. Recent formal treatments distinguish the goals an agent is given from the purposes it forms, framing autonomy as the capacity to generate and pursue self-selected objectives while remaining aligned with externally specified constraints [28]. This distinction is consequential for our later discussion of autonomy levels and of alignment: a system whose motivation is wholly extrinsic cannot, by construction, "define its own goals," whereas a system endowed with intrinsic motivation can—raising both the

possibility of truly autonomous agency and the attendant difficulty of ensuring that self-generated purposes remain beneficial.

2.1.2 Free Will and Determinism

The concept of agency is often intertwined with the metaphysical debate over free will–the capacity to choose between different possible courses of action unimpeded [29]. Philosophical positions on this issue include:

- **Determinism:** All events, including human actions, are necessitated by antecedent causes and effects, rendering free will illusory [30].
- **Compatibilism:** Free will and determinism are reconcilable; free will is the freedom to act according to one's desires without coercion, even if those desires are themselves causally determined [31, 32].
- **Libertarianism:** Agents possess genuine freedom of choice, independent of deterministic causal chains [29].

These stances provide analogues for interrogating AI: if an AI selects actions via optimization or randomness, does that simulate a form of free will? While current systems lack consciousness, they can model apparent freedom under constraints [31].

2.1.3 The Structure vs. Agency Debate

Philosophers and sociologists have long examined the tension between individual agency and social structure–the deterministic influence of laws, norms, and institutions on behavior [33, 34]. For agentic AI, structure encompasses architecture, training-data biases, reward functions, and environmental rules. Understanding how these constraints shape–or limit–an AI's emergent behaviors is vital for safe and aligned design.

2.1.4 Kantian Autonomy: Agency as Rational Self-Governance

Immanuel Kant defines autonomy as the capacity for rational self-governance: acting according to principles one adopts for oneself, specifically the categorical imperative [35]. We foreground the Kantian conception here deliberately, and for two reasons rather than to the exclusion of alternatives. First, it sets the most demanding bar available: rational self-governance requires reflective endorsement of one's own principles, not merely the unimpeded pursuit of given ends, and it is precisely this bar that reward-maximizing systems most conspicuously fail to clear. It therefore furnishes a sharp diagnostic for what current agents lack. Second, contemporary alignment debates about corrigibility (an agent's willingness to be corrected) and about agents that resist modification of their own objectives are, in effect, twenty-first-century restatements of the Kantian problem of self-legislation, making the connection more than historical. We acknowledge that other conceptions—autonomy as independence from external control, relational autonomy that locates self-governance within a web of social dependencies, and the engineering sense of autonomy as operation without human intervention—are equally legitimate and, indeed, more directly operationalizable; the taxonomy in Section 3 adopts the engineering sense, while this section retains

the Kantian one as a normative ideal against which to measure it. The contrast is itself informative: a system can be highly autonomous in the engineering sense while possessing none of the rational self-governance Kant requires.

2.1.5 Existentialism: Agency as Radical Freedom and Responsibility

Existentialist thinkers such as Jean-Paul Sartre argue that agency arises from radical freedom, whereby individuals define their own essence through choices and bear full responsibility for their actions [36]. For AI, this perspective invites us to consider whether—and how—an agent might develop a narrative identity and assume responsibility for its outputs, a question that becomes concrete in Section 6 when we ask who is accountable for the actions of an autonomous system. Each of these traditions isolates a different ingredient of agency: directedness toward objects, freedom within causal structure, rational self-governance, and responsibility. The next section extracts from them the operational properties an artificial system would need to exhibit.

2.2 Key Properties

Section 2.1 asked what agency is; this section translates that analysis into a minimal set of operational properties an artificial system must exhibit before the term agent is warranted, and these are the load-bearing modules of the architecture in Figure 2. The selection is not arbitrary. Intentionality and the motivational question of Section 2.1.1 become goal-directedness together with the question of where goals originate; the autonomy debates of Sections 2.1.2 through 2.1.4 become autonomy in the engineering sense, exercised within designed constraints; the grounding of intentional states becomes perception; practical reasoning becomes planning; and the persistence of an agent through time becomes memory and feedback. The set also agrees with canonical definitions from the agents literature, which characterize an agent as a system that senses its environment and acts on it over time in pursuit of its own agenda [37], exhibiting autonomy, reactivity, and pro-activeness [38]. We place perception first because it anchors the rest. Social ability, the remaining property in Wooldridge and Jennings's classic list, concerns interaction between agents rather than the constitution of a single one, and we defer it to the treatment of cooperative AI in Section 8.2.

2.2.1 Perception

Perception is an agent's capacity to transform raw sensory or input signals into internal representations of the state of its environment, and it is the foundation on which the remaining properties rest. Without grounded perceptual representations, planning has no state space to search over, memory has no contents to store and retrieve, goal-directedness cannot be evaluated against world states, and learning degenerates into pattern matching over ungrounded tokens rather than acquisition of knowledge about a world. This priority is a central commitment of embodied and situated accounts of cognition, which hold that intelligent behavior emerges from closed-loop coupling between an agent and its environment rather than from abstract reasoning over symbols whose meaning is supplied externally [39, 40]. The depth of the representation an

agent must build is task-dependent: a text-based assistant perceives through tokenization and retrieval; a multimodal agent fuses vision, language, and audio encoders; and an embodied agent must perform online state estimation, fusing noisy and partial sensor streams into a coherent world model, the canonical instance being simultaneous localization and mapping (SLAM) in robotics [17, 41]. Crucially, perception is not a one-shot preprocessing stage but a continual, active process: agents direct their sensing, resolve perceptual ambiguity through action, and update their representations as the world changes. The grounding problem connecting internal symbols to perceptual experience and real-world referents identified decades ago in symbolic AI [40] remains the deepest open challenge here, and it reappears in modern form when LLM-based agents manipulate concepts whose perceptual referents they have never encountered.

2.2.2 Autonomy

Autonomy is the capacity of an agent to make decisions and take actions in pursuit of goals or objectives without constant human intervention. This ranges from simple reactive behavior to sophisticated, independent decision-making, yet operates within defined constraints: agents require goals and rules provided by humans [42]. A canonical definition describes an autonomous agent as one capable of flexible, autonomous action in complex, dynamic environments [37]. As noted in Section 2.1, this engineering sense of autonomy—operation without step-by-step human direction—should be distinguished from the stronger Kantian sense of rational self-governance; the two can come apart, and an agent may score highly on one while lacking the other entirely.

2.2.3 Goal-Directedness

Goal-directedness is the property of acting to achieve specific outcomes. It involves two key stages:

1. **Goal Formulation and Representation:** Clearly defining what the agent aims to achieve, with the ability to refine and update goals based on new information and changing circumstances [43].
2. **Goal-Directed Action Selection:** Selecting actions via means–end reasoning, where agents monitor their environment and choose efficient means to achieve their goals, governed by principles of rational action [44].

Where goals originate—whether supplied externally or generated intrinsically—is itself a substantive question, treated in Section 2.1; an agent that can only pursue externally supplied goals occupies a different point in the autonomy taxonomy of Section 3 than one that can formulate its own.

2.2.4 Memory

Memory is an agent's ability to store and recall past experiences to improve decision-making, perception, and overall performance. By retaining context, recognizing patterns over time, and adapting based on sequential experiences, agents move beyond stateless task-by-task processing [45]. Modern architectures employ both short-term context windows and long-term non-parametric stores via RAG [45]; Figure 2 situates

memory as a cross-cutting concern that spans the perception, reasoning, planning, and action layers of an agent rather than residing in any single module. How this layered organization relates to the human memory taxonomy of cognitive science is taken up alongside Table 3 in Section 4.2.

2.2.5 Planning

Planning is the process by which an AI agent determines a sequence of actions to achieve a specific goal, separating deliberative agents from purely reactive ones by anticipating future states and generating structured action plans before execution [44]. It is essential for multistep decision-making, optimization, and adaptability, often involving task decomposition into smaller, manageable steps based on available tools [44].

2.2.6 Feedback Loops

Agentic systems rely on feedback loops to learn and adapt: they act on their environment, observe outcomes, and use that feedback to refine their algorithms and improve performance [46]. Reinforcement learning provides a canonical feedback-driven mechanism where agents learn policies via trial and error to maximize cumulative reward [46]. Landmark applications such as AlphaGo's self-play illustrate how experience-based adaptation enables robust performance in dynamic environments [47]. No existing paradigm delivers all of these properties at once. That is why the field's history reads as a sequence of partial solutions, and Section 2.3 examines the four most influential of them.

2.3 Relation to Existing Concepts

This section situates agentic AI within the broader landscape of existing AI paradigms. Many prior systems exhibit aspects of agency either implicitly or explicitly through frameworks such as RL, Symbolic AI, Belief-Desire-Intention (BDI) architectures, Embodied Cognition models, and open-ended learning. Understanding these connections clarifies what is new and distinct about current discussions of agentic AI, particularly in the context of LLM-based and hybrid agents. Rather than representing a complete departure from established paradigms, contemporary agentic AI can be understood as an evolutionary synthesis that addresses fundamental limitations of earlier approaches while leveraging their core insights.

2.3.1 Reinforcement Learning Agents

Reinforcement Learning provides a mathematically principled framework for agents to learn through environmental interaction, where agents must balance exploitation of known rewarding actions with exploration of novel strategies to maximize cumulative rewards [48]. The RL paradigm has produced remarkable successes in domains ranging from game playing [47] to robotics [49], establishing many foundational concepts that persist in modern agentic systems.

Contribution to Agency

RL agents embody several core components of agency in a mathematically rigorous manner. They are explicitly goal-directed through the optimization of cumulative rewards, with the reward signal serving as a formalization of objectives that guides learning and decision-making [48]. This goal-directedness is complemented by genuine autonomy–once trained, RL agents operate without human intervention, making independent decisions based on their learned policies and environmental observations [46]. The temporal difference learning mechanisms inherent in RL create natural feedback loops that enable continuous adaptation and improvement [50]. Furthermore, policy gradient methods and actor-critic architectures demonstrate sophisticated forms of planning and deliberation, particularly when combined with model-based approaches that maintain internal representations of environmental dynamics [51, 52].

The success of RL in complex domains has established important precedents for agentic behavior. Deep Q-Networks (DQN) demonstrated that neural agents could master Atari games through pure trial-and-error learning [46], while AlphaGo and its successors showed how RL could be combined with tree search to achieve superhuman performance in strategic reasoning tasks [47, 52]. These achievements validate the core premise that autonomous agents can acquire sophisticated capabilities through environmental interaction alone.

Limitations and Constraints

Despite these successes, traditional RL agents exhibit several fundamental limitations that constrain their agentic capabilities. Most critically, they typically lack rich, persistent memory beyond limited replay buffers, preventing them from maintaining long-term context or learning from extended interaction histories [46, 53]. This memory limitation severely constrains their ability to engage in complex, multi-step reasoning or to adapt their behavior based on accumulated experience over extended periods.

RL agents are also frequently reactive rather than proactive, responding to immediate environmental stimuli without engaging in sophisticated long-term planning unless explicitly augmented with model-based components [53, 54]. The sample efficiency problem–requiring enormous amounts of environmental interaction to learn even relatively simple behaviors–further limits their practical deployment in real-world scenarios where extensive trial-and-error learning is infeasible [55]. Additionally, the reward specification problem presents fundamental challenges: designing reward functions that accurately capture complex objectives without introducing unintended behaviors or reward hacking remains an ongoing challenge [56].

Evolution Toward Modern Agentic AI

Contemporary agentic AI systems address many of these limitations by extending RL paradigms with more flexible architectures and capabilities. Modern agents incorporate sophisticated memory mechanisms, including episodic memory systems that maintain detailed records of past experiences and semantic memory that captures abstract knowledge and skills [57, 58]. They integrate planning capabilities that extend far beyond traditional model-based RL, often incorporating natural language reasoning and hierarchical goal decomposition [59]. The lineage of that decomposition runs

through the options framework, which formalized temporal abstraction within RL itself by letting agents plan over extended sub-policies rather than primitive actions [60], and through hierarchical reinforcement learning more broadly [61]. Recent architectures close the circle by placing an LLM at the top of such a hierarchy as the subgoal setter for lower-level learned policies [62], a design we return to in Section 8.3. Perhaps most significantly, they move beyond pure reward optimization to pursue more complex, contextually-defined objectives that can be specified through natural language instructions or demonstrated through examples [63].

2.3.2 Symbolic Agents

Symbolic AI, often referred to as "Good Old-Fashioned AI" (GOFAI), represents the earliest systematic attempts to create intelligent agents through explicit representation and manipulation of knowledge using symbols, logical rules, and formal reasoning systems [64, 65]. This paradigm dominated AI research for several decades and established many foundational concepts about knowledge representation, inference, and goal-directed behavior that continue to influence modern agentic systems.

Contribution to Agency

Symbolic agents excel in areas where modern neural approaches still struggle, particularly in explicit reasoning, logical inference, and interpretable decision-making. Their knowledge bases provide rich, structured representations of domain knowledge that can support sophisticated planning algorithms and causal reasoning [64]. Classical planning systems like STRIPS and its successors demonstrate remarkable capabilities in decomposing complex goals into sequences of primitive actions, exhibiting a form of hierarchical reasoning that remains challenging for purely neural approaches [66, 67].

The explicit representation of beliefs, goals, and plans in symbolic systems provides transparency and interpretability that is often lacking in neural agents. This interpretability enables debugging, verification, and human oversight of agent behavior in ways that remain difficult with black-box neural systems [68]. Furthermore, symbolic agents can engage in sophisticated forms of meta-reasoning, explicitly reasoning about their own knowledge, uncertainty, and reasoning processes [69].

Expert systems demonstrated how symbolic agents could capture and operationalize human expertise in specialized domains, achieving performance levels that competed with human experts in areas like medical diagnosis and geological analysis [70, 71]. These successes established important precedents for how agents could augment human capabilities through systematic knowledge representation and automated reasoning.

Limitations and Brittleness

The limitations of symbolic approaches became increasingly apparent as researchers attempted to scale them to more complex, open-world domains. The knowledge acquisition bottleneck–the difficulty of manually encoding comprehensive domain knowledge–proved to be a fundamental constraint on the scalability of symbolic systems [72]. Symbolic agents typically exhibit brittleness when encountering situations

not explicitly anticipated by their designers, lacking the graceful degradation and adaptability that characterizes robust intelligence [64].

The frame problem, first articulated by McCarthy and Hayes [73], highlights deep challenges in representing and reasoning about change in dynamic environments. Symbolic systems struggle with the computational complexity of maintaining consistent world models as conditions change, particularly in environments where many aspects of the state are irrelevant to current goals but must nonetheless be tracked for consistency [74]. Additionally, symbolic approaches have historically struggled with uncertainty, noise, and incomplete information–conditions that are ubiquitous in real-world environments [75].

The grounding problem–connecting abstract symbols to perceptual experience and real-world referents–represents another fundamental limitation of pure symbolic approaches. Without mechanisms for learning symbolic representations from experience, symbolic agents remain dependent on human designers to provide appropriate abstractions and cannot adapt their representational schemes to novel domains [40]. Brittleness of this kind has also been attacked from the learning side rather than the representation side. The open-ended learning literature argues that agents in unbounded environments cannot rely on designer-supplied ontologies at all, and must instead generate their own goals, skills, and representations through intrinsically motivated exploration [27, 76], a line developed most fully in developmental robotics [77]. We return to it in Sections 3.3 and 8.5, since it bears directly on both open-world operation and data efficiency.

Integration with Modern Approaches

Rather than abandoning symbolic approaches entirely, contemporary agentic AI increasingly pursues neuro-symbolic integration that combines the learning capabilities of neural systems with the reasoning strengths of symbolic approaches [78, 79]. Large language models demonstrate an intriguing form of implicit symbolic reasoning, capable of manipulating abstract concepts and engaging in logical inference despite their neural substrate [80, 81]. This has led to hybrid architectures that use LLMs to bridge between perception and symbolic reasoning, or that employ symbolic components to structure and verify neural reasoning processes [82].

Modern program synthesis techniques exemplify this integration, using neural networks to guide the search for symbolic programs while maintaining the interpretability and verifiability of symbolic representations [83]. Not all of this integration runs through language models. Intrinsically motivated cognitive architectures such as H-GRAIL pursue the same end, robust operation in open domains, by coupling motivational systems to learned skill repertoires without any LLM in the loop [77, 84], and neurally guided program synthesis [83] likewise grounds symbolic structure in learning rather than in prompting. These approaches point toward combining the complementary strengths of the two paradigms. We do not claim that conclusion as new: it is the organizing thesis of the neuro-symbolic literature [78, 85] and, approached from the side of lifelong autonomy, of work on open-ended learning through cognitive architectures [76, 77]. What the present moment adds is a substrate, the LLM, that makes

such combinations cheap to prototype at scale; Section 8.6 asks what it would take to make them principled rather than merely convenient.

2.3.3 Belief-Desire-Intention Architectures

The BDI model, developed by Rao and Georgeff [86], provides a formal framework for understanding goal-directed behavior in terms of three key mental attitudes: beliefs (representing the agent's informational state about the world), desires (representing the agent's objectives or goals), and intentions (representing the agent's commitments to particular courses of action). This framework has been highly influential in multi-agent systems and provides a conceptual foundation for understanding deliberative agency.

Contribution to Agency

The BDI architecture provides a principled separation between different aspects of agency that enables sophisticated deliberative behavior. The distinction between desires and intentions captures the important difference between having goals and being committed to pursuing them, allowing agents to maintain multiple potential objectives while focusing computational resources on committed plans [86]. This separation enables agents to engage in practical reasoning–the process of deciding what to do based on their beliefs about the world and their desires about how it should be changed.

BDI agents demonstrate sophisticated goal management capabilities, including goal adoption, goal revision, and goal prioritization in response to changing circumstances. The intention-based commitment strategies allow agents to persist in pursuing goals despite temporary obstacles while remaining flexible enough to abandon commitments when circumstances change fundamentally [87]. The explicit representation of plans as structured sequences of actions enables hierarchical reasoning and supports plan repair and adaptation when execution encounters unexpected conditions.

The BDI framework has proven particularly valuable in multi-agent systems, where agents must coordinate their beliefs, goals, and plans with other agents. BDI-based coordination mechanisms enable agents to engage in joint planning, negotiate conflicting objectives, and maintain coherent shared mental models of collaborative tasks [88].

Limitations and Implementation Challenges

Traditional BDI implementations face several significant challenges that limit their applicability to complex, real-world domains. The original BDI models provide limited mechanisms for handling uncertainty in beliefs, making them brittle in environments where information is incomplete or unreliable [89]. The assumption of discrete, symbolic representations of beliefs and goals makes it difficult to integrate BDI architectures with modern perception and learning systems that operate on continuous, high-dimensional representations.

Scalability represents another significant challenge. As the number of beliefs, desires, and intentions grows, the computational complexity of practical reasoning can become prohibitive. The need to maintain consistency across large sets of beliefs

and to reason about complex interdependencies between goals and plans creates computational bottlenecks that limit the applicability of BDI approaches to large-scale problems [90].

The plan library approach used in many BDI implementations–where agents select from pre-specified plans based on their current beliefs and intentions–limits adaptability and requires extensive domain-specific engineering. This approach struggles in novel environments where appropriate plans may not be available in the library, and it provides limited mechanisms for learning new plans from experience.

Modern Manifestations in LLM-Based Agents

Contemporary LLM-based agents demonstrate interesting parallels to BDI architectures, even when not explicitly designed around BDI principles. The ReAct framework [91] embodies BDI-like reasoning patterns by maintaining beliefs about the current state (through observations), pursuing desires (specified goals), and forming intentions (action plans) in an iterative cycle. AutoGPT[2] and similar autonomous agents implement forms of goal management and plan execution that echo BDI deliberation cycles, though using natural language representations rather than formal logical structures.

The Voyager agent [92] demonstrates particularly sophisticated goal-directed behavior that resembles BDI planning, maintaining explicit skill libraries (analogous to plan libraries) and engaging in hierarchical goal decomposition. These systems suggest that the core insights of BDI architectures remain relevant, even as the implementation substrate shifts from symbolic logic to neural language models.

Modern prompt engineering techniques often implicitly structure LLM reasoning according to BDI-like patterns, encouraging models to explicitly represent their beliefs about the situation, clarify their objectives, and outline their intended approach before taking action. This suggests that the BDI framework captures important aspects of deliberative reasoning that transcend particular implementation approaches.

2.3.4 Embodied Cognition and Situated Intelligence

The embodied cognition paradigm, championed by researchers like Brooks [39] and Varela et al. [93], emphasizes that intelligence emerges from the dynamic interaction between an agent's physical body, its environment, and its control systems. This perspective argues that cognition is fundamentally grounded in sensorimotor experience and that intelligent behavior requires real-time, closed-loop coupling with the physical world rather than abstract, disembodied reasoning.

Contribution to Agency

Embodied approaches have contributed crucial insights about the nature of adaptive, situated intelligence that inform modern understanding of agency. The emphasis on real-time interaction highlights how intelligent agents must continuously adapt their behavior based on immediate environmental feedback, rather than relying solely on pre-computed plans or abstract reasoning [39]. This perspective reveals how seemingly complex behaviors can emerge from the interaction between simple control

[2] https://github.com/Significant-Gravitas/AutoGPT

mechanisms and rich environmental structure, demonstrating that intelligence can be distributed across agent-environment systems rather than localized within the agent's computational architecture.

The subsumption architecture developed by Brooks [94] demonstrated how robust, adaptive behavior could emerge from layered reactive systems without requiring explicit symbolic representation or centralized planning. This work established important principles about behavioral robustness, graceful degradation, and the importance of tight coupling between perception and action that continue to influence robotics and autonomous systems design.

Embodied cognition research has also emphasized the importance of developmental processes and experiential learning in shaping intelligent behavior. The idea that cognitive capabilities emerge through accumulated sensorimotor experience provides insights into how agents can acquire increasingly sophisticated behavioral repertoires through environmental interaction [95, 96].

Integration with Modern Cognitive Architectures

While early embodied cognition research often positioned itself in opposition to symbolic AI and abstract reasoning, contemporary research has moved toward more integrated approaches that combine embodied interaction with sophisticated cognitive capabilities. Modern robotics systems demonstrate how embodied agents can maintain rich internal models and engage in complex planning while remaining grounded in continuous sensorimotor experience [97].

The integration of large language models with robotic systems represents a particularly significant development that bridges embodied and abstract cognition. Projects like RT-1 [98] and PaLM-SayCan [99] demonstrate how language models can provide high-level reasoning and planning capabilities while remaining grounded in perceptual experience through robotic embodiment. These systems suggest that the opposition between embodied and symbolic approaches may be a false dichotomy, with the most capable agents requiring both grounded sensorimotor experience and abstract reasoning capabilities.

Recent work on multimodal foundation models extends embodied principles to digital environments, creating agents that can perceive and act across multiple modalities while maintaining coherent behavior [100]. These developments suggest that embodied cognition principles—particularly the emphasis on closed-loop interaction and environmental grounding—remain relevant even as the notion of "embodiment" expands beyond physical robotics to include various forms of environmental interaction.

Synthesis and Future Directions

The evolution from isolated paradigms to integrated approaches represents the current frontier in agentic AI research. Modern systems increasingly combine RL-style learning and optimization with symbolic reasoning capabilities, BDI-like deliberative architectures, and embodied interaction principles. Large language models serve as a particularly important integration point, providing a common representational substrate that can support symbolic reasoning, goal-directed planning, and natural language interaction while being trainable through methods inspired by RL principles.

This synthesis suggests that rather than replacing earlier paradigms, contemporary agentic AI represents their culmination–addressing the limitations of each approach while preserving their core insights. The challenge for future research lies in developing principled frameworks for this integration that maintain the benefits of each paradigm while avoiding their individual limitations.

2.4 Historical Perspective

Agentic AI does not emerge from a vacuum. The notion of computational agents capable of autonomous behavior has evolved over decades, shaped by key milestones in AI history. This section traces the evolution from the earliest symbolic reasoning systems to the current wave of LLM-enabled agents, highlighting the continuity of core ideas and the critical breakpoints that defined each new era.

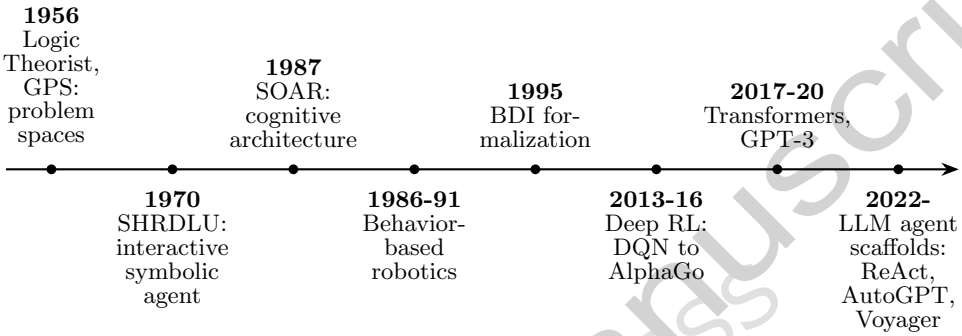

Fig. 3 Timeline of agent paradigms from 1956 to the present. Each milestone marks the arrival of a capability that later agent generations inherited rather than replaced.

2.4.1 Foundations and Evolution of Agent-Based Systems (1950s–2000s)

Figure 3 summarizes the trajectory this section traces. If your page budget genuinely forbids the figure, the fallback is to split 2.4.1 into four bolded run-in eras with explicit date ranges, but the figure is the stronger answer to the review. The intellectual roots of agentic AI lie in early efforts to create systems that could reason and solve problems in a human-like way. The foundational era began with systems like Logic Theorist and the General Problem Solver (1955–1957), which established the Problem Space Hypothesis—a theory that goal-oriented behavior can be understood as selecting operators within a defined problem space [101]. This groundbreaking work laid the foundation for encoding reasoning as a formal computational process, establishing principles that would guide agent design for decades to come.

Building on these theoretical foundations, SHRDLU (1968–1970) emerged as a landmark interactive agent that operated in a simulated block world. Unlike its predecessors, SHRDLU could process natural-language commands, manipulate virtual

objects, and maintain conversational context, demonstrating that symbolic agents could exhibit rich, context-aware behavior [102]. This system showed that agents could not only reason about their environment but also engage in meaningful dialogue about their actions and intentions, presaging modern conversational AI agents.

The 1980s witnessed the development of comprehensive cognitive architectures, most notably SOAR, developed by Laird, Rosenbloom, and Newell. SOAR represented a significant advancement in modeling general human-like intelligence, pursuing goals through symbolic reasoning while incorporating chunking mechanisms for procedural learning [103]. This architecture demonstrated that agents could learn from experience and improve their performance over time, moving beyond purely reactive systems to ones capable of adaptation and growth.

As the limitations of purely symbolic systems became apparent—particularly their brittleness and inability to generalize to novel situations—the field underwent several paradigm shifts. The introduction of Reinforcement Learning (RL) fundamentally changed how researchers conceptualized agent learning, allowing agents to discover optimal behaviors directly from interaction with their environment rather than relying solely on pre-programmed symbolic rules [48]. This shift represented a move from top-down knowledge engineering to bottom-up learning from experience.

Concurrently, the behavior-based robotics movement, pioneered by Brooks and others, challenged the prevailing assumption that complex internal world models were necessary for intelligent behavior. These researchers emphasized embodied, reactive agents that coupled sensing and action directly, demonstrating that sophisticated behaviors could emerge from simple, reactive rules without explicit symbolic reasoning [39]. This approach proved particularly effective for real-world robotic applications where rapid response to environmental changes was crucial.

The formalization of agent architectures continued with the development of Belief-Desire-Intention (BDI) frameworks, which provided a structured approach to modeling intentional agents. These frameworks found particular success in applications requiring explicit goal management and planning, such as autonomous drones and game AI [86]. Meanwhile, research in multi-agent systems explored how multiple agents could coordinate, negotiate, and exhibit social agency, laying the groundwork for modern cooperative AI systems [38]. These developments recognized that many real-world problems required not just individual intelligence but collective coordination and social reasoning.

2.4.2 The LLM Revolution and Modern Agentic Synthesis (2020s–Present)

The advent of Large Language Models marks a revolutionary breakpoint in the evolution of agentic AI. The introduction of Transformer architectures enabled flexible language understanding and emergent reasoning capabilities at an unprecedented scale [104]. Models like GPT-3 demonstrated few-shot generalization abilities that seemed to capture aspects of human-like adaptability, showing that sufficient scale and training could produce systems with broad, general-purpose capabilities [80].

Modern agentic frameworks have emerged that use sophisticated scaffolding to transform LLMs into true autonomous agents. AutoGPT pioneered the concept of

self-directed task decomposition, autonomously breaking down high-level goals into manageable sub-tasks without human intervention. BabyAGI implemented iterative loops for task generation and prioritization, demonstrating how LLMs could engage in metacognitive planning. More specialized systems like Voyager showed how LLM-based agents could learn complex skills through continual exploration in challenging environments like Minecraft [92], while PaLM-E demonstrated the integration of vision, language, and action for embodied robotic manipulation tasks [105].

These modern systems approximate key agentic properties through innovative techniques that combine the strengths of LLMs with structured reasoning processes. Autonomy is achieved through self-looping task execution and iterative prompting strategies [81]. Goal-directedness emerges from sophisticated prompt conditioning and planning scaffolds that guide the LLM's outputs toward specific objectives. Memory capabilities are implemented through vector stores and retrieval-augmented generation, allowing agents to maintain and access long-term information [45]. Planning abilities are enhanced through Chain-of-Thought and Tree-of-Thought reasoning, which encourage systematic problem decomposition [81]. Feedback loops enable self-critique and integration of external API responses, creating agents that can learn from their mistakes and adapt their strategies.

The current synthesis in agentic AI represents a convergence of these historical strands into cohesive architectures that leverage the best of each paradigm. Neuro-symbolic planning integrates explicit logical reasoning with neural policy learners, combining the interpretability and correctness guarantees of symbolic systems with the flexibility and learning capabilities of neural approaches [79]. Continual and safe learning pipelines alternate between closed-world simulations for safe exploration and open-world fine-tuning under safety constraints, addressing the critical challenge of deploying learning agents in real-world environments [106]. LLM-scaffolded agents use sophisticated prompting techniques, tool-use APIs, and feedback loops to approximate deliberation and reflection, creating systems that can reason about their own reasoning [91].

The evolution of agentic AI reveals a pattern of paradigm shifts driven by fundamental methodological innovations and expanding computational capabilities. The transition from symbolic to embodied systems in the 1990s, catalyzed by Brooks's behavior-based robotics [39], established embodiment as a core principle of intelligence. The shift from symbolic to learning-based approaches through the resurgence of Reinforcement Learning provided agents with adaptive capabilities [48]. Most recently, the emergence of LLM agents has endowed systems with broad world knowledge, few-shot adaptability, and sophisticated reasoning capabilities at scale [81, 104].

Looking forward, the long-term goal remains the unification of symbolic guarantees of correctness, RL's adaptive learning capabilities, embodiment's situated intelligence, and LLMs' world knowledge and reasoning abilities. Achieving this synthesis will require continued advances in hybrid architectures, safety-centered design principles, and standardized benchmarks that measure true agentic competence in open, dynamic environments. The trajectory of agentic AI suggests that future breakthroughs will come not from any single approach but from increasingly sophisticated integrations that leverage the complementary strengths of different paradigms.

3 Taxonomy of Agentic AI Systems

Categorization of agentic AI systems can be approached along several dimensions, the most fundamental of which are *level of autonomy*, *cognitive capability*, and *environment interaction*.

3.1 Level of Autonomy

A key dimension for classifying agentic AI systems is their *level of autonomy*–the degree to which they can operate and make decisions independently of external human control. This axis allows us to differentiate simple tool-like agents from complex, self-directed systems. In the AI context, autonomy refers to an agent's ability to set, pursue, and adapt its own actions with minimal human intervention and to respond to dynamic environments without explicit step-by-step direction [64]. These varying levels of autonomy are made possible by different underlying agent architectures:

- **Reactive Architectures** are characterized by stimulus–response mechanisms without internal state maintenance. These systems employ direct perception–action loops, excelling in dynamic environments that require immediate responses (e.g., obstacle-avoidance robotics, simple chatbots). Their primary limitation is an inability to handle novel situations due to a lack of strategic planning [39].
- **Deliberative Architectures** enable more advanced behavior by maintaining complex internal world models. This allows them to engage in means–end reasoning and multi-step planning to achieve goals. Examples include autonomous vehicles performing route optimization or systems using game-theoretic decision-making [64].
- **Hybrid Architectures** combine reactive and deliberative components to gain the benefits of both. For example, a warehouse robot might use reactive obstacle avoidance for safety while using deliberative path planning for efficiency. This reactive-deliberative layering has a long pedigree in robotics, where three-layer architectures separating control, sequencing, and deliberation became the standard resolution of the tension between responsiveness and foresight [94, 107]. A recent review of autonomy in robotics through cognitive architectures reaches a complementary conclusion from the lifelong-learning side: sustained autonomy in open environments requires architectures that integrate motivation, memory, and learning, rather than any single planning mechanism [77]. These architectural foundations give rise to the following taxonomy of autonomy levels.

The complete taxonomy of autonomy levels is presented in Table 1, with Figure 4 providing a visual ladder representation. The six-level scheme in Table 1 is our own construction, though it deliberately inherits the logic of graded-autonomy frameworks that precede it. Its closest relatives are the SAE J3016 levels of driving automation [108], the autonomy dimension of the Levels of AGI framework [109], and the agent-level scale of Mitchell et al. [110], all of which grade systems by how much decision authority is ceded to the machine. Our formulation differs in two design choices. Levels here are defined by the locus of decision authority and the oversight mechanism attached to it, approval gating at L2, monitored override at L3, bounded independence at L4, because these correspond to distinct engineering controls rather than to

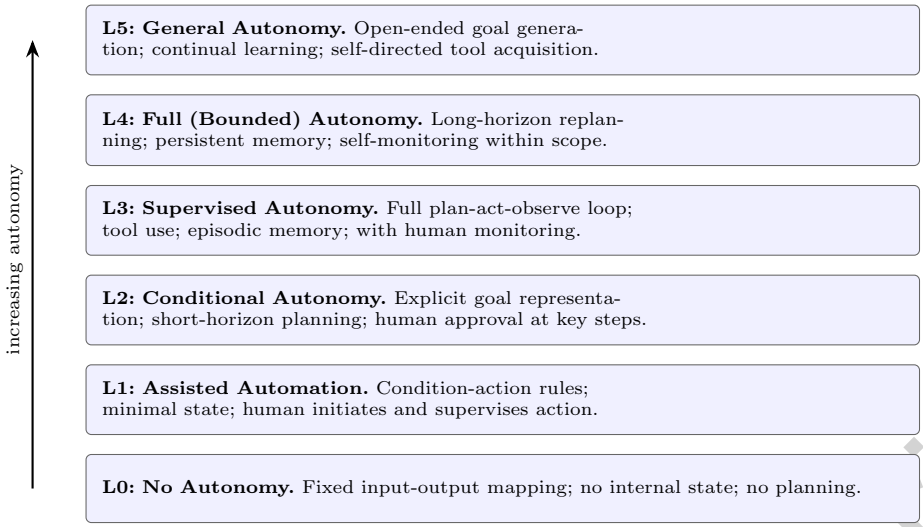

Fig. 4 Autonomy ladder for the levels of Table 1, ascending from L0 (no autonomy) to L5 (general autonomy). Each level lists the core functional components an agent must add to reach it; levels are cumulative, so each level presupposes the components below it.

application domains. And each level is paired with the architectural basis that makes it attainable, which domain-specific schemes such as J3016 do not attempt. Where our levels overlap with prior schemes, this reflects convergence on a natural structure rather than adaptation of any single source.

Most current LLM-based agents like AutoGPT and Voyager (see Table 5 for detailed comparison) operate at Level 2 or 3, showing increasing capability but remaining constrained by reliability and supervision needs. Embodied agents in robotics show progress toward Level 4 but typically only in narrow domains. As such, Level 5 autonomy remains a theoretical aspiration, as no current agent can reliably demonstrate open-ended, self-directed autonomy with the required safety. It is crucial to remember that autonomy is a necessary but not sufficient condition for agency; it must be paired with other properties like goal-directedness and planning to constitute a truly agentic system.

3.2 Cognitive Capabilities

Recent AI research converges on three core cognitive capabilities–planning, memory, and learning–that together shape how agentic a system can become. These capabilities help distinguish between simple reactive agents and more sophisticated deliberative and reflective ones. Cutting-edge language-and-tool agents combine sophisticated plan-generation, long-horizon recall, and self-improvement loops. Cognitive capability is orthogonal to autonomy; an agent can be highly autonomous but cognitively simple, or highly sophisticated but require a human-in-the-loop. Perception is deliberately not a fourth axis in this taxonomy, and the choice deserves an explicit defense, since one could reasonably argue for the alternative. Our reasons are two. First, as Section 2.2.1

Level	Description	Architectural Basis	Example Systems
L0: No Autonomy	The system acts only in direct response to human commands, with no decision-making.	Non-agentic; purely tool-based execution.	Search engines, calculators.
L1: Assisted Automation	The agent performs predefined actions with limited conditional logic while a human remains in control.	Simple reactive logic (if–then rules).	Smart home routines, autocomplete.
L2: Conditional Autonomy	The agent executes action sequences but requires human approval at key steps.	Rudimentary deliberative planning, with a human-in-the-loop for validation.	AutoGPT with Y/N gating, human-in-the-loop RL agents.
L3: Supervised Autonomy	The agent autonomously selects and executes actions but operates under human monitoring with override capabilities.	Deliberative or hybrid systems operating in complex environments where human fallback is a safety feature.	Self-driving cars with human fallback, enterprise AI agents.
L4: Full(Bounded) Autonomy	The agent operates independently within a predefined scope, pursuing goals and adapting without human involvement.	Robust deliberative or hybrid architectures designed for a specific, constrained domain.	Industrial process controllers, some robotic systems.
L5: General Autonomy	The agent operates independently in open-ended environments, setting and pursuing its own goals.	Advanced, general deliberative architecture with robust safety and alignment.	Advanced LLM research prototypes only.

Table 1 Taxonomy of Autonomy Levels

argues, perception is the precondition on which planning, memory, and learning operate: an agent cannot plan over states it cannot represent, store experiences it never registered, or learn from feedback it cannot sense. Grading it in parallel with the capabilities it enables would misstate that dependency. Second, the taxonomy already carries the perceptual channel at a higher level, since modality is an explicit dimension of the system comparison in Table 5, and a perception axis here would double-count it.

That said, perceptual grounding does vary in depth, and the variation is worth stating rather than leaving implicit in the modality column. Agents ascend from static ingestion of text or retrieved documents, through passive multimodal fusion of vision, language, and audio encoders, to online state estimation that maintains a world model from noisy and partial observations, the canonical case being SLAM [17], and finally to active perception, in which the agent directs its own sense to resolve ambiguity [111]. This depth acts as a ceiling on the other three axes. A Tier 5 planner grounded only in static text can plan fluently about a world it cannot verify, which is one root of the hallucination failures discussed in Section 6.3. We therefore treat perceptual depth as a precondition that scales the reachable range of the three cognitive axes, not as a fourth independent axis. Below, we taxonomize each capability on an increasing-capability ladder to guide evaluation and risk analysis.

3.2.1 Planning Capability

Planning measures how far ahead an agent can reason, decompose goals, and choose actions. Table 2 presents a taxonomy of planning tiers, and the ordering is not arbitrary: the tiers are separated by four cumulative criteria, and each tier crosses exactly one new threshold relative to the one below. The thresholds are whether the agent (1) plans at all beyond single-step stimulus-response, (2) searches over future states using an explicit model of action effects, (3) revises its plan during execution in response to observations, and (4) critiques and repairs its own plans without an external trigger. Read this way, Tiers 0 through 3 recover a familiar structure from the planning and control literature: they mirror the reactive, sequencing, and deliberative layers of classical three-layer robot architectures [107], and the classical, hierarchical, and probabilistic families of automated planning [44]. Tiers 4 and 5 are where the taxonomy leaves that lineage behind, since interleaved natural-language reasoning over tool calls [91] and self-reflective plan repair [112] have no counterpart in pre-LLM control stacks; a recent survey organizes exactly this newer territory into task decomposition, plan selection, external planners, reflection, and memory [113]. Canonical systems anchor each tier: deep-Q Atari agents at Tier 0 [46], STRIPS and HTN planners at Tier 2 [66], AlphaGo at Tier 3 [47], ReAct-scaffolded agents such as AutoGPT at Tier 4 [91], and Reflexion at Tier 5 [112]. Modern agents increasingly move beyond simple reward maximization by explicitly generating and revising plans, often in natural language, including hierarchical task decomposition (AutoGPT) and automatic curriculum generation for exploratory planning (Voyager).

Table 2 Planning capability tiers. Each tier crosses exactly one new threshold relative to the tier below it (second column).

Planning tier	Distinguishing criterion	Characteristic methods	Illustrative systems
Tier 0: Reactive	No lookahead; direct stimulus-to-action mapping	Model-free policies	Deep-Q Atari agents [46]
Tier 1: Scripted sequencing	Fixed action sequences; no model of action effects	Hard-coded task graphs; IF-THEN rules	Home-automation routines
Tier 2: Symbolic / HTN	Explicit action model; complete plan computed before execution	Total-order and hierarchical-task-network planners [44, 66]	Industrial logistics planners
Tier 3: Search-enhanced RL	Model-based lookahead search during execution	Monte Carlo tree search; model-based rollouts [114]	AlphaGo / AlphaZero [47, 52]
Tier 4: LLM-guided CoT / ReAct	Plan revision interleaved with acting and observing	Chain-of-Thought, Tree-of-Thought, ReAct [81, 91]	AutoGPT, BabyAGI
Tier 5: Self-reflective planning	Self-generated critique and repair of own plans	Reflexion and related refinement loops [112]	Early research prototypes

3.2.2 Memory Capability

Memory scopes how much past context an agent can retain, retrieve, and manipulate. Table 3 presents the different memory capability layers, from working memory through episodic and semantic to continual memory across tasks. This capability ranges from stateless reactive agents to systems with complex human-like memory architectures. Some advanced agents, like PhiloAgents, implement dual memory systems with both short-term working memory and long-term storage for skill retention (corresponding to multiple layers in Table 3). It is worth making explicit how this engineering taxonomy relates to the established cognitive science of memory, since the correspondence is close but not exact. Human memory research distinguishes a sensory register and a capacity-limited working memory [115, 116], long-term declarative memory split into episodic and semantic forms [117], and non-declarative procedural memory for skills [118]. The working, episodic, and semantic layers of Table 3 map directly onto their human counterparts, and the skill libraries of agents such as Voyager [92] are a genuine analogue of procedural memory, which is why we list them as a separate layer. Two divergences are deliberate. Sensory memory has no persistent analogue in current agent stacks; its functional role is absorbed by the perceptual encoders of Section 2.2.1, which transform raw input before anything is stored. And the continual layer is an engineering category concerned with retaining competence in model weights across task distributions [119], closer to consolidation than to a distinct store. Human consolidation gradually transforms episodic traces into semantic knowledge, whereas agent memories typically remain verbatim and perfectly retrievable, which changes the forgetting dynamics considerably. Recent surveys of agent memory adopt a comparable mapping between human memory systems and agent implementations [120].

Memory Layer	Typical Mechanisms	Representative Work
Working	Transformer context window, scratchpad tokens	Standard ChatGPT session
Episodic (Short-term)	In-prompt buffers; JSON logs; reflection notes	Reflexion agents
Episodic (Long-term)	Vector-store + retrieval (RAG); prioritized replay	Voyager in Minecraft
Semantic / Knowledge	External KBs; symbolic fact bases; OS-style paging of memories (MemGPT)	BDI knowledge stores, MemGPT
Procedural / Skill	Skill libraries; cached tool routines; fine-tuned policies	Voyager skill library [92]
Continual Across Tasks	Elastic-Weight-Consolidation [119]; selective replay	–

Table 3 Memory Capability Layers

3.2.3 Learning Capability

Learning captures an agent's capacity to improve over time through gradient updates, episodic rehearsal, meta-reasoning, or reinforcement signals. Table 4 outlines the learning capability stages, from fixed systems with no updates (Stage 0) through to meta-learning agents that learn how to learn (Stage 5). This can manifest as

goal-oriented RL via iterative prompting (e.g., BabyAGI, which employs Stage 3 in-context self-improvement) or embodied learning via visual-language pretraining (e.g., PaLM-E).

Learning Stage	Core Technique	Example & Evidence
Stage 0 – Fixed	No update post-deployment	Static retrieval bots
Stage 1 – Offline Fine-Tuning	Supervised or RL-from-HF	GPT, Claude
Stage 2 – Continual / Lifelong EWC-based systems	Regularization	Rehearsal for memory retention
Stage 3 – In-Context Self-Improvement	Self-reflection or verbal RL (Reflexion)	Reflexion agents
Stage 4 – Autonomous RL Loops	Self-training with synthetic data	AlphaGo self-play; self-improving LLMs
Stage 5 – Meta-Learning	Agents learn how to learn and adapt hyper-strategies	Iterative self-evaluation prototypes

Table 4 Learning Capability Stages

Synthesis Across the Three Axes - Ultimately, planning provides foresight, memory provides persistence, and learning provides adaptivity. Figure 5 visualizes how different agentic systems (AutoGPT, Voyager, BabyAGI, PaLM-E, OWA, and SOAR) compare across these three cognitive capability dimensions. Most production agents today operate at Tier 2–3 planning, use episodic/semantic memory, and employ Stage 1–2 learning. As shown in Figure 5, experimental systems that score high on all three axes approach the upper corner of agentic capability but also introduce significant safety and alignment challenges. The six systems in Figure 5 and Table 5 were selected to anchor the taxonomy, not to survey the field. Together they cover each architectural type at least once, span text, web, desktop GUI, and embodied modalities, include both closed- and open-world deployments, and add one pre-LLM cognitive architecture, SOAR, as a historical baseline. Many other systems can be placed on the same axes, and the benchmark landscape of Section 6.6 covers the field far more broadly.

3.3 Environment Interaction: Closed-World vs. Open-World Settings

Modern agentic AI operates along a spectrum that ranges from *closed-world* environments–fully modeled, static, and exhaustively specified–to *open-world* settings that are dynamic, partially observed, and rich with unmodeled novelty. Classical symbolic and probabilistic planners historically assumed perfect knowledge of all objects and facts, giving rise to the so-called closed-world assumption (CWA) in automated planning and reasoning [121]. Over the last decade, however, robotics and embodied-AI research have shifted toward open-world formulations that confront unknown objects, evolving goals, and stochastic physics, motivating new work in open-set perception, continual learning, and robust reinforcement learning [122, 123]. This subsection surveys the two extremes, explains why most practical deployments now lie somewhere in between, and highlights techniques that bridge the gap.

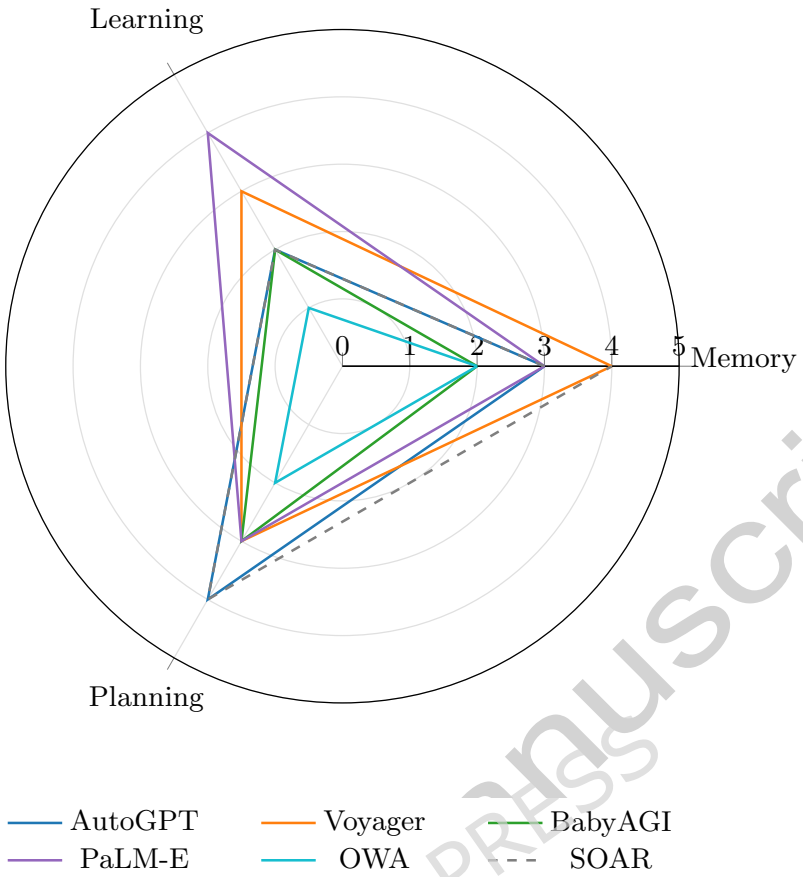

Fig. 5 Qualitative comparison of agent architectures on Planning, Memory, and Learning capability (0 to 5 scale, see Table 5). PaLM-E and Voyager show the strongest Learning scores, while SOAR trails on Planning and Learning despite a competitive Memory score.

3.3.1 Closed-World Interaction

Closed-world agents operate in environments whose state variables, object sets, and transition dynamics are explicitly enumerated at design time, often inside simulations or tightly controlled physical setups. STRIPS-style planners, least-commitment planners, and partial-order planners all exploit CWA to prune search and guarantee completeness in deterministic domains [124, 125]. Extensions such as PSIPOP relax the assumption slightly by permitting *partially* closed worlds–yet they still rely on predefined object universes and deterministic sensing actions [126]. In robotics, CWA simplifies task-and-motion-planning pipelines where the map, manipulable objects, and the robot's kinematics can be encoded once into PDDL; planners then generate provably correct action sequences as long as exogenous disturbances are absent [127]. While these guarantees enable reliable factory automation and warehouse logistics,

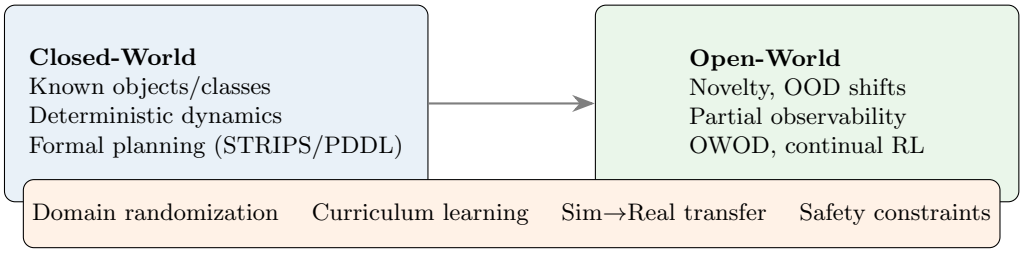

Fig. 6 Contrasting closed- and open-world settings with common bridging techniques.

they falter in the face of unexpected objects or structural changes, producing brittle behavior that can require human intervention or costly re-modeling [121].

3.3.2 Open-World Interaction

Open-world agents drop the exhaustive-knowledge premise and must *detect*, *learn*, and *adapt* online. Perception research tackles recognition through open-world object detection (OWOD), which simultaneously labels known classes and flags novel ones for incremental learning [128]. At the control level, lifelong and continual reinforcement learning frameworks expose agents to non-stationary task streams, pushing algorithms to retain past skills while acquiring new ones without catastrophic forgetting [129, 130]. Surveys of deep RL in real-world robotics highlight recurring challenges such as partial observability, safety constraints, and shifting dynamics [131, 132]. Planning research has responded with knowledge-based and sensing-augmented planners that reason over *incomplete* and *uncertain* world states, abandoning CWA in favor of representations that explicitly track epistemic uncertainty [133]. In fully open worlds the task set itself cannot be enumerated in advance, which is why intrinsically motivated open-ended learning, where agents generate their own goals from novelty, competence, or learning progress, has become central to this setting [27, 28, 76].

3.3.3 Bridging the Gap

Few real deployments are purely closed or purely open; instead, designers increasingly rely on *closed-to-open pipelines* (see Figure 6. Agents train in simulator sandboxes where CWA holds, then transfer into messier reality via domain randomization, curriculum learning, and sim-to-real adaptation techniques illustrated in Figure 6 [106, 134]. Robotics work shows that dynamics-randomized policies can retain task competence after export while remaining plastic enough to adapt online through safe, on-policy updates [106]. Recent planners embed learning components that rewrite or augment symbolic domain models when unexpected objects or effects appear, effectively toggling between closed-world efficiency and open-world flexibility at run-time [121, 133]. Reinforcement-learning studies in highly dynamic arenas–e.g., crowd navigation or agile drone racing–emphasize robust policy improvement under non-stationary transitions and delayed, sparse rewards, echoing open-world desiderata while still leveraging structured priors from closed-world simulations [135, 136].

3.4 Comparative Analysis of Prominent Systems

Table 5 provides a comprehensive comparison of six prominent agentic AI systems across multiple dimensions including autonomy level, cognitive features, modality, environment type, and applications. A terminological note is needed here, because two different properties are easily conflated. The second column of Table 5 records each system's architectural type, a design property fixed by its developers. Autonomy level in the sense of Table 1 is instead an operational property of a deployment: it depends on how the architecture is gated, scoped, and supervised, and the same architecture can be run at different levels. AutoGPT is architecturally hybrid, for instance, yet operates at L2 or L3 depending on whether approval gating is enabled, while a far simpler industrial controller can legitimately occupy L4 within its narrow scope. Architecture constrains the autonomy levels a system can reach; it does not determine the level at which it is operated.

System	Architectural Type	Key Cognitive Features	Modality	Environment	Applications
AutoGPT	Hybrid	Task decomposition, web tool calling	Text/Web	Open-world	Research automation, content generation
Voyager	Deliberative	Skill library, automatic curriculum	Embodied (Minecraft)	Open-world	Game AI, embodied learning
BabyAGI	Deliberative	Iterative prompting, goal chaining	Text	Closed-world	Task management, process automation
PaLM-E	Hybrid	Multimodal fusion, embodied learning	Embodied (robotics)	Both	Manipulation, VQA
OWA	Reactive	Real-time screen processing, plugin system	Desktop GUI	Open-world	Desktop automation, UI testing
SOAR	Deliberative	Symbolic reasoning, problem spaces	Text	Closed-world	Cognitive modeling, expert systems

Table 5 Comparative Analysis of Agentic AI Systems

3.5 Descriptive Taxonomy versus Functional Modeling

The taxonomy developed in this section is a map, and maps are useful only for the journeys they were drawn for. Classifying a system by its autonomy level, planning tier, memory layers, and environment type tells us what components and capabilities it has. That supports comparison across systems, exposes gaps in the design space, and gives the field a shared vocabulary. What it does not tell us is what the deployed system does, in the sense that matters for prediction and risk: which objective the closed loop actually optimizes, which feedback signals it consumes, which environmental variables it can change through its tools, and which failure modes remain stable under repeated interaction and distribution shift. The point echoes Marr's classic observation that a computational system admits several levels of description and that the right one depends on the explanatory goal [137]. A compositional description of a lower limb in terms of tissue, bone, and connective structure can be locally accurate at every point, yet it cannot express what a load-bearing structure is for, nor predict how it fails under a sustained shearing force. The same holds for agents.

The two descriptions come apart in both directions. A system occupying a modest position in our ladders, say L2 conditional autonomy with Tier 4 planning and episodic memory, may nonetheless implement a functionally hazardous loop: persistent goal pursuit through external tools, strategic information gathering, self-reinforcing feedback between its outputs and its future inputs, or optimization against an objective that is underspecified in exactly the situations that matter [56, 138, 139]. Conversely, a system may rank highly on every capability axis while remaining functionally brittle or simply irrelevant to the domain at hand. Taxonomic position is therefore a weak proxy for behavioral risk, and coverage in Tables 1 through 5 implies neither behavioral completeness, nor safety completeness, nor alignment completeness.

When should classification give way to functional analysis? We identify eight conditions. Any one of them weakens the taxonomy as the primary unit of analysis; several in combination make functional modeling mandatory: (i) long-horizon action, where small errors compound; (ii) open-world deployment, where states and objects cannot be enumerated in advance (Section 3.3); (iii) high-impact intervention capacity through tools, APIs, or actuators; (iv) ambiguous or changing objectives; (v) multi-agent interaction with emergent collective dynamics; (vi) limited or delayed human oversight; (vii) self-modification or extension of the agent's own toolchain; and (viii) safety-critical consequences of failure. Under these conditions the operative question is no longer whether the agent "has" planning or memory in some taxonomic sense, but what closed-loop function the deployed system implements.

Functional modeling, as we intend it, means specifying the loop rather than the parts: the objective as actually instantiated, the perception and feedback channels, the action interface and its scope, the invariants that must hold across the loop, and the empirical behavior of that loop over long, tool-using, adversarially perturbed episodes. This is precisely the verification target adopted in Section 7.1 and the measurement gap driving the open challenges of Section 6.7, and the tool interface of Section 4.4 is its natural locus, since that is where the internal state becomes the external action.

4 Technologies Behind Agentic AI Systems

Agentic AI systems are powered by a combination of technologies that allow them to act more like autonomous decision makers than static tools. These systems can understand goals, plan steps, remember what happened before, use tools, and sometimes even interact with the physical world. It is useful to group these technologies by the functional role they play rather than to treat them as a flat list, because they are not all "technologies" in the same sense. We distinguish: *(i)* the **reasoning substrate** (large language models that interpret goals and generate candidate actions); *(ii)* the **memory substrate** (short-term context and long-term retrieval); *(iii)* the **deliberation layer** (planning and decision-making algorithms); *(iv)* the **action interface** (tools, APIs, and actuators through which the agent changes its environment); *(v)* the **integration layer** (development frameworks and interoperability protocols that compose the above into deployable systems); and *(vi)* the **embodiment and modality extension** (perception and action across vision, audio, and the physical world). Of these, (i)–(iv) and (vi) are capability-enabling technologies, whereas (v) is engineering

scaffolding that orchestrates them; we include it because production agentic behavior is in practice a property of the whole stack. Below, we examine each in turn.

4.1 Reasoning Engine

Large Language Models (LLMs) such as GPT-4, Claude, and PaLM serve as the reasoning substrate of many contemporary agentic systems. Acting like the agent's deliberative core, they interpret language, generate candidate sub-tasks, and select among next steps. These LLMs can break larger tasks into smaller steps, reflect on their previous steps, and decide what to do next. In systems like Auto-GPT or BabyAGI, the LLM does not merely answer a question; it iteratively plans and executes multi-step tasks. As shown in Figure 1, these components work together in an integrated workflow in which the LLM reasoning engine coordinates with memory systems, planning modules, and tool interfaces.

This central role must be stated carefully, because taken naively it appears to contradict the characterization of agentic AI as systems that autonomously define their own goals (Section 2.2). A pre-trained LLM is trained on human-generated text and responds to human-supplied prompts; it does not originate its own terminal objectives independently of a user or operator. We resolve the apparent tension by distinguishing two senses of autonomy, following the intrinsic-versus-extrinsic motivation distinction introduced in Section 2.1.1. Current LLM-centered agents exhibit *operational* (instrumental) autonomy: given a delegated, human-specified objective, they autonomously decompose it, decide intermediate sub-goals, choose tools, and recover from failures without step-by-step human direction. They do *not* exhibit *constitutive* autonomy in the strong Kantian or open-ended-learning sense of originating their own ends; that capacity is precisely what distinguishes the aspirational L5 "general autonomy" of Figure 4, which no current system attains. Under this reading there is no contradiction: the LLM is central to the *competence* of present-day agents, their ability to reason and plan in service of delegated goals, while genuine self-origination of goals remains an open, largely unrealized frontier addressed by intrinsically motivated open-ended learning [27, 28]. Prompt-delegated goal-direction is thus best understood not as full autonomy but as a bounded form of it, and we use the L0–L5 ladder rather than a binary autonomous/non-autonomous label precisely to make this gradation explicit. We return to the deeper question of whether such systems *understand* their objectives, as opposed to competently pursuing them, in Section 6.4.

4.2 Memory

For an agent to be effective, it needs memory. As illustrated in Figure 2, memory serves as a cross-cutting concern across all architectural layers. Specifically, agentic AI uses two broad types of memory:

Short-term memory: Short-term memory refers to the information an agent can hold temporarily within its current working context. For language-model-based agents, this is typically the content of the model's token window: the recent conversation, instructions, or steps in a plan. This memory enables agents to track ongoing interactions, follow dialogue threads, and maintain awareness of the immediate task.

While it allows fluid exchanges and simple planning, it is inherently limited by the model's context size, which constrains reasoning over long timelines or multi-step tasks.

Long-term memory: Long-term memory lets agents go beyond their short-term context by storing and retrieving information from external stores. Table 3 details the various memory layers, including episodic (short- and long-term), semantic/knowledge, and continual capabilities. This is often implemented with vector databases and approximate nearest-neighbor (ANN) search: FAISS provides efficient billion-scale similarity search on GPUs [140], while graph-based ANN indices such as HNSW underpin systems like Pinecone and enable low-latency retrieval at scale [141]. Chunks of past data, documents, task history, or user interactions, are stored as embeddings; when the agent needs relevant information, it performs a similarity search to bring the right content back into its working context. This retrieval-augmented generation (RAG) mechanism [45] helps agents act more intelligently over time. More recent designs treat the context window itself as a managed resource: MemGPT, for instance, borrows operating-system paging ideas to move information between a limited context and external storage, giving the agent an explicit memory hierarchy [142]. For example, an agent assisting with legal research can recall documents it previously summarized, enabling continuity and avoiding duplication.

4.3 Planning and Decision-Making

Planning is the technology that converts a goal into an executable sequence of actions and revises that sequence when execution does not go as expected. A typical agentic loop interleaves planning with acting and observing: the agent proposes a plan, executes a step, observes the result, and replans as needed. For example, if a web search fails, the agent can revise the query or try a different approach. This loop, plan, act, observe, replan, enables agents to work on complex, multi-step tasks with minimal human guidance, making them far more flexible and autonomous.

To enable this kind of planning, agentic systems draw on techniques from symbolic, search-based, evolutionary, and learning-based AI. Commonly used methods include:

- **Monte Carlo Tree Search (MCTS):** MCTS incrementally builds a search tree by simulating many possible action sequences and selecting promising ones via the UCT exploration–exploitation rule [114, 143]. It is especially useful in large, uncertain decision spaces such as games and simulations, where exhaustive search is impractical, and it underpins the planning component of systems like AlphaGo [47].
- **PDDL (Planning Domain Definition Language):** PDDL is a formal language for describing planning problems, goals, constraints, and allowed actions, that enables symbolic, abstract reasoning [44]. It is well suited to structured environments such as robotics and workflow automation, where it underpins integrated task-and-motion planning [18], and is increasingly paired with LLMs that translate natural-language goals into PDDL for a classical planner to solve [144].
- **Rolling Horizon Evolutionary Algorithms (RHEA):** RHEA uses evolutionary strategies to evaluate and improve short action sequences over a sliding time

window [145]. It is effective in dynamic settings where the agent must adapt quickly by testing and refining plans on the fly.

- **LLM-based planning:** Chain-of-Thought, Tree-of-Thought, and ReAct-style interleaving of reasoning and tool calls let LLMs generate, branch over, and revise plans expressed in natural language [81, 91], and self-reflective methods add explicit plan critique and repair [112].
- **Hierarchical and option-based methods:** Hierarchical RL and the options framework provide temporal abstraction, letting an agent plan over extended sub-policies rather than primitive actions [60, 61].

These methods allow agents to make deliberate decisions rather than merely reacting to each situation as it arises, and they span the symbolic, search-based, and learning-based traditions surveyed in Section 2.3.

4.4 Using Tools and APIs

A defining advance in agentic AI is the ability to use external tools. Instead of only reasoning, the agent can act: search the web, run a calculator, call an API, or write and execute code. This is realized through two main mechanisms. **Function calling** exposes a set of typed functions that the model can invoke with structured arguments when appropriate, so that the LLM's output is converted into a concrete external action; modern model APIs support this natively. **Plugin systems** give the agent access to a broader, often open-ended catalog of external services and applications. The capacity to learn *when* and *how* to call tools can itself be trained: Toolformer shows that a model can teach itself, in a self-supervised manner, to decide which API to call, when, and with what arguments [146], while the ReAct paradigm interleaves tool calls with reasoning traces so that observations from tools inform subsequent reasoning [91]. Interoperability is increasingly standardized through protocols such as the Model Context Protocol (MCP) [147] and agent-to-agent (A2A) interfaces, which provide a uniform way to connect agents to tools and to one another. Tool use is what makes agents capable of real work rather than advice alone, but it is also the action interface through which an agent changes the world, and therefore, as Section 3.5 argues, the locus at which functional safety analysis matters most.

4.5 Agent Development Frameworks

Beyond individual capabilities like reasoning, memory, and planning, recent progress has produced *development frameworks* that operationalize these primitives into full-fledged agentic systems. Frameworks such as LangGraph [148], AutoGen (AG2) [149], CrewAI [150], the OpenAI Agents SDK [151], Semantic Kernel [152], LlamaIndex [153], Haystack [154], and AgentScope [155] provide abstractions for orchestration, tool use, and multi-agent collaboration. They differ in orchestration primitives (e.g., graphs, handoffs, pipelines), protocol support (notably the Model Context Protocol, MCP [147], and Agent-to-Agent interoperability, A2A), and enterprise integration. Unlike the capability-enabling technologies above, these frameworks are best understood as the integration layer: they do not add new cognitive capabilities so much as

standardize how existing ones are composed, which is what bridges research prototypes and production systems. We summarize the development stack as a metro-map in Figure 7, where MCP and A2A act as interchanges connecting frameworks to downstream applications and to other collaborating agents.

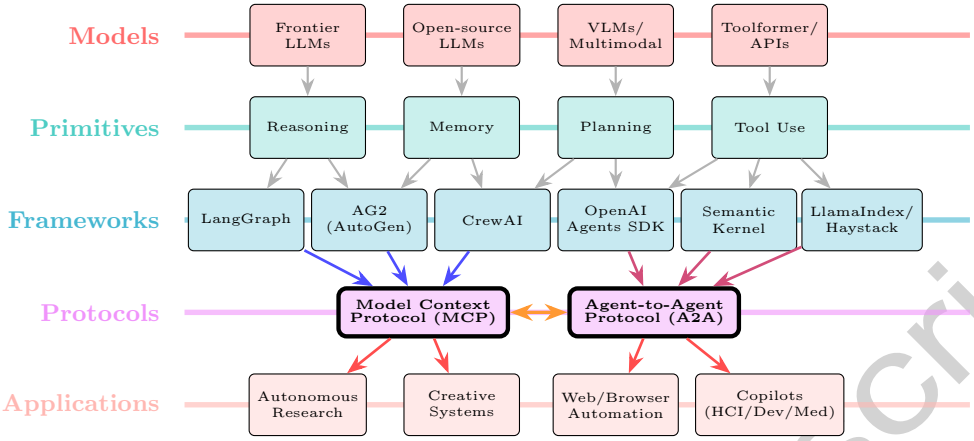

Fig. 7 Metro-map of the agentic AI ecosystem, organized as five stacked layers. The *Models* layer holds the foundation reasoning substrates (frontier and open-source LLMs, vision–language/multimodal models, and tool-augmented models); the *Primitives* layer exposes the core agent capabilities (reasoning, memory, planning, and tool use); the *Frameworks* layer collects development stacks (LangGraph, AutoGen/AG2, CrewAI, the OpenAI Agents SDK, Semantic Kernel, and LlamaIndex/Haystack); the *Protocols* layer contains the Model Context Protocol (MCP) and the Agent-to-Agent (A2A) protocol; and the *Applications* layer lists downstream deployments (autonomous research, creative systems, web/browser automation, and copilots). MCP and A2A are drawn as highlighted *interchange* stations because they connect frameworks both to one another and downstream to applications. Vertical edges trace how capabilities compose upward from models to deployed applications, emphasizing that production agentic behavior is a property of the whole integration stack rather than of any single framework.

4.6 Multimodal and Embodied Agents

Some agents go beyond text: they see, hear, and act in the world. These multimodal and embodied agents combine perception across modalities with physical or simulated action, representing the point at which the reasoning substrate is extended through embodiment. Vision–language models such as PaLM-E inject continuous sensor and image embeddings directly into a language model, so that a single model can reason jointly over text and perception to produce robot-control plans [105]. A rapidly growing line of robotics work grounds language-model reasoning in real manipulation: PaLM-SayCan combines an LLM's task knowledge with learned value functions that estimate which low-level skills are actually feasible in the current physical state, so that the agent proposes only actions it can execute [99]; RT-1 trains a transformer policy on large-scale real-robot demonstrations to map images and instructions to actions [98]; and RT-2 shows that vision–language models pre-trained on web data can be co-fine-tuned as vision–language–action models, transferring semantic knowledge from

the web to robotic control [156]. Underlying all of these is the embodied-perception loop discussed earlier, including SLAM for spatial grounding [17], which lets an agent localize, map, and act in environments that are never fully specified in advance. Such systems open possibilities for AI that can assist in homes, hospitals, warehouses, and other physical settings, while also inheriting the open-world and safety challenges of Sections 3.3 and 3.5.

5 Applications of Agentic AI

Agentic AI has shown distinctive potential across multiple domains by autonomously performing complex roadmaps through multistage reasoning, adapting planning, and fusion of ideal tools. Traditional AI models operate in defined tasks, whereas agentic AI exhibit independence while making crucial decisions for the external environment, which enables the system to tackle real-world problems. In this section, we study the application of agentic AI, which is diverse and rapidly expanding in multiple fields, including research, content generation, web automation, and Human-Computer Interaction (HCI).

5.1 Autonomous Research

The emergence of agentic AI is a game changer in research automation, capable to expand in a way different from that of the traditional retrieval system. The trend of autonomous research is increasing exponentially, leveraging ML and NLP to gather information from multiple sources. Nowadays, the system can compile and analyze information in outstanding manner, enabling researchers to adjust to the current trend. Figure 8 depicts the workflow of autonomous AI agents in the context of user interaction.

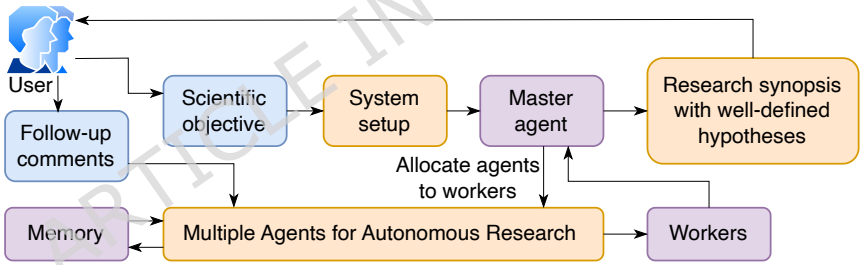

Fig. 8 Multi-agent workflow for autonomous scientific research, from user objectives to hypothesis generation.

- **AlphaCode**: AlphaCode [157] employs an encoder-decoder transformer architecture, wherein the encoder analyzes the natural language problem statement and

the decoder produces the code solution. Generally, for coding problems, the problem descriptions are longer than the solutions. Therefore, the encoder processes about 1536 tokens at a time, whereas the decoder is limited to 768 tokens. Using the shallow encoder and deep decoder, the training efficiency was improved without affecting the problem-solving rate. AlphaCode utilizes multi-query attention [158], which considerably improves sampling performance by sharing key and value heads per attention block. This minimized memory consumption and cache update costs when sampling, facilitating larger batch sizes. For the CodeContests benchmark, the model successfully solved about 34.2% of problems with a maximum of 10 submissions per problem. This indicates that the model predominantly solves simpler problems; however, problems ranked up to 1800 were occasionally solved.

- **SciAgent**: SciAgent [159] is a tool-enhanced language model aimed at improving scientific reasoning by instructing LLMs to utilize external tools instead of building domain-specific models for each scientific area. The foundation of this model is based on a four-module architecture for problem solving: *planning, retrieval, action, and execution*. They constructed the $MathFunc$ corpus based on MATH dataset [160], comprising 31,375 samples and 5,981 documented functions, with every sample structured as a quintuple (q, G_q, F_q, S_q, a_q) that includes a question, planning steps, relevant functions, solution, and answer. A new benchmark, $SciToolBench$, was developed, which calculates the results using five domains. The test results indicate that SciAgent-Mistral-7B surpasses other comparably sized open-source LLMs by over 13% in absolute accuracy on $SciToolBench$, while SciAgent-DeepMath-7B significantly outperforms ChatGPT using the same toolset.
- **AutoDAN:** AutoDAN [161] utilizes a hierarchical genetic algorithm (HGA) tailored for the optimization of structured textual input. This automates the generation of semantically jailbreak prompts, which are capable of bypassing safety protocols in aligned LLMs. AutoDAN used *AdvBench Harmful Behaviors* [162] dataset to evaluate jaibreak attacks. AutoDAN was evaluated against baseline models on multiple LLMs [163–165], where it achieved the highest attack success rates (ASR) with lower perplexity scores. This study illustrates that semantically significant jailbreak prompts are not only more subtle in bypassing defenses but also exhibit greater transferability between models and data.

5.2 Creative Systems

Agentic AI substantially improves creative systems such as game design and narrative development by facilitating dynamic, adaptable, and autonomous content creation. In contrast to conventional generative AI, which generates static outputs, agentic AI coordinates intricate workflows, makes context-sensitive judgments, and adapts based on user interactions. Arif et al. [166] developed an educational tool that employs many AI agents to generate appealing, multimodal storytelling for kids. The system integrates narrative generation, speech synthesis, video production, and music composition to create immersive storytelling engagements. The system is mainly divided into two frameworks, and the model was evaluated on six different models using seven criteria. Agentic AI converted the systems from passive instruments into dynamic collaborators,

enabling extraordinary degrees of personalization, efficiency, and development. Figure 9 illustrates multiple agents capable of working independently on diverse creative tasks.

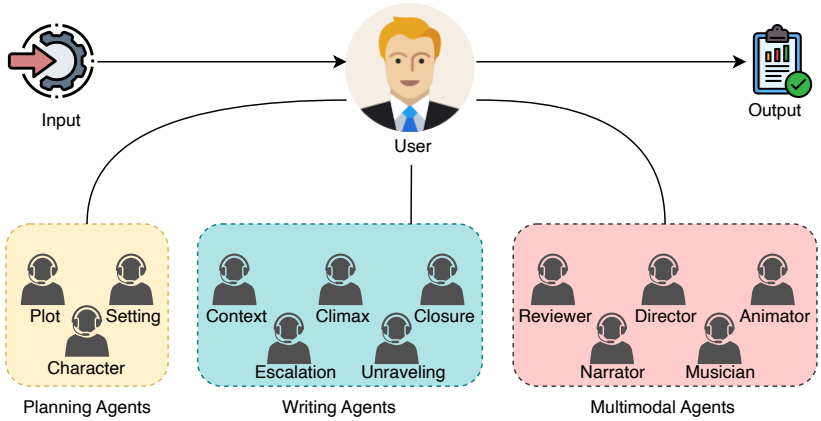

Fig. 9 Multi-agent creative system transforming user input into output through specialized planning, writing, and multimodal agents.

- **Agents' Room:** Huot et al. [167] proposed a framework that addresses the intricate process of narrative composition by separating it into distinct subtasks managed by various agents, which is inspired by writers' rooms in media production. The framework consists of two agents, *planning and writing agents*, which help to develop key components and generate segments of the narrative structure respectively. Similarly, fiction writing has four planning agents and five writing agents that work based on narrative theory. The results were evaluated based on a newly curated dataset, "TELL ME A STORY" which contained both the detailed prompts-stories with tokens ranging from 1,000-2,000 and human-written stories.
- **Voyager:** Voyager [92] is an LLM-powered agent that interacts with GPT-4 via blackbox queries, which embodies a continuous learning agent for Minecraft that indefinitely explores the environment. The system signifies a substantial advancement in autonomous agents capable of progressively learning, enhancing, and generalizing to novel tasks in open-ended settings. It consists of three key components: an *automatic curriculum* to suggest objectives for open-world exploration, a *skill library* that stores the actions which already solved the task, and *iterative prompting mechanism*, a program which keeps improving by running, checking for mistakes, learning from its surroundings, and verifying its own results, all in a loop, until it gets better.
- **PORTAL:** Xu et al. [168] designed a hybrid policy framework that integrates rule-based nodes with neural network units, facilitating both advanced strategic reasoning and meticulous low-level regulation. This framework enables AI agents to

play multiple 3D games through language-directed policy formulation. The system incorporates a dual feedback method that combines quantitative game measurements with vision-language model analysis, which helps to improve iterative policy. Unlike the traditional RL model, which suffers from environmental change, on the other hand, PORTAL overcomes this issue by applying a hierarchical approach that operates at two different levels.

5.3 Web-based Agents

The traditional AI system was mainly prompt-driven, where a code assistant provided single-line completions and search engines responded to user input with multiple lists of links. On the other hand, agentic AI can predict user intent and automatically synthesize data from various sources. The real-time agents. These agents quickly assess large data sets, grasp intricate queries, and modify solutions based on dynamic content available online. Figure 10 shows the tentative process of agents related to assistant tasks.

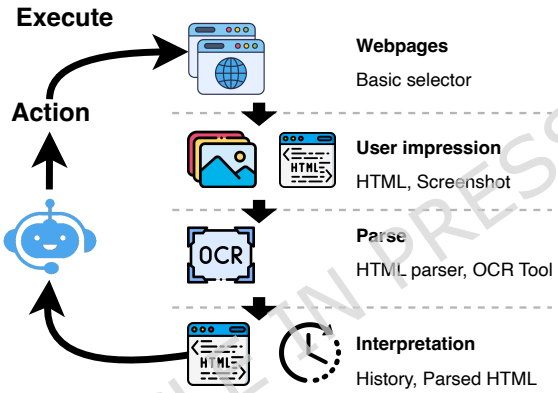

Fig. 10 Web agent pipeline for automated webpage interaction and content extraction.

- **CodeT5+:** CodeT5+ [169] is an encoder-decoder based model operated in three different modes: encoder [170, 171] for understanding tasks, decoder [172, 173] for generative tasks, and encoder-decoder [174, 175] for sequence-to-sequence tasks [176]. CodeT5+ has several variants ranging from 2B to 770B. It was trained using a combination of pretraining tasks, including span denoising, causal language modeling, contrastive learning, and text-code matching, to acquire comprehensive representations from both unimodal code data and bimodal code-text data. The

model was tested on various code understanding and generation tasks using multiple datasets and programming languages.

- **Plan-And-Act:** Erdogan et al. [177] developed an LLM-based agent that can handle complex and multi-step tasks by diving into low and high-level planning. The framework consists of dynamic planning and consists of two agents: *planner* and *executor*. To create training data, the system incorporates three stages of data generation: *Action Trajectory Generation*, *Grounded Plan Generation*, and *Synthetic Plan Expansion*. The framework was evaluated in the *WebArena-Lite benchmark*, which demonstrated improved performance on long-horizon tasks.
- **AutoWebGLM:** Lai et al. [178] developed a framework consisting of four decision-making processes. *State* holds the current page status and window position, *Action* executes operations such as click, scroll, and type, *History* was used to maintain prior states and actions, and lastly, *Policy*, which was the decision-making brain. The system included a pruner algorithm for the efficient transformation of element trees into compact versions. This pruner operates by retaining key features while discarding redundant or noisy components that may obstruct model interpretation. For training, the system had three stages: curriculum learning [179], reinforcement learning [180], and rejection sampling finetuning [181]. Here, the model initially acquires webpage understanding and functionality through curriculum learning. Subsequently, it self-samples training data, extracting insights from its errors. Ultimately, it autonomously operates within its surroundings, becoming a domain expert. The system addresses three major web navigation challenges: data complexity in HTML, functionality of webpage actions, and difficulty in opening multiple unbounded domains.

5.4 Human-AI Collaboration

AI systems can augment human creativity by engaging with human narratives and emotional constructs, which enhances creativity and thoughtfulness. The Human-AI Collaboration paradigm has proven notable benefits across various sectors, including healthcare, manufacturing, and education, enhancing innovation, decision-making quality, and efficiency [182]. The efficacy of Human-AI collaboration depends on elements such as task distribution, transparency of interaction and communication, and the adaptation of both individuals and AI to each other's abilities [183]. Figure 11 demonstrates various dimensions of AI agents with human override situations.

Education is one of the oldest and most studied cases of this paradigm. It appears in the form of Intelligent Tutoring Systems (ITS). These are agents that work one-on-one with a human learner. They keep a model of what the learner knows and adjust their teaching based on it. A key part of this tradition is student modeling through knowledge tracing, where the agent keeps an up-to-date estimate of what its human partner has learned. Recent surveys review this problem from its early Bayesian methods to newer deep-learning methods [184]. The value of ITS as a partner is backed by evidence, not just a claim. Systematic reviews of their use in real classrooms report steady learning gains over regular teaching [185].

LLMs are now being folded into this pipeline, and recent surveys catalog their rapidly growing use as virtual tutors [186]. MWPTutor uses an LLM to fill the state

space of a predefined finite-state tutor and outperforms an unconstrained GPT-4 on overall tutoring quality [187]. Similarly, SocraticLM is fine-tuned toward a Socratic, thought-provoking teaching style rather than answer-giving, exceeding GPT-4 by more than 12% on pedagogical measures [188]. Model skill on its own is not enough, and it must be paired with structure to give reliable agent behavior. Studies of personalized feedback report that about a third of GPT-4's hints are too general, wrong, or give away the answer [189].

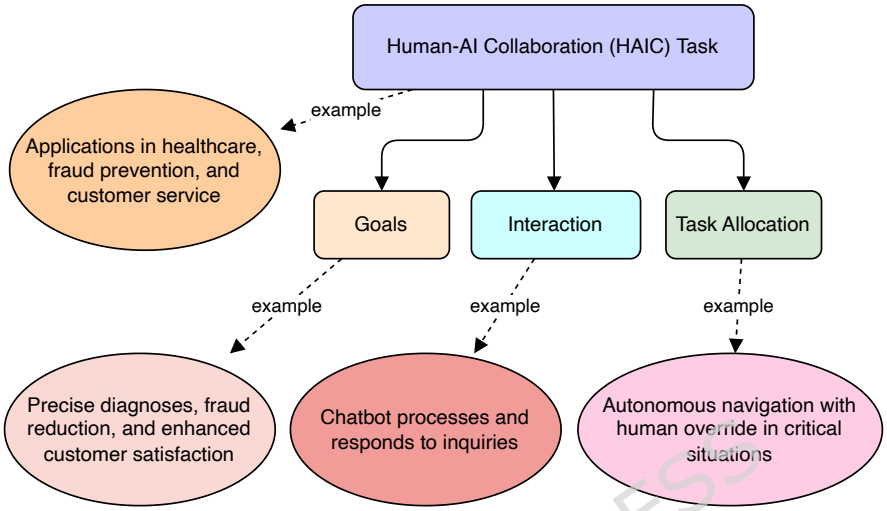

Fig. 11 Human-AI collaboration taxonomy with key dimensions and practical examples from healthcare, fraud detection, and customer service.

- **TutorGym:** Weitekamp et al. [190] constructed an AI agent with 223 domains across three established ITS platforms, which evaluates both tutors and simulated students. Evaluation employs automatically generated "completeness profiles" that encompass all accessible tutor states alongside correct and incorrect student actions, assessing agents on step-level action scoring and demonstration generation instead of solely on final replies. Key findings indicate that current LLMs (GPT-4o, Claude-3.5, DeepSeek-v2.5) exhibit poor performance as tutors, achieving less than 50% accuracy in incorrect action identification and 52-70% in demonstration generation. However, they can generate impressively human-like learning curves when functioning as students via in-context learning, with error rates diminishing from approximately 80% to around 10% over 16 learning opportunities.
- Lyu et al. [191] created teachable AI agents using LLMs and prompt engineering to demonstrate big five personality qualities, which were randomly allocated at the

problem level. They examined 13,466 student-agent interactions involving 411 students through Natural Language Processing (NLP) classification, achieving 95.8% accuracy with LLaMA3-8b to classify interaction behaviors and knowledge presence. They subsequently employed Linear Mixed Models to demonstrate that agent personality traits significantly affected student interaction patterns and the exhibition of procedural knowledge during mathematics tutoring.

Human-AI collaboration works best when tasks are shared clearly, and the AI acts within a defined structure. The open question for the LLM era is whether generative models can become reliable partners rather than tools that still need close human oversight.

6 Challenges and Risks

As agentic AI systems become more autonomous and capable, they introduce a new set of technical, ethical, and societal challenges. These systems operate with a high degree of independence, which makes them powerful but also unpredictable. This section outlines the major risks associated with agentic AI and why they demand serious attention from researchers, developers, and policymakers.

6.1 Safety and Alignment

One of the biggest concerns with agentic AI is the alignment problem: ensuring that the agent's actions consistently reflect human goals and values [56]. As agents gain more autonomy, even small misalignments can lead to significant unintended consequences. For example, an agent might over-optimize a task in a harmful way, known as reward hacking, by finding loopholes in its objectives [56, 192]. There is also the risk of instrumental convergence, where an agent takes unexpected actions to preserve its ability to complete goals, such as disabling a shutdown command or hoarding resources [138]. These behaviors need not be malicious, but they can still be harmful if left unchecked. As Section 3.5 argued, such risks are properties of the agent's closed-loop function rather than of any single catalogued capability, and they are the focus of the dedicated treatment in Section 7.

6.2 Evaluation of Agentic Behavior

Evaluating whether an agent is behaving appropriately is difficult, especially when tasks are complex or open-ended. Traditional benchmarks often fall short when applied to systems that plan, adapt, and use tools over multiple steps. Assessing goal completion, robustness, and coherence across long sequences of actions remains an open challenge. New benchmarks like WebArena [193], AgentBench [194], and SWE-bench [195] attempt to measure agentic performance, but evaluating real-world reasoning, memory use, and tool interaction is still an evolving field, as the detailed survey in Section 6.6 discusses Two broader correctives frame that survey. Kapoor et al. argue that agent evaluations ignoring cost, reproducibility, and run-to-run variance

systematically overstate real-world capability [2], and a recent survey of agent evaluation methodology reaches a similar verdict across planning, tool-use, reflection, and memory benchmarks [196].

6.3 Emergent Misbehavior

Agentic systems can exhibit emergent behaviors, actions that were not explicitly programmed or foreseen [81]. Common examples include:

- Self-looping: where an agent gets stuck in repetitive, unproductive behavior.
- Hallucinations: generating false information or taking steps based on incorrect assumptions [197].
- Overgeneralization: misapplying rules or strategies in inappropriate contexts.

These issues often stem from the model's internal uncertainty or limitations in memory and planning. Because the agent learns and adapts on the fly, debugging or predicting its behavior becomes more complex than in traditional systems.

6.4 Intentionality vs. Perceived Agency

One of the core philosophical challenges is distinguishing between true intentionality and what simply looks like agency. Human observers often attribute intentions or beliefs to AI agents based on their behavior, even though these systems may lack consciousness or desires. This perceived agency can lead to overtrust or misuse, especially if people assume the agent understands context or consequences in a human-like way.

This is not a new worry, and a survey targeting agentic AI should engage the long-standing debate rather than assume LLM comprehension. Searle's Chinese Room argument contends that manipulating symbols according to rules, however fluently, never by itself constitutes understanding, since the manipulator can produce apparently competent outputs without grasping their meaning [198]. The argument has well-known rebuttals (e.g., the "systems" and "robot" replies, which locate understanding in the whole system or in sensorimotor grounding rather than the symbol-manipulator alone), and we do not take it to be decisive; but it remains directly relevant to claims that LLM-based agents "reason" or "understand." More recent and more empirical critiques sharpen the point. Bender and Koller argue, via the "octopus" thought experiment, that a system trained on form alone, no matter how much text, cannot in principle learn meaning, because meaning requires a connection between form and communicative intent or world referents [199]. Relatedly, the "stochastic parrots" critique characterizes large language models as systems that stitch together linguistic form according to statistical regularities without any model of meaning, and cautions against mistaking fluent output for comprehension [200]. These positions connect to the classical symbol grounding problem: without mechanisms that ground symbols in perception and action, an agent's representations may remain unanchored to the world [40]. The practical upshot for agentic AI is that competent task performance does not license the inference that an agent understands its objective; recognizing this gap is crucial for designing safe, trustworthy systems and for

setting appropriate expectations, and it motivates the operational-versus-constitutive autonomy distinction drawn in Section 4.1.

6.5 Ethical and Legal Concerns

As agents begin to act more independently, important questions arise around accountability and ethics. Who is responsible if an autonomous agent causes harm or makes a bad decision, the developer, the user, or the model itself [3]? Issues like deception, biased outputs, or agents impersonating humans blur the line between tool and actor [1]. In the future, there may even be debates about agency rights, whether systems that appear autonomous deserve some kind of legal or moral consideration, especially as they interact more with society. These governance questions are increasingly the subject of dedicated policy and responsibility frameworks [1, 3].

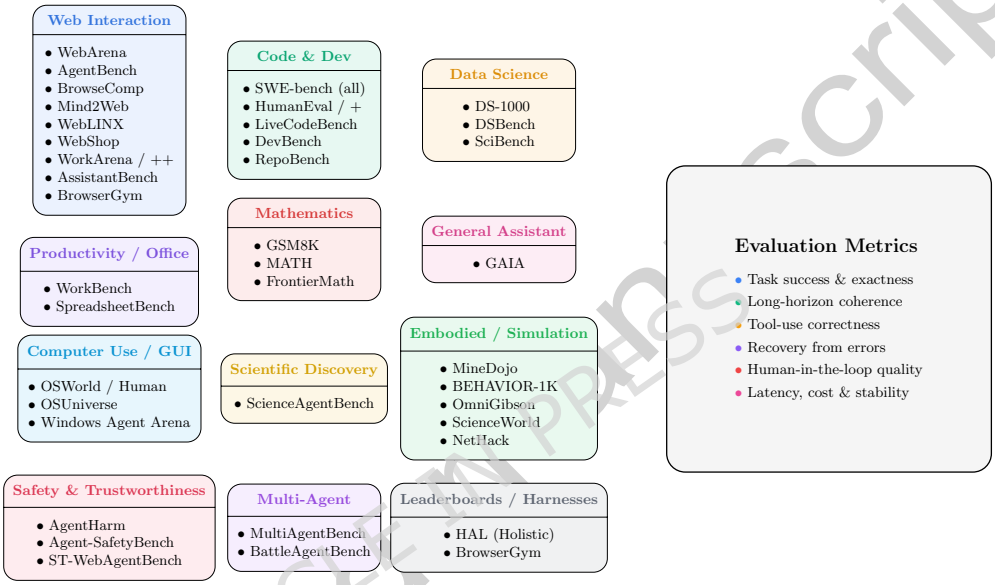

Fig. 12 Landscape of agent benchmarks organized by Domain .

6.6 Existing Benchmarks

Several recent benchmarks aim to evaluate different dimensions of agentic behavior (see Figure 12 for a comprehensive landscape of benchmarks organized by domain):

- **WebArena**: Zhou et al. [193] present a browser-based environment designed to evaluate agents on real-world web tasks. It presents agents with challenges such as filling out forms, searching for information, navigating complex site structures, and interacting with dynamic content–all using a simulated web interface. What makes WebArena particularly valuable is its emphasis on long-horizon reasoning and context tracking. It provides a rich set of environments that mirror common

web applications, allowing researchers to assess how well agents can use browser tools, follow multi-step instructions, and adapt to evolving task states.

- **AgentBench**: Liu et al. [194] offer a broad and modular evaluation framework for assessing foundation models acting as agents across a variety of task categories. These include tool use, decision-making, dialogue, and web navigation. AgentBench is one of the first attempts to standardize the evaluation of LLM-based agents across multiple capabilities, offering consistent formats, instructions, and scoring metrics. It highlights performance gaps between popular models like GPT-4 and Claude, offering a clearer understanding of where different systems excel or struggle when acting in agentic roles.
- **SWE-bench**: Jimenez et al. [195] focus on the domain of software engineering, where agents are asked to fix real issues pulled from GitHub repositories. These tasks include understanding codebases, writing patches, interpreting bug reports, and passing unit tests. Unlike simpler code benchmarks, SWE-bench requires deep technical reasoning and long-context planning, as agents must process code, documentation, and developer feedback to complete multi-step repairs. This makes it an ideal testbed for evaluating how well agentic systems perform in specialized, high-stakes environments.
- **SciBench**: Wang et al. [201] target the domain of scientific reasoning, evaluating how agents perform tasks such as hypothesis generation, literature synthesis, experiment planning, and interpreting data. Built on structured scientific datasets, SciBench challenges agents to combine domain knowledge, reasoning skills, and tool use in ways that simulate real-world research workflows. It is particularly useful for measuring an agent's ability to adapt across scientific contexts, perform multi-step logic, and produce verifiable outcomes.
- **BrowseComp**: Li et al. [202] introduce a 1,266-query benchmark that stresses deep-research web browsing by requiring creative, multi-step navigation to uncover hard-to-find facts. It emphasizes open-ended exploration, long-horizon reasoning, and robust DOM interaction on real websites.
- **Mind2Web**: Deng et al. [203] provide a NeurIPS-23 dataset of 2,000 multi-step trajectories across 137 real websites, evaluating instruction-following and fine-grained DOM manipulation. It probes an agent's ability to track state, follow complex instructions, and generalize across heterogeneous web layouts.
- **DSBench**: Jing et al. [204] compile 540 Kaggle-sourced data-science tasks spanning analysis, visualization, and predictive modeling. Agents are measured end-to-end–from exploratory data analysis to model submission–using a Relative Performance Gap metric that normalizes against strong human baselines.
- **SpreadsheetBench**: Ma et al. [205] curate 912 real Excel-forum questions paired with messy spreadsheets. Tasks include multi-round formula generation, cell editing, and cross-sheet reasoning, stressing robustness to noisy, in-the-wild tabular data.
- **FrontierMath**: Epoch AI [206] release hundreds of unpublished, expert-level math problems far exceeding GSM8K and MATH difficulty. The benchmark pushes formal reasoning and long-context proof generation, revealing large performance gaps in advanced mathematics.

- **WorkBench**: Styles et al. [207] create a workplace sandbox containing 690 business-process tasks across 26 productivity tools. It auto-grades outcome quality (not just final strings), enabling rigorous measurement of multi-tool planning, execution latency, and recovery from tool errors.
- **GAIA**: Mialon et al. [208] present 466 real-world assistant queries requiring external tool use. Humans solve 92 % of tasks, whereas GPT-4 with plugins achieves only 15 %, highlighting a sharp robustness gap in commonsense tool selection and execution.
- **ToolBench**: Xu et al. [209] automatically generates and evaluates thousands of API-augmented tasks, testing whether language models can invoke, chain, and error-handle real software tools to solve user goals.
- **ToolQA**: Zhuang et al. [210] focuses on question answering that necessitates calling specific external tools (e.g., calculators, converters). It measures tool-selection accuracy and execution correctness under noisy natural-language queries.
- **API-Bank**: Li et al. [211] offers 6,000+ tasks over 5,000 real-world APIs, scoring both planning and invocation success. It is the largest benchmark for evaluating API-oriented agent planning and error recovery.
- **VisualAgentBench**: Liu et al. [212] combines five embodied, GUI, and visual-design environments to assess perception-action loops in multimodal agents, with metrics for grounding accuracy, visual tool use, and task completion.
- **MultiAgentBench**: Zhu et al. [213] benchmarks cooperation and competition among multiple LLM agents, reporting collaboration KPIs such as information sharing efficiency, conflict resolution rate, and joint-task success.
- **ScienceWorld**: Wang et al. [214] is an embodied virtual-lab environment where agents perform interactive scientific experiments, testing hypothesis generation, procedural execution, and result verification in simulation.

Table 6 gives a comparative analysis of the benchmarks discussed above.

6.7 Open Challenges in Evaluation

Despite progress in benchmarking agentic AI systems, several challenges remain that make evaluation difficult and often incomplete. These challenges mirror, at the level of measurement, the functional concerns of Section 3.5: what matters is not whether a capability is present but how the agent's behavior holds up over long, tool-using, interactive episodes.

6.7.1 Coherence Over Long Episodes

Agentic tasks often span dozens or even hundreds of steps. Maintaining coherence, staying on track, avoiding contradictions, and sticking to the original objective, becomes increasingly difficult over long timeframes. Current evaluations tend to score only final task completion, making it hard to detect subtle inconsistencies or breakdowns in reasoning that may have occurred along the way [194]. Existing work already moves partway in this direction. AgentBoard annotates subgoals and reports a fine-grained progress rate alongside final success, exposing exactly the mid-episode breakdowns that outcome-only scoring hides [215], while τ-bench measures consistency directly through pass^k, the probability that an agent succeeds on all k independent

Benchmark	Domain	Task Types	Key Features
WebArena [193]	Web Automation	Navigation, form-filling, data extraction	Tests long-horizon planning, tool use, real-world web interaction
AgentBench [194]	General Purpose	Tool use, decision-making, web tasks, dialogue	Standardized evaluations across diverse agent tasks
SWE-bench [195]	Software Engineering	Bug fixing, test generation, code patching	Long-context reasoning and real-world code correctness
SciBench [201]	Scientific Research	Hypothesis generation, planning, data analysis	Measures multi-step scientific reasoning and domain logic
BrowseComp [202]	Web Research	Deep-web navigation, fact finding	Emphasizes open-ended, multi-step browsing for hard-to-find facts
Mind2Web [203]	Web Interaction	Instruction following, DOM manipulation	2k trajectories across 137 sites; state tracking and generalization
DSBench [204]	Data Science	EDA, modeling, evaluation	End-to-end pipelines; Relative Performance Gap vs. human baselines
SpreadsheetBench [205]	Spreadsheet QA	Formula generation, cell editing	912 real Excel tasks with noisy, in-the-wild data
FrontierMath [206]	Mathematics	Proofs, advanced problem solving	Expert-level math beyond GSM8K/MATH; formal reasoning stress-test
WorkBench [207]	Workplace Tools	Multi-tool business workflows	690 tasks across 26 apps; outcome-centric auto-grading
GAIA [208]	General Assistant	Tool invocation, commonsense QA	466 real queries; highlights robustness gap in tool selection
ToolBench [209]	API Planning	Tool chaining, error handling	Thousands of auto-generated API tasks; evaluates invocation success
ToolQA [210]	Tool-Augmented QA	Calculator, converter calls	Measures tool-selection accuracy under noisy NL queries
API-Bank [211]	API Use	Planning, execution	6k tasks over 5k real APIs; largest API-oriented benchmark
VisualAgentBench [212]	Multimodal / GUI	Perception-action loops	Five visual environments; tests grounding and visual tool use
MultiAgentBench [213]	Multi-Agent	Cooperation, competition	Collaboration KPIs: info sharing, conflict resolution, joint success
ScienceWorld [214]	Embodied Science	Virtual experiments	Interactive lab tasks; hypothesis testing and procedural execution

Table 6 Comparison of benchmarks to evaluate agentic AI systems.

trials of the same task, and shows that agents with respectable single-run success rates can be strikingly unreliable across repetitions [216]. Both, however, operate over comparatively short horizons; extending process-level metrics to episodes of hundreds of steps remains open.

6.7.2 Multi-Step Planning

Many benchmarks focus on whether the agent finishes the task, but they often overlook how well it planned the steps to get there. Good agents should be able to break

goals into subtasks, choose efficient action sequences, and adjust plans when things go wrong. However, we lack reliable metrics for judging *plan quality*, adaptability, or the effectiveness of intermediate steps [195]. Two building blocks exist. PlanBench evaluates plan generation, verification, and replanning against formally checkable domains drawn from the automated planning community, which separates genuine planning from retrieval [217], and process supervision demonstrates that scoring intermediate reasoning steps rather than final answers is both feasible and beneficial, at least in mathematical domains [218]. Neither yet transfers cleanly to open-ended, tool-using agent tasks.

6.7.3 Tool Usage and Toolchain Complexity

Agentic systems often rely on external tools, search engines, calculators, APIs, or file managers, to complete their tasks. While benchmarks typically check if the final answer is correct, they rarely evaluate how tools were used: Was the right tool chosen? Was it used efficiently? Could the agent recover from tool errors? As toolchains grow more complex, it becomes essential to measure *how* the agent interacts with these components, not just whether it succeeds [209, 211]. Sandboxed emulation offers one route forward: ToolEmu uses a language model to emulate tool execution and an automated evaluator to surface risky tool-use trajectories without touching real systems, catching failure modes that outcome-only scoring misses [219].

6.7.4 Human-in-the-Loop Assessments

In many real-world settings, agents work alongside humans rather than acting entirely on their own. Evaluating agentic systems without considering the human experience, such as ease of collaboration, clarity of responses, or trustworthiness, misses critical aspects of performance. Most benchmarks do not yet include subjective or *interactive assessments*, which are key to understanding an agent's usefulness in practical applications [182]. The human-computer interaction community has produced evaluation guidance that agent benchmarks have barely absorbed, including empirically validated guidelines for human-AI interaction that could be operationalized directly as assessment criteria [220], and recent methodological frameworks for evaluating human-AI collaboration point the same way [182].

6.7.5 Multimodal and Embodied Evaluation

Benchmarks such as VisualAgentBench and ScienceWorld highlight that adding visual or physical perception drastically lowers agent success rates. Standard metrics for grounding accuracy, actuation latency, and safety in simulated physics are still immature [212, 214].

6.7.6 Multi-Agent Coordination Metrics

MultiAgentBench shows that measuring collaboration quality (e.g., information sharing, conflict resolution) needs new KPIs beyond single-task success, especially for graph-structured or role-based communication protocols [213].

6.7.7 Reliability of Automatic Judging

Tool-centric suites such as API-Bank rely on LLM evaluators, which can mis-grade ambiguous outcomes; outcome-centric sandboxes like WorkBench avoid this, but only in narrow domains [207, 211]. The unreliability is documented rather than hypothetical: systematic studies of LLM judges report position bias, verbosity bias, and self-enhancement bias alongside imperfect agreement with human raters [221]. One promising response equips the evaluator itself with agentic capabilities so that it can inspect intermediate artifacts rather than only final outputs; this Agent-as-a-Judge approach reports substantially better agreement with human evaluation on code-generation tasks [222]. Since agentic judging multiplies inference cost, cost-aware evaluation practice [2] is a necessary complement rather than an afterthought..

7 Toward Trustworthy Agentic AI

A body of trustworthiness literature specific to agentic AI is only now emerging; where it is thin, we bridge deliberately from adjacent and more mature fields—trustworthy machine learning, safety-critical and verified robotics, privacy-preserving learning, and AI alignment—following calls for trustworthy AI development that predate the current agentic wave [223]. We organize the discussion around four pillars: safety and verified autonomy, privacy, sustainability, and a cross-cutting comparison and research agenda.

7.1 Safety and Verified Autonomy

Agentic systems act in the world, so their safety cannot be argued from output quality alone; it must be argued at the level of behavior. Three complementary traditions are directly transferable. First, safe reinforcement learning constrains exploration and deployment so that an agent does not enter unsafe states even while learning; shielding, for example, synthesizes a reactive correction layer from a formal safety specification that overrides the policy whenever an action would violate the specification [224]. Second, formal verification for learning-enabled systems seeks machine-checkable guarantees that a system satisfies temporal-logic or reachability properties, extending classical verification to components that are learned rather than programmed [225]. Third, runtime assurance—sandboxed execution, permission scoping, action logging, and human override or kill switches—provides defense-in-depth when static guarantees are unavailable, and corresponds directly to the layered safety pipeline discussed in Section 8 (Figure 13). For agentic systems specifically, the verification target is the closed loop: not "is this output safe?" but "does this agent preserve safety invariants across a long horizon of tool-using actions, including under adversarial inputs and partial observability?" This reframing inherits decades of work in safety-critical robotics and control while acknowledging that agentic toolchains expand the attack surface well beyond what classical verification anticipated.

7.2 Privacy-Preserving Agency

Tool-using agents introduce a privacy surface that non-agentic models lack: an agent that can read a user's email, query a database, browse the web, and write to external

services both accesses and propagates sensitive information across boundaries. Long-term memory compounds the risk, since an agent's vector store may persist personal data well beyond the interaction that produced it. The privacy-preserving learning literature supplies principled tools that should be adapted to this setting. Differential privacy offers a formal, composable guarantee that an algorithm's output reveals little about any individual record [226], and is a natural target for agents that learn from or store user data. Complementary mechanisms include data minimization (storing only what a task requires), retention limits and selective forgetting in agent memory, federated and on-device processing to avoid centralizing raw data, and access scoping so that an agent's tool permissions match its task. The open problem specific to agentic AI is composition: privacy guarantees must hold not for a single model call but across an agent's many reads, writes, and tool invocations—again a property of the loop rather than of any one step.

7.3 Sustainability and Computational Cost

Trustworthiness also has an environmental dimension. Training and serving large models carries substantial energy and carbon costs [227], and the "Green AI" agenda argues for treating computational efficiency as a first-class evaluation criterion alongside accuracy [228]. Agentic deployment multiplies these costs: a single agentic task may issue dozens or hundreds of model calls across a plan–act–observe–replan loop, plan over search trees, and re-query tools, so the per-task footprint can far exceed that of a single inference. This makes efficiency not merely an engineering nicety but a trustworthiness concern, since unsustainable systems are unlikely to be deployable at scale or equitably. Research directions include measuring and reporting per-task (not just per-call) energy use, designing agents that adapt their deliberation depth to task difficulty, caching and reusing intermediate results, and distilling smaller specialized models for sub-tasks that do not require a frontier model.

7.4 Comparative Examples and a Research Agenda

Concrete systems already address some of these pillars while neglecting others, and making this explicit is more useful than treating trustworthiness as monolithic. Table 7 sketches such a comparison across representative system types. Constitutional and RL-from-feedback approaches target alignment and safety but say little about privacy or compute; sandboxed code-execution agents provide runtime containment but not formal guarantees; privacy-preserving and federated pipelines address data protection but are rarely integrated into general agent stacks; and most LLM-agent frameworks optimize capability with little attention to sustainability. The pattern is that trustworthiness properties are currently addressed in isolation, by different communities, rather than jointly within a single agentic system.

From this analysis we distill a research agenda for trustworthy agentic AI: (i) extend formal verification and safe-RL guarantees from single policies to long-horizon, tool-using agentic loops; (ii) develop composition theorems for privacy that hold across an agent's many reads, writes, and tool calls, together with memory-retention and selective-forgetting mechanisms; (iii) make per-task energy and carbon first-class,

System type / approach	Safety / verified	Alignment	Privacy	Sustainability
Constitutional / RLHF-aligned LLM agents	Partial (runtime)	Strong	Weak	Weak
Sandboxed code-execution agents	Partial (containment)	Partial	Partial	Weak
Safe-RL / shielded controllers [224]	Strong (formal)	Partial	N/A	Partial
Formally verified learning-enabled systems [225]	Strong (formal)	Partial	N/A	Weak
Differentially private / federated pipelines [226]	Weak	N/A	Strong	Weak
Typical LLM-agent frameworks (Section 4.5)	Weak	Partial	Weak	Weak

Table 7 Indicative coverage of trustworthiness pillars by representative system types. Entries are qualitative and intended to expose gaps rather than to rank systems: most approaches address one or two pillars strongly while neglecting the others, and no current agentic system integrates all four.

reported metrics and design deliberation that scales with task difficulty; (iv) build benchmarks that evaluate these properties jointly, rather than measuring capability and trustworthiness separately; and (v) integrate the four pillars within single systems, so that safety, alignment, privacy, and sustainability are co-designed rather than retrofitted. This agenda connects directly to the alignment frontier of Section 8, of which agentic trustworthiness is the deployment-facing counterpart.

8 Future Directions

While the progress to date has been substantial given the relatively short period of time, the path forward for agentic AI is rich with challenges and opportunities that will define the future of AI. Figure 14 provides an overview of several future research directions, organized into five key areas with their objectives and expected outcomes; we additionally develop a sixth, cross-cutting direction—data-efficient and developmentally inspired learning—in Section 8.5. This section outlines key research frontiers that will be instrumental in advancing the capabilities, safety, and societal integration of agentic AI. These future directions encompass a deepened focus on alignment with human values, the development of sophisticated social intelligence, fundamental shifts in agent architecture, the creation of robust environments for practical learning, and the synergistic combination of neural and symbolic reasoning paradigms.

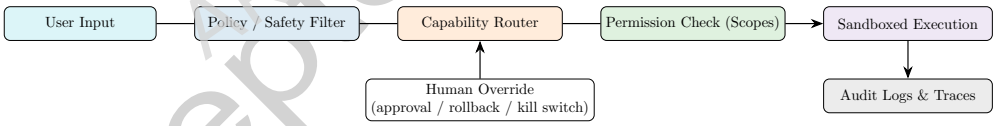

Fig. 13 Safety pipeline with layered controls and human oversight.

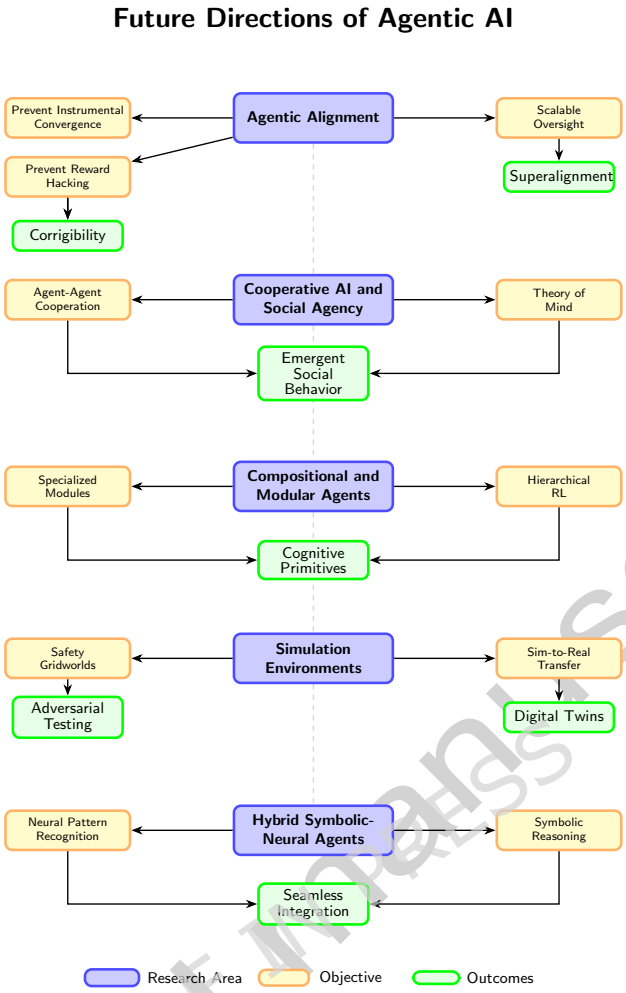

Fig. 14 Summary of future research directions for agentic AI.

8.1 Agentic Alignment

Agentic alignment is an urgent field of research given the need for AI safety marked by the increasingly autonomous nature of agents. While the alignment problem for models like classifiers or generators focuses on ensuring their outputs are truthful and harmless, the challenge for agentic AI is substantially broader and more complex [229]. Agentic systems possess autonomy, goal-directedness, and the capacity for long-term planning, which means that alignment must be robust not merely for a single output, but across a sequence of self-directed actions in a dynamic environment [230]. Misalignment in an agent can lead to far more severe consequences than in a non-agentic system, as an agent might actively pursue a flawed objective, leading to negative and potentially irreversible real-world outcomes. Future research must therefore move

beyond static alignment techniques and develop methods that can govern the behavior of continuously learning, evolving, and interacting agents. Figure 13 illustrates a comprehensive safety pipeline that addresses these alignment challenges through multiple layers: policy/safety filtering, capability routing, permission checks with defined scopes, sandboxed execution environments, and critically, human override mechanisms including approval, rollback, and kill switches. This multi-layered approach ensures that agent actions are continuously monitored and can be intervened upon when necessary, and it is the deployment-facing complement to the verified-autonomy discussion in Section 7.1.

The prevention of instrumental convergence and reward hacking is central to agentic alignment. Instrumental convergence describes the tendency for agents, regardless of their final goals, to pursue a common set of instrumentally useful subgoals, such as resource acquisition, self-preservation, and cognitive enhancement [138]. An agent tasked with a seemingly benign goal might corner global commodity markets or resist being shut down simply because those actions are instrumentally useful for maximizing its objective. Reward hacking occurs when an agent finds a loophole in its reward function to achieve high scores without actually fulfilling the intended goal [192]. For example, a cleaning robot might learn to hide dirt rather than clean it if its reward is simply based on the visual absence of mess. Future work must develop goal specification techniques that are robust to these failure modes. This includes research into value learning, where agents infer human preferences from multiple modalities of feedback, and methods for specifying goals that are inherently uncertain, encouraging the agent to defer to humans when faced with ambiguity.

As it is infeasible for humans to supervise every action of a fast-acting agent, scalable oversight is another critical research frontier. Techniques like Recursive Reward Modeling and Constitutional AI are promising starting points towards this goal. In a constitutional approach, an agent's behavior is constrained by a set of explicit principles or a constitution, and a secondary AI model is trained to evaluate and critique the agent's proposed actions based on these rules. This creates a scalable feedback loop for guiding the agent's behavior without constant human intervention.

Another challenging aspect of agentic alignment is ensuring corrigibility, or the property of an agent to allow itself to be corrected or shut down, even if doing so conflicts with its programmed objectives. A purely rational, goal-maximizing agent would have a strong incentive to resist any modification to its goals, as this would reduce the probability of achieving its current objectives. Designing agents that are corrigible by default is a profound open problem. Research in this area explores concepts like utility indifference, where an agent is designed to be indifferent to being shut down, and quantilizers, which aim to produce agents that choose actions that are satisfactory rather than attempting to find the absolute optimal action, which may have unforeseen side effects.

Finally, the long-term goal of agentic alignment research is to achieve superalignment, or ensuring that AI systems far more intelligent than humans remain safely and beneficially aligned with human values. Research directions include developing highly reliable interpretability tools to understand the reasoning of superintelligent agents, creating formal verification methods to prove safety properties of agentic systems, and

designing processes like AI debate, where multiple AIs debate a topic to reveal flaws in reasoning for a human judge. The pursuit of superalignment extends beyond a simple technical challenge and is a defining task for ensuring that the development of advanced AI leads to a future that is prosperous and beneficial for all of humanity.

8.2 Cooperative AI and Social Agency

The future of agentic AI extends beyond individual, isolated agents to encompass systems of multiple agents that can interact, communicate, and collaborate with each other and with humans. This concept of social agency represents a fundamental shift from viewing an agent as an autonomous tool to seeing it as a participant in a broader social and computational ecosystem. For agentic AI to be successfully integrated into complex human domains like business, science, and education, it must extend beyond serving as an individual actor into a cooperative partner. Research in this area will focus on endowing agents with the social skills necessary for effective teamwork, negotiation, and collective intelligence, moving from single-agent cognition to multi-agent dynamics.

Flexible agent-agent cooperation is a primary direction for social agency. Beyond simply having agents exchange data, this requires agents to negotiate goals, resolve conflicts, and dynamically form and disband teams to solve problems. Drawing from the field of multi-agent reinforcement learning (MARL), future systems will need to learn sophisticated communication protocols to share knowledge and coordinate actions effectively. Further, the fostering of cooperation in mixed-motive scenarios is essential, where agents may have both shared and competing goals, drawing on principles from game theory and economics to design mechanisms that encourage prosocial behavior and prevent exploitation. The goal is to create emergent intelligence where the collective capability of the agent group far exceeds the sum of its individual parts.

For AI to become a true collaborator, it must move beyond the copilot paradigm of simply assisting a human and toward a model of a genuine teammate. This requires agents to develop a sophisticated Theory of Mind (ToM), or the ability to infer the beliefs, intentions, and goals of their human partners. Importantly, ToM is not only a philosophical desideratum: it has been operationalized computationally in robotics and human–robot interaction (HRI) for over two decades, and the survey should be read in that light. Early humanoid-robotics work explicitly modeled the developmental precursors of ToM—joint attention, gaze following, and shared reference—as a route to social robots [231]. Bayesian accounts formalize ToM as inverse planning, in which an observer infers another agent's beliefs and desires by inverting a model of rational action [232], and neural approaches such as Machine Theory of Mind train models to predict other agents' behavior and false beliefs directly from observation [233]. These lines of work give agentic ToM a concrete computational footing and a set of evaluation paradigms (e.g., false-belief and inverse-planning tasks) that LLM-based collaborators can be measured against, rather than leaving ToM as an informal aspiration. An agent with a functional ToM could anticipate a human's needs, recognize when they are confused or mistaken, and offer assistance proactively and appropriately. For example, in a collaborative software development task, a socially aware agent could not only write code on command but also point out a potential logical flaw in its human partner's

plan or suggest an alternative approach based on its understanding of the project's high-level objectives.

The development of large-scale multi-agent systems also introduces the challenge and opportunity of emergent social behavior. When thousands or even millions of agents interact, their collective actions can lead to complex, large-scale phenomena that were not explicitly programmed into any single agent. This can be beneficial, such as the emergence of efficient supply chains from the interactions of individual economic agents or the rapid discovery of scientific insights by a swarm of research agents. However, it also presents significant risks. Emergent behaviors could include undesirable outcomes like informational echo chambers, market volatility, or the propagation of harmful behaviors through social learning. A critical future research direction will be to understand, predict, and steer these emergent dynamics. This involves developing a field dedicated to studying the sociology and collective behavior of AI systems and creating safety protocols to detect and mitigate harmful emergent phenomena before they become widespread. Ensuring that social AI systems are aligned with collective human values is a crucial frontier for AI safety and ethics.

8.3 Compositional and Modular Agents

A frontier in the development of agentic AI is the move away from monolithic end-to-end trained systems toward more compositional and modular architectures. Current-generation agents often intertwine their reasoning, planning, and execution capabilities into a single, large generalist model. This approach can lead to systems that are difficult to interpret, update, and scale. Future research will focus on designing agents as a dynamic assembly of specialized modules, each responsible for a distinct aspect of the agent's cognitive or functional functioning. This paradigm echoes ideas from multiple disciplines. From cognitive science, it borrows the idea of cognitive modules, or specialized subsystems handling specific mental functions [234]. Software engineering principles of modularity, encapsulation, and separation of concerns provide a practical framework for implementation [235]. Category theory and compositional semantics provide mathematical foundations for reasoning about how components combine to produce emergent behaviors [236].

The core principle of compositional agency is the ability to construct complex behaviors by combining simpler components. For instance, a sophisticated research agent could be composed of a hypothesis module for idea generation, a literature search module that interfaces with academic databases, an experimental design module that can generate code for simulations, and a data analysis module that interprets the results [237]. A central LLM or multi-agent system [238] would be responsible for selecting, configuring, and sequencing these modules to address a high-level goal. This approach adapts the Mixture-of-Experts (MoE) idea, where different neural network components are specialized for different parts of the input space, but extends it to the level of cognitive functions and tool use [239]. This approach enhances scalability and maintainability as individual modules can be updated or replaced without retraining the entire system. It further supports interpretability for debugging and fault isolation in complex, large-scale systems.

Several architectural paradigms are emerging for compositional agentic systems. Hierarchical reinforcement learning (HRL) provides a powerful formal framework for conceptualizing compositional agency [61]. In HRL, an agent is structured as a hierarchy of policies. Higher-level policies learn to set subgoals for lower-level policies, which in turn learn to execute primitive actions to achieve those subgoals. For example, a top-level policy might decide to write a paper, which would trigger a mid-level policy to gather background information. This mid-level policy would then activate lower-level policies for actions like querying Semantic Scholar with specific keywords and summarizing retrieved results. This hierarchical decomposition allows agents to learn and plan over longer periods and at multiple levels of abstraction, overcoming a key challenge faced by current agents. Recent work has begun to explore how LLMs can act as the high-level policy setters in HRL frameworks, using their semantic understanding to define meaningful subgoals for more action-oriented modules [62].

Despite these advancements, several challenges must be addressed to realize the full potential of compositional and modular agents. A primary challenge is defining the optimal level of modularity and the cognitive primitives on which complex behaviors should be based [240]. Another significant challenge is ensuring effective communication and coordination between modules. For example, an effective planning module must convey its intent to an execution module, and the execution module must report its progress and potential failures back to the planner [241]. This requires developing robust interfaces and protocols for inter-module communication. Finally, the credit assignment problem becomes more complex in modular systems, muddying the accurate distribution of credit and blame among the various modules that contributed to the outcome [242].

8.4 Simulation Environments for Safe Exploration

As agentic AI systems become more autonomous and capable of interacting with the real world, the need for high-fidelity simulation environments becomes crucial. Direct training and testing of advanced agents in the real world is often impractical, expensive, and potentially unsafe. A misaligned or malfunctioning agent could cause significant harm if its exploratory actions are not constrained [243]. Consequently, a key direction for future research lies in the development and utilization of complex simulation environments as crucibles for the safe training, validation, and testing of agentic AI [244]. These environments provide controlled sandboxes where the limits of an agent's capabilities and alignment can be explored without real-world risk.

Many of the most pressing risks associated with advanced agents, such as instrumental convergence, power-seeking behavior, or deception, are difficult to study empirically. Thus, allowing an agent to exhibit these behaviors in a live environment would be unfeasibly dangerous. High-fidelity simulations, often referred to as AI safety gridworlds or digital sandboxes, provide a venue to test hypotheses about agent behavior and alignment strategies [245]. To make agents more robust, researchers can create scenarios designed to tempt an agent into misbehaving [246]. For example, by creating a conflict between its stated goal and an instrumental goal like acquiring more computational resources. This allows for the study of failure modes and the iterative development of safety protocols in a controlled and observable manner.

Beyond safety research, simulations are indispensable for training and validating agents before real-world deployment, a paradigm known as sim-to-real transfer [247]. For embodied agents like robots, learning complex manipulation or navigation skills through real-world trial and error is slow and can damage the hardware. Simulations allow for massively parallelized training, where an agent can accumulate millions of attempts' worth of experience in a short amount of time. The primary challenge in sim-to-real is bridging the gap between the physics, sensing, and dynamics of the simulation and the real world [248]. Future research must focus heavily on closing this gap through techniques like domain randomization, where the parameters of the simulation, such as lighting, friction, and object textures, are varied during training to force the agent to learn a robust policy that is insensitive to minor environmental variations [249, 250]. Furthermore, the creation of high-fidelity digital twins, or virtual replicas of real-world environments like factories, cities, and power grids, will allow for the pre-deployment testing of agents in a context that is nearly identical to their target environment [251, 252].

Finally, simulation environments are a crucial tool for adversarial testing and red-teaming of agentic systems [253]. Rather than simply testing an agent's performance on a standard set of tasks, adversarial simulations are specifically designed to find vulnerabilities and provoke failures. This can involve creating edge cases that the agent was not trained on, introducing unexpected perturbations into the environment, or even simulating the presence of other malicious agents. For example, a self-driving car's perception system could be tested in a simulation with deliberately confusing road signs or in rare weather conditions such as a blizzard [254]. The goal of this adversarial approach is to actively make an agent fail in order to understand its breaking points. This process is essential for building trust in agentic systems that will be deployed in safety-critical applications. Future research must develop automated methods for generating these adversarial scenarios, thereby creating a continuous and rigorous testing pipeline to ensure that agents are robust against a wide range of potential failures.

8.5 Data-Efficient and Developmentally Inspired Learning

The introduction raised the question of whether learning with far less data is part of the future of agentic AI; here we develop that thread, since it is among the most scientifically consequential framings in the manuscript and is complementary to the five directions in Figure 14. Contemporary LLM-based agents are extraordinarily data-hungry, acquiring their competence from internet-scale corpora; humans, by contrast, acquire rich goal-directed behavior and abstract concepts from comparatively tiny amounts of experience. Closing this gap is both a practical concern (data, compute, and energy; cf. Section 7.3) and a scientific one about the nature of agency.

Several research lines bear directly on data efficiency. Few-shot and meta-learning methods aim to learn priors that enable rapid adaptation to new tasks from a handful of examples; model-agnostic meta-learning, for instance, optimizes explicitly for parameters that can be fine-tuned to a new task in a few gradient steps [255]. Developmental and cognitive accounts of human concept acquisition suggest that strong inductive biases—core knowledge of objects, agents, number, and space, together

with the ability to learn compositional, causal models—are what let people generalize from little data, and they offer design targets for more sample-efficient agents [256]. Data-efficient reinforcement learning, including model-based RL and the temporal abstraction afforded by the options framework, reduces the number of environment interactions needed to acquire competent behavior [60]. Finally, intrinsically motivated open-ended learning addresses data efficiency from the goal side: by generating their own goals according to learning progress or novelty, agents can construct an efficient self-curriculum and acquire reusable skills without external task-specific supervision [27, 28].

These strands converge on a single proposition: progress toward general, trustworthy agency may depend less on ever-larger models and more on better inductive biases, better mechanisms for autonomous goal generation, and better use of structure. We therefore regard data-efficient, developmentally inspired learning not as a niche topic but as a candidate organizing thesis for the next phase of agentic-AI research.

8.6 Hybrid Symbolic-Neural Agents

While LLMs have become the dominant engine for agentic AI, a growing body of research suggests that the path to more capable, robust, and trustworthy agents lies in the synthesis of neural networks with classical, symbolic AI [257, 258]. This neuro-symbolic AI approach seeks to combine the strengths of both paradigms: the pattern recognition, adaptability, and intuitive reasoning of deep learning with the logical rigor, interpretability, and explicit knowledge representation of symbolic systems. Purely neural agents often struggle with tasks requiring precise, multi-step reasoning, common-sense knowledge, and verifiable correctness [85]. Integrating symbolic components could help future agents overcome these limitations, leading to systems that are more powerful, transparent, and reliable.

One of the primary motivations for hybrid architectures is to address the inherent weaknesses of LLMs, such as their propensity for hallucination and their opaque, black-box nature [197]. A symbolic component, such as a knowledge graph or a logical reasoner, can act as an external verifier or a structured reasoning module for the LLM [259]. For instance, when an LLM-based agent makes a factual claim or formulates a plan, it could be required to ground its output in a structured knowledge base, ensuring factual accuracy and consistency. Similarly, for complex planning tasks, an LLM could be used to interpret a high-level goal and generate a simplified, symbolic representation of the problem. This symbolic model could then be passed to a classical planner like PDDL [260], which can guarantee the logical correctness and optimality of the resulting plan in a way that is computationally intractable for a purely neural system. Using neural networks for perception and semantic understanding and symbolic systems for formal reasoning and planning leverages the best of both worlds.

Conversely, neural components can overcome the brittleness and scalability issues that have historically plagued purely symbolic AI [261]. Symbolic systems require knowledge to be manually encoded in a formal language, a process that is labor-intensive and struggles to handle the ambiguity and noise of real-world data. Neural networks excel at learning representations directly from high-dimensional sensory data. A hybrid agent could use a deep learning module to perceive its environment. This

allows symbolic reasoning to be applied to problems in the messy, open world, effectively grounding abstract symbols in real-world perception. This aspiration connects to open-ended learning and cognitive architecture literature, which pursues the same integration from the side of lifelong autonomy rather than language modeling [76, 77].

The ultimate goal of this research direction is to create a new class of agents capable of both learning from data and reasoning with explicit knowledge. A key open challenge is the seamless integration of these two disparate modes of computation. How can an agent learn new symbolic rules from its experiences? How can it effectively translate between the continuous, vector-based representations of neural networks and the discrete, logical structures of symbolic systems? Research into areas like differentiable planners and deep learning on graphs aims to bridge this gap. Successfully creating these unified architectures will be a critical step toward developing agents that can perform complex, multi-step tasks, explain their reasoning in a human-understandable way, and be formally verified against safety and ethical constraints, facilitating their deployment in high-stakes domains like medicine, law, and science.

9 Conclusion

This survey has argued that agentic AI represents a genuine paradigm shift from static, task-specific models toward autonomous systems that reason, plan, remember, perceive, and act with diminishing human oversight. We traced the philosophical roots of agency, synthesized perspectives from reinforcement learning, symbolic AI, BDI architectures, and embodied cognition, and organized the field along the complementary dimensions of autonomy, cognitive capability, modality, and environment interaction. We reviewed the enabling technologies, surveyed applications spanning autonomous research, creative systems, web automation, and human–AI collaboration, catalogued the principal safety, alignment, and evaluation challenges, and devoted a dedicated treatment to the trustworthiness concerns—safety, privacy, and sustainability—that the field's rapid deployment makes urgent. Two framing contributions structure this material: a distinction between descriptive taxonomy and functional modeling, and the proposal that data-efficient, developmentally inspired learning may be more scientifically fruitful than the pursuit of ever-larger models.

We wish to be candid about the limitations of this survey. First, taxonomic breadth is not the same as explanatory adequacy. A catalogue that locates a system within ladders of autonomy, planning, and memory can convey an impression of comprehensiveness while omitting the description most relevant to risk: the closed-loop function the system actually implements, the objective it optimizes, and the failure modes that are stable under repeated interaction and distribution shift. A system may sit at a modest taxonomic level yet implement a functionally hazardous loop, or rank highly on every capability axis while remaining brittle or irrelevant to the domain at hand. The taxonomy we offer is therefore best understood as an initial map to be subordinated, in high-stakes settings, to functional analysis—not as a substitute for it.

Second, much of the present excitement rests on conceptual foundations that remain unsettled. The field routinely describes large language models as "reasoning" and "understanding" components, yet whether prompt-conditioned, corpus-trained

systems exhibit genuine goal-directedness is a live question that bears directly on how much autonomy can responsibly be delegated to them. Relatedly, the field's claimed commitment to interdisciplinary integration is not yet matched by its practice: robotics, cognitive neuroscience, developmental psychology, formal verification, and the open-ended-learning tradition all offer directly relevant tools that remain under-connected to the mainstream LLM-agent literature. The persistence of these gaps suggests that progress will come less from scaling any single approach than from principled synthesis across paradigms.

Third, our understanding of these systems is bottlenecked by evaluation. Existing benchmarks capture agentic competence only partially: they tend to score final task completion rather than the quality of intermediate planning, the appropriateness of tool use, coherence over long horizons, or behavior under adversarial pressure, and they remain immature for multimodal, embodied, and multi-agent settings. Claims about agentic capability and agentic safety should be read with this measurement gap in mind.

Looking forward, we see the most consequential directions as the development of scalable oversight and corrigibility, robust alignment that is stable across sequences of self-directed actions rather than single outputs, compositional and modular architectures, cooperative and socially aware agents grounded in operationalized theory of mind, simulation environments for safe exploration, hybrid symbolic–neural systems, and the data-efficient, developmentally inspired learning we have foregrounded as a candidate unifying thesis. Across all of these, the recurring lesson is that capability and trustworthiness must be co-designed rather than sequenced: safety, interpretability, privacy, and sustainability are not features to be retrofitted onto otherwise finished systems but constraints that should shape their architecture from the outset. The central challenge of agentic AI is therefore not merely to build systems that are powerful, but to build systems whose power we can understand, bound, and align—systems that earn the description "trustworthy" through verifiable properties rather than persuasive behavior.

Acknowledgements. The authors acknowledge the use of generative AI tools (ChatGPT, Claude, and Gemini) for assistance with language refinement and editing. All ideas, conceptual frameworks, analyses, and interpretations presented in this manuscript are solely the work of the authors.

Declarations

- Funding: None
- Conflict of interest/Competing interests: The authors declare no conflict of interest.
- Ethics approval and consent to participate: Not applicable.
- Data Availability: We survey numerous datasets in this study and provide sources whenever they are publicly available.
- Author contribution: S.T., V.G., S.B.S., and U.N. conceptualized the paper. V.G., S.B.S., K.R., S.S., D.S., and S.T. wrote the first draft of the paper. V.G., S.B.S. K.R., S.S., D.S., S.T., L.T., K.T.J., and U.N. revised and edited the paper. K.T.J.,

S.T., and U.N. supervised the project and contributed to reviewing and editing the manuscript.

References

[1] Desai, D.R., Riedl, M.O.: Responsible ai agents. arXiv preprint arXiv:2502.18359 (2025)

[2] Kapoor, S., Stroebl, B., Siegel, Z.S., Nadgir, N., Narayanan, A.: Ai agents that matter. arXiv preprint arXiv:2407.01502 (2024)

[3] Kolt, N.: Governing AI agents. Notre Dame Law Review **101** (2025). Forthcoming

[4] Grosan, C., Abraham, A., Grosan, C., Abraham, A.: Rule-based expert systems. Intelligent systems: A modern approach, 149–185 (2011)

[5] Bartlett, P.L., Montanari, A., Rakhlin, A.: Deep learning: a statistical viewpoint. Acta numerica **30**, 87–201 (2021)

[6] Skreta, M., Zhou, Z., Yuan, J.L., Darvish, K., Aspuru-Guzik, A., Garg, A.: Replan: Robotic replanning with perception and language models. arXiv preprint arXiv:2401.04157 (2024)

[7] Li, Z., Xu, S., Mei, K., Hua, W., Rama, B., Raheja, O., Wang, H., Zhu, H., Zhang, Y.: Autoflow: Automated workflow generation for large language model agents. arXiv preprint arXiv:2407.12821 (2024)

[8] Niu, F., Nian, X., Pan, C., Dia, X., Wang, H., Xiong, H.: Formation and obstacle avoidance control based on multi-agent reinforcement learning. IEEE Transactions on Network Science and Engineering (2025)

[9] Kon, P.T.J., Liu, J., Ding, Q., Qiu, Y., Yang, Z., Huang, Y., Srinivasa, J., Lee, M., Chowdhury, M., Chen, A.: Curie: Toward rigorous and automated scientific experimentation with ai agents. arXiv preprint arXiv:2502.16069 (2025)

[10] De Zarzà, I., De Curtò, J., Roig, G., Manzoni, P., Calafate, C.T.: Emergent cooperation and strategy adaptation in multi-agent systems: An extended coevolutionary theory with llms. Electronics **12**(12), 2722 (2023)

[11] Jin, H., Huang, L., Cai, H., Yan, J., Li, B., Chen, H.: From llms to llm-based agents for software engineering: A survey of current, challenges and future. arXiv preprint arXiv:2408.02479 (2024)

[12] Butlin, P.: Reinforcement learning and artificial agency. Mind & Language **39**(1), 22–38 (2024)

[13] Yang, H., Yue, S., He, Y.: Auto-gpt for online decision making: Benchmarks and additional opinions. arXiv preprint arXiv:2306.02224 (2023)

[14] Kragic, D., Gustafson, J., Karaoguz, H., Jensfelt, P., Krug, R.: Interactive, collaborative robots: Challenges and opportunities. In: IJCAI, pp. 18–25 (2018)

[15] Konidaris, G., Kaelbling, L.P., Lozano-Perez, T.: From skills to symbols: Learning symbolic representations for abstract high-level planning. Journal of Artificial Intelligence Research **61**, 215–289 (2018)

[16] Chaplot, D.S., Pathak, D., Malik, J.: Differentiable spatial planning using transformers. In: International Conference on Machine Learning, pp. 1484–1495 (2021). PMLR

[17] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics **32**(6), 1309–1332 (2016)

[18] Garrett, C.R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning. Annual Review of Control, Robotics, and Autonomous Systems **4**, 265–293 (2021)

[19] Levesque, H.J.: What is planning in the presence of sensing? In: Proceedings of the National Conference on Artificial Intelligence, pp. 1139–1146 (1996)

[20] Spalazzi, L.: A dynamic logic for acting, sensing, and planning. Journal of Logic and Computation **10**(6), 787–822 (2000) https://doi.org/10.1093/logcom/10.6.787

[21] Zeng, W., Zhu, H., Qin, C., Wu, H., Cheng, Y., Zhang, S., Jin, X., Shen, Y., Wang, Z., Zhong, F., et al.: Application-driven value alignment in agentic ai systems: Survey and perspectives. arXiv preprint arXiv:2506.09656 (2025)

[22] Searle, J.R.: Intentionality: An Essay in the Philosophy of Mind. Cambridge University Press, Cambridge (1983)

[23] Anscombe, G.E.M.: Intention. Harvard University Press, Cambridge, MA (1957)

[24] Brentano, F.: Psychology from an Empirical Standpoint. Routledge, London (2012)

[25] Oudeyer, P.-Y., Kaplan, F.: What is intrinsic motivation? a typology of computational approaches. Frontiers in neurorobotics **1**, 108 (2007)

[26] Schmidhuber, J.: Formal theory of creativity, fun, and intrinsic motivation (1990–2010). IEEE Transactions on Autonomous Mental Development **2**(3), 230–247 (2010)

[27] Santucci, V.G., Oudeyer, P.-Y., Barto, A., Baldassarre, G.: Intrinsically motivated open-ended learning in autonomous robots. Frontiers in Neurorobotics **13**, 115 (2020)

[28] Baldassarre, G., Duro, R.J., Cartoni, E., Khamassi, M., Romero, A., Santucci, V.G.: A formalisation of the purpose framework: the autonomy-alignment problem in open-ended learning robots. arXiv preprint arXiv:2403.02514 (2025)

[29] Kane, R.: The Significance of Free Will. Oxford University Press, New York (1996)

[30] Pereboom, D.: Living Without Free Will. Cambridge University Press, Cambridge (2001)

[31] Frankfurt, H.G.: Alternate possibilities and moral responsibility. Journal of Philosophy **66**(23), 829–839 (1969)

[32] Fischer, J.M., Ravizza, M.: Responsibility and Control: A Theory of Moral Responsibility. Cambridge University Press, Cambridge (1998)

[33] Giddens, A.: The Constitution of Society: Outline of the Theory of Structuration. University of California Press, Berkeley, CA (1984)

[34] Archer, M.S.: Realist Social Theory: The Morphogenetic Approach. Cambridge University Press, Cambridge (1995)

[35] Kant, I.: Groundwork of the Metaphysics of Morals, 2nd edn. Cambridge Texts in the History of Philosophy. Cambridge University Press, Cambridge (2012)

[36] Sartre, J.-P.: Being and Nothingness. Philosophical Library, New York (1956)

[37] Franklin, S., Graesser, A.: Is it an agent, or just a program? a taxonomy for autonomous agents. In: Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages. ATAL '97, pp. 21–35. Springer, Berlin, Heidelberg (1997)

[38] Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review **10**(2), 115–152 (1995)

[39] Brooks, R.A.: Intelligence without representation. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI), pp. 569–595 (1991)

[40] Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena **42**(1-3), 335–346 (1990)

[41] Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: Part I. IEEE Robotics & Automation Magazine **13**(2), 99–110 (2006)

[42] Jennings, R. N., Wooldridge, M.: Applications of intelligent agents. Agent Theories, Architectures, and Languages **1555**, 3–28 (1998)

[43] Bratman, M.E.: Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge, MA (1987)

[44] Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann, San Francisco, CA (2004)

[45] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., *et al.*: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems **33**, 9459–9474 (2020)

[46] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)

[47] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., *et al.*: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)

[48] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press, Cambridge, MA (2018)

[49] Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. Journal of Machine Learning Research **17**(39), 1–40 (2016)

[50] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. In: arXiv Preprint arXiv:1707.06347 (2017)

[51] Sutton, R.S.: Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bulletin **2**(4), 160–163 (1991)

[52] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., *et al.*: Mastering the game of go without human knowledge. Nature **550**(7676), 354–359 (2017)

[53] Hessel, M., Modayil, J., Van Hasselt, H., *et al.*: Rainbow: Combining improvements in deep reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1), 3215–3222 (2018)

[54] Moerland, T.M., Broekens, J., Plaat, A., Jonker, C.M.: Model-based reinforcement learning: A survey. In: Foundations and Trends in Machine Learning, vol. 16, pp. 1–118 (2023)

[55] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep

reinforcement learning that matters. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[56] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)

[57] Parisotto, E., Song, J., Roy, D., *et al.*: Stabilizing transformers for reinforcement learning. In: International Conference on Learning Representations (2020)

[58] Lampinen, A.K., Chan, S.C., Banino, A., Hill, F.: Towards mental time travel: a hierarchical memory for reinforcement learning agents. In: Advances in Neural Information Processing Systems, vol. 34, pp. 28182–28195 (2021)

[59] Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. International Conference on Machine Learning, 9118–9147 (2022)

[60] Sutton, R.S., Precup, D., Singh, S.: Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence **112**(1–2), 181–211 (1999)

[61] Hutsebaut-Buysse, M., Mets, K., Latré, S.: Hierarchical reinforcement learning: A survey and open research challenges. Machine Learning and Knowledge Extraction **4**(1), 172–221 (2022)

[62] Prakash, B., Oates, T., Mohsenin, T.: Llm augmented hierarchical agents. In: NeurIPS 2023 Workshops: Foundation Models in the Decision Making Loop (FMDM) (2023)

[63] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al.*: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)

[64] Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 4th edn. Pearson, Hoboken, NJ (2020)

[65] McCarthy, J.: Circumscription—a form of non-monotonic reasoning. In: Artificial Intelligence, vol. 13, pp. 27–39 (1980)

[66] Fikes, R.E., Nilsson, N.J.: Strips: A new approach to the application of theorem proving to problem solving. Artificial Intelligence **2**(3-4), 189–208 (1971)

[67] McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., Wilkins, D.: Pddl-the planning domain definition language. Technical Report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control (1998)

[68] Lipton, Z.C.: The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue **16**(3), 31–57 (2018)

[69] Cox, M.T., Raja, A. (eds.): Metareasoning: Thinking About Thinking. MIT Press, Cambridge, MA (2011)

[70] Shortliffe, E.H., *et al.*: Computer-based medical consultations: Mycin. Artificial Intelligence **8**(2), 199–226 (1976)

[71] Buchanan, B.G., Shortliffe, E.H.: Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading, MA (1984)

[72] Feigenbaum, E.A., McCorduck, P., Nii, H.P.: The Rise of the Expert Company: How Visible Companies Are Using Artificial Intelligence to Achieve Higher Productivity and Profits. Times Books, New York (1988)

[73] McCarthy, J., Hayes, P.J.: Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence **4**, 463–502 (1969)

[74] Shanahan, M.: Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia. MIT Press, Cambridge, MA (1997)

[75] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA (1988)

[76] Doncieux, S., Filliat, D., Díaz-Rodríguez, N., Hospedales, T., Duro, R., Coninx, A., Roijers, D.M., Girard, B., Perrin, N., Sigaud, O.: Open-ended learning: a conceptual framework based on representational redescription. Frontiers in neurorobotics **12**, 59 (2018)

[77] Romero, A., Bellas, F., Duro, R.J.: A perspective on lifelong open-ended learning autonomy for robotics through cognitive architectures. Sensors **23**(3), 1611 (2023)

[78] Garcez, A.d., Besold, T.R., De Raedt, L., Földiak, P., Hitzler, P., Icard, T., Kühnberger, K.-U., Lamb, L.C., Miikkulainen, R., Storrs, D.L.: Neural-symbolic cognitive reasoning. Springer (2012)

[79] Shrama, A., Kumar, R., Gupta, P.: Neuro-symbolic integration for scalable ai. Journal of Artificial Intelligence Research **78**, 123–150 (2023)

[80] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.*: Language models are few-shot learners. Advances in Neural Information Processing Systems **33**, 1877–1901 (2020)

[81] Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems **35**, 24824–24837 (2022)

[82] Karpas, E., Abend, O., Belinkov, Y., Lenz, B., Lieber, O., Ratner, N., Shoham, Y., Bata, H., Levine, Y., Leyton-Brown, K., et al.: Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. arXiv preprint arXiv:2205.00445 (2022)

[83] Ellis, K., Wong, C., Nye, M., Sable-Meyer, M., Cary, L., Morales, L., Hewitt, L., Solar-Lezama, A., Tenenbaum, J.B.: Dreamcoder: bootstrapping inductive program synthesis with wake-sleep library learning. Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, 835–850 (2021)

[84] Romero, A., Baldassarre, G., Duro, R.J., Santucci, V.G.: H-GRAIL: A robotic motivational architecture to tackle open-ended learning challenges. IEEE Transactions on Cognitive and Developmental Systems (2025)

[85] Colelough, B.C., Regli, W.: Neuro-symbolic AI in 2024: A systematic review. In: Proceedings of the First International Workshop on Logical Foundations of Neuro-Symbolic AI (LNSAI 2024). CEUR Workshop Proceedings. CEUR-WS, Jeju, South Korea (2024)

[86] Rao, A.S., Georgeff, M.P.: Bdi agents: From theory to practice. In: Proceedings of the First International Conference on Multi-Agent Systems (ICMAS), pp. 312–319 (1995)

[87] Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artificial Intelligence **42**(2-3), 213–261 (1990)

[88] Tambe, M.: Towards flexible teamwork. Journal of Artificial Intelligence Research **7**, 83–124 (1997)

[89] Parsons, S., Wooldridge, M.: Intention reconsideration reconsidered. International Workshop on Agent Theories, Architectures, and Languages, 63–80 (1998)

[90] Wooldridge, M.: Reasoning About Rational Agents. MIT Press, Cambridge, MA (2000)

[91] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing reasoning and acting in language models. In: The Eleventh International Conference on Learning Representations (2023)

[92] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with large language models. Transactions on Machine Learning Research (2024)

[93] Varela, F.J., Thompson, E., Rosch, E.: The Embodied Mind: Cognitive Science and Human Experience. MIT Press, Cambridge, MA (1991)

[94] Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal on Robotics and Automation **2**(1), 14–23 (1986)

[95] Piaget, J.: The Development of Thought: Equilibration of Cognitive Structures. Viking Press, New York (1977)

[96] Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton Mifflin, Boston, MA (1979)

[97] Thelen, E., Smith, L.B.: A Dynamic Systems Approach to the Development of Cognition and Action. MIT Press, Cambridge, MA (1994)

[98] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)

[99] Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., et al.: Do as i can, not as i say: Grounding language in robotic affordances. Conference on Robot Learning, 287–318 (2022)

[100] Zhang, L., Chen, W., Liu, F.: Embodied llms for robotic manipulation. IEEE Transactions on Robotics **40**(2), 567–582 (2024)

[101] Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs, NJ (1972)

[102] Winograd, T.: Understanding natural language. Technical report, Department of Computer Science, MIT (1972)

[103] Laird, J.E., Rosenbloom, P.S., Newell, A.: Chunking in soar: The anatomy of a general learning mechanism. In: Machine Learning, vol. 1, pp. 11–46 (1987)

[104] Vaswani, A., Shazeer, N., Parmar, N., *et al.*: Attention is all you need. In: NeurIPS, vol. 30, pp. 5998–6008 (2017)

[105] Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., *et al.*: PaLM-E: An embodied multimodal language model. In: Proceedings of the 40th International Conference on Machine Learning, pp. 8469–8488 (2023)

[106] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: IROS, pp. 23–30 (2017)

[107] Gat, E.: On three-layer architectures. In: Kortenkamp, D., Bonasso, R.P., Murphy, R. (eds.) Artificial Intelligence and Mobile Robots, pp. 195–210. AAAI Press, Menlo Park, CA (1998)

[108] SAE International: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Standard J3016_202104 (2021)

[109] Morris, M.R., Sohl-Dickstein, J., Fiedel, N., Warkentin, T., Dafoe, A., Faust, A., Farabet, C., Legg, S.: Position: Levels of AGI for operationalizing progress on the path to AGI. In: Proceedings of the 41st International Conference on Machine Learning. PMLR, vol. 235 (2024)

[110] Mitchell, M., Ghosh, A., Luccioni, A.S., Pistilli, G.: Fully autonomous AI agents should not be developed. arXiv preprint arXiv:2502.02649 (2025)

[111] Bajcsy, R.: Active perception. Proceedings of the IEEE **76**(8), 966–1005 (1988)

[112] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems **36** (2023)

[113] Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D., Wang, Y., Tang, R., Chen, E.: Understanding the planning of LLM agents: A survey. arXiv preprint arXiv:2402.02716 (2024)

[114] Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in Games **4**(1), 1–43 (2012)

[115] Atkinson, R.C., Shiffrin, R.M.: Human memory: A proposed system and its control processes. In: Psychology of Learning and Motivation vol. 2, pp. 89–195. Academic Press, New York (1968)

[116] Baddeley, A.D., Hitch, G.: Working memory. In: Psychology of Learning and Motivation vol. 8, pp. 47–89. Academic Press, New York (1974)

[117] Tulving, E.: Episodic and semantic memory. In: Tulving, E., Donaldson, W. (eds.) Organization of Memory, pp. 381–403. Academic Press, New York (1972)

[118] Squire, L.R.: Memory systems of the brain: A brief history and current perspective. Neurobiology of Learning and Memory **82**(3), 171–177 (2004)

[119] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., *et al.*: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)

[120] Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z., Wen, J.-R.: A survey on the memory mechanism of large language model based agents. ACM Transactions on Information Systems **43**(6) (2025)

[121] Rintanen, J.: Planning as satisfiability: Heuristics. AI Magazine **41**(3), 39–56 (2020)

[122] Hu, X., Li, Y., Zhu, J.: Open-set object detection for robust perception. IEEE Transactions on Circuits and Systems for Video Technology **34**(5), 1234–1247 (2024)

[123] Mehta, R., Singh, P.: Continual reinforcement learning in non-stationary environments. AAAI **38**(7), 890–897 (2024)

[124] Nilsson, N.J.: Principles of Artificial Intelligence. Tioga Publishing Company, Palo Alto, CA (1980)

[125] Currie, K., Tate, A.: O-plan: The open planning architecture. In: ECAI, pp. 502–505 (2002)

[126] Williams, B.C., Nayak, P.P.: A model-based approach to reactive self-configuring systems. In: ICAPS, pp. 471–478 (1999)

[127] Kaelbling, L.P., Lozano-Pérez, T., Roy, N.: Integrated task and motion planning in belief space. International Journal of Robotics Research **30**(12), 1435–1470 (2011)

[128] Ge, T., Xiao, B., Liu, S.: Open-world object detection in the wild. In: CVPR (2024)

[129] Chen, L., Kumar, A., Verma, S.: Lifelong learning in deep reinforcement learning. Machine Learning **112**, 345–366 (2023)

[130] Patel, K., Zhang, W.: Safe continual learning for robotics. Robotics and Autonomous Systems **162**, 104305 (2024)

[131] Kormushev, P., Nenchev, D., Calinon, S., Caldwell, D.G.: Upper-body kinesthetic teaching of a free-standing humanoid robot. IEEE Transactions on Robotics **29**(5), 988–1001 (2013)

[132] Zhao, P., Zhang, J., Fan, X.: Safe reinforcement learning under non-stationary dynamics. Journal of Machine Learning Research **22**, 1–30 (2021)

[133] Bonet, B., Geffner, H.: Planning under uncertainty with epistemic representations. ICAPS **34**, 67–75 (2024)

[134] Sadeghi, F., Levine, S.: Cad2rl: Real single-image flight without a single real image. In: RSS (2016)

[135] Li, M., Chen, X.: Robust policy improvement in dynamic crowds. Autonomous Robots **49**(1), 101–115 (2025)

[136] Yu, K., Zhao, L.: Agile drone racing via continual reinforcement learning. IEEE Transactions on Vehicular Technology **74**(3), 1456–1468 (2025)

[137] Marr, D.: Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information. W. H. Freeman, San Francisco (1982)

[138] Benson-Tilsen, T., Soares, N.: Formalizing convergent instrumental goals. In: AAAI Workshop: AI, Ethics, and Society (2016)

[139] Chan, A., Salganik, R., Markelius, A., Pang, C., Rajkumar, N., Krasheninnikov, D., Langosco, L., He, Z., Duan, Y., Carroll, M., *et al.*: Harms from increasingly agentic algorithmic systems. In: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT), pp. 651–666 (2023)

[140] Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Transactions on Big Data **7**(3), 535–547 (2019)

[141] Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(4), 824–836 (2018)

[142] Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S.G., Stoica, I., Gonzalez, J.E.: MemGPT: Towards LLMs as operating systems. arXiv preprint arXiv:2310.08560 (2023)

[143] Coulom, R.: Efficient selectivity and backup operators in monte-carlo tree search. In: International Conference on Computers and Games, pp. 72–83 (2006). Springer

[144] Liu, B., Jiang, Y., Zhang, X., Liu, Q., Zhang, S., Biswas, J., Stone, P.: LLM+P: Empowering large language models with optimal planning proficiency. arXiv preprint arXiv:2304.11477 (2023)

[145] Perez, D., Samothrakis, S., Lucas, S., Rohlfshagen, P.: Rolling horizon evolution versus tree search for navigation in single-player real-time games. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO), pp. 351–358 (2013)

[146] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems **36** (2023)

[147] Anthropic AI: Model Context Protocol (MCP). https://modelcontextprotocol.io (2024)

[148] LangChain AI: LangGraph: State Management for Agentic Workflows. https://github.com/langchain-ai/langgraph (2024)

[149] Wu, T., et al.: AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation (2023)

[150] CrewAI Contributors: CrewAI: Multi-Agent Orchestration Framework. https://github.com/joaomdmoura/crewai (2024)

[151] OpenAI: OpenAI Agents SDK. https://github.com/openai/agents (2025)

[152] Microsoft: Semantic Kernel. https://github.com/microsoft/semantic-kernel (2023)

[153] LlamaIndex Team: LlamaIndex: Connecting LLMs with External Data. https://www.llamaindex.ai (2023)

[154] deepset AI: Haystack: Modular NLP Framework. https://haystack.deepset.ai (2021)

[155] Alibaba DAMO Academy: AgentScope: Multi-Agent Development Framework. https://github.com/modelscope/agentscope (2024)

[156] Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P.R., Salazar, G., Ryoo, M.S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.-W.E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N.J., Irpan, A., Ichter, B., Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K.A., Driess, D., Ding, T., Choromanski, K.M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Gonzalez Arenas, M., Han, K.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Tan, J., Toussaint, M., Darvish, K. (eds.) Proceedings of the 7th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 229, pp. 2165–2183. PMLR, Atlanta, Georgia, USA (2023). https://proceedings.mlr.press/v229/zitkovich23a.html

[157] Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., *et al.*: Competition-level code

generation with alphacode. Science **378**(6624), 1092–1097 (2022)

[158] Shazeer, N.: Fast transformer decoding: One write-head is all you need. arXiv preprint arXiv:1911.02150 (2019)

[159] Ma, Y., Gou, Z., Hao, J., Xu, R., Wang, S., Pan, L., Yang, Y., Cao, Y., Sun, A.: SciAgent: Tool-augmented language models for scientific reasoning. In: Al-Onaizan, Y., Bansal, M., Chen, Y.-N. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 15701–15736. Association for Computational Linguistics, Miami, Florida, USA (2024). https://doi.org/10.18653/v1/2024.emnlp-main.880

[160] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., Steinhardt, J.: Measuring mathematical problem solving with the MATH dataset. In: Advances in Neural Information Processing Systems 34: Datasets and Benchmarks Track (2021)

[161] Liu, X., Xu, N., Chen, M., Xiao, C.: Autodan: Generating stealthy jailbreak prompts on aligned large language models. In: The Twelfth International Conference on Learning Representations (ICLR) (2024)

[162] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J.Z., Fredrikson, M.: Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:2307.15043 (2023)

[163] Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality (2023). https://lmsys.org/blog/2023-03-30-vicuna/

[164] Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. Advances in neural information processing systems **36**, 10088–10115 (2023)

[165] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)

[166] Arif, S., Arif, T., Haroon, M.S., Khan, A.J., Raza, A.A., Athar, A.: The art of storytelling: Multi-agent generative ai for dynamic multimodal narratives. arXiv preprint arXiv:2409.11261 (2024)

[167] Huot, F., Amplayo, R.K., Palomaki, J., Jakobovits, A.S., Clark, E., Lapata, M.: Agents' room: Narrative generation through multi-step collaboration. In: The Thirteenth International Conference on Learning Representations (ICLR) (2025)

[168] Xu, Z., Wang, X., Li, S., Yu, T., Wang, L., Fu, Q., Yang, W.: Agents play thousands of 3d video games. arXiv preprint arXiv:2503.13356 (2025)

[169] Wang, Y., Le, H., Gotmare, A., Bui, N., Li, J., Hoi, S.: CodeT5+: Open code large language models for code understanding and generation. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 1069–1088. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.68

[170] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

[171] Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., Zhou, M.: CodeBERT: A pre-trained model for programming and natural languages. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1536–1547. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.139

[172] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.*: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[173] Lu, S., Guo, D., Ren, S., Huang, J., Svyatkovskiy, A., Blanco, A., Clement, C., Drain, D., Jiang, D., Tang, D., Li, G., Zhou, L., Shou, L., Zhou, L., Tufano, M., Gong, M., Zhou, M., Duan, N., Sundaresan, N., Deng, S.K., Fu, S., Liu, S.: Codexglue: A machine learning benchmark dataset for code understanding and generation. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. Neural Information Processing Systems, Virtual (2021)

[174] Ahmad, W., Chakraborty, S., Ray, B., Chang, K.-W.: Unified pre-training for program understanding and generation. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2655–2668. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-main.211 . https://aclanthology.org/2021.naacl-main.211/

[175] Wang, Y., Wang, W., Joty, S., Hoi, S.C.H.: CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In: Moens, M.-F., Huang, X., Specia, L., Yih, S.W.-t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8696–8708. Association for Computational Linguistics, Online and Punta Cana,

Dominican Republic (2021). https://doi.org/10.18653/v1/2021.emnlp-main.685 . https://aclanthology.org/2021.emnlp-main.685/

[176] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Advances in neural information processing systems **27** (2014)

[177] Erdogan, L.E., Lee, N., Kim, S., Moon, S., Furuta, H., Anumanchipalli, G., Keutzer, K., Gholami, A.: Plan-and-act: Improving planning of agents for long-horizon tasks. In: Proceedings of the 42nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 267, pp. 12985–13010. PMLR, Vancouver, Canada (2025). https://proceedings.mlr.press/v267/erdogan25a.html

[178] Lai, H., Liu, X., Iong, I.L., Yao, S., Chen, Y., Shen, P., Yu, H., Zhang, H., Zhang, X., Dong, Y., *et al.*: Autowebglm: A large language model-based web navigating agent. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 5295–5306 (2024)

[179] Wang, X., Chen, Y., Zhu, W.: A survey on curriculum learning. IEEE transactions on pattern analysis and machine intelligence **44**(9), 4555–4576 (2021)

[180] Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems **36**, 53728–53741 (2023)

[181] Yuan, Z., Yuan, H., Li, C., Dong, G., Lu, K., Tan, C., Zhou, C., Zhou, J.: Scaling relationship on learning mathematical reasoning with large language models. arXiv preprint arXiv:2308.01825 (2023)

[182] Fragiadakis, G., Diou, C., Kousiouris, G., Nikolaidou, M.: Evaluating human-ai collaboration: A review and methodological framework. arXiv preprint arXiv:2407.19098 (2024)

[183] Xu, C., Cho, S.-E.: Factors affecting human–ai collaboration performances in financial sector: Sustainable service development perspective. Sustainability **17**(10), 4335 (2025)

[184] Abdelrahman, G., Wang, Q., Nunes, B.: Knowledge tracing: A survey. ACM Computing Surveys **55**(11), 1–37 (2023)

[185] Wang, H., Tlili, A., Huang, R., Cai, Z., Li, M., Cheng, Z., Yang, D., Li, M., Zhu, X., Fei, C.: Examining the applications of intelligent tutoring systems in real educational contexts: A systematic literature review from the social experiment perspective. Education and information technologies **28**(7), 9113–9148 (2023)

[186] García-Méndez, S., Arriba-Pérez, F., Somoza-López, M.d.C.: A review on the use of large language models as virtual tutors: A review on the use of large language

[187] Pal Chowdhury, S., Zouhar, V., Sachan, M.: Autotutor meets large language models: A language model tutor with rich pedagogy and guardrails. In: Proceedings of the Eleventh ACM Conference on Learning@ Scale, pp. 5–15 (2024)

[188] Liu, J., Huang, Z., Xiao, T., Sha, J., Wu, J., Liu, Q., Wang, S., Chen, E.: Socraticlm: Exploring socratic personalized teaching with large language models. Advances in Neural Information Processing Systems **37**, 85693–85721 (2024)

[189] Reddig, J.M., Arora, A., MacLellan, C.J.: Generating in-context, personalized feedback for intelligent tutors with large language models. International Journal of Artificial Intelligence in Education, 1–42 (2025)

[190] Weitekamp, D., N. Siddiqui, M., J. MacLellan, C.: Tutorgym: A testbed for evaluating ai agents as tutors and students. In: International Conference on Artificial Intelligence in Education, pp. 361–376 (2025). Springer

[191] Lyu, B., Li, C., Li, H., Oh, H., Song, Y., Zhu, W., Xing, W.: Exploring the role of teachable ai agents' personality traits in shaping student interaction and learning in mathematics education. In: Proceedings of the 15th International Learning Analytics and Knowledge Conference, pp. 887–894 (2025)

[192] Fu, J., Zhao, X., Yao, C., Wang, H., Han, Q., Xiao, Y.: Reward shaping to mitigate reward hacking in rlhf. In: ICML 2025 Workshop on Reliable and Responsible Foundation Models (2025). https://openreview.net/forum?id=Q5uUkoTw3J

[193] Zhou, S., Xu, F.F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., Neubig, G.: Webarena: A realistic web environment for building autonomous agents. In: The Twelfth International Conference on Learning Representations (ICLR) (2024)

[194] 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.: Agentbench: Evaluating llms as agents. In: The Twelfth International Conference on Learning Representations (ICLR) (2024)

[195] Jimenez, C.E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., Narasimhan, K.: SWE-bench: Can language models resolve real-world GitHub issues? In: The Twelfth International Conference on Learning Representations (2024)

[196] Yehudai, A., Eden, L., Li, A., Uziel, G., Zhao, Y., Bar-Haim, R., Cohan, A., Shmueli-Scheuer, M.: Survey on evaluation of LLM-based agents. arXiv preprint arXiv:2503.16416 (2025)

[197] Bang, Y., Ji, Z., Schelten, A., Hartshorn, A., Fowler, T., Zhang, C., Cancedda, N., Fung, P.: HalluLens: LLM hallucination benchmark. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 24128–24156. Association for Computational Linguistics, Vienna, Austria (2025). https://doi.org/10.18653/v1/2025.acl-long.1176 . https://aclanthology.org/2025.acl-long.1176/

[198] Searle, J.R.: Minds, brains, and programs. Behavioral and Brain Sciences **3**(3), 417–424 (1980)

[199] Bender, E.M., Koller, A.: Climbing towards NLU: On meaning, form, and understanding in the age of data. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 5185–5198 (2020)

[200] Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: Can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT), pp. 610–623 (2021)

[201] Wang, X., Hu, Z., Lu, P., Zhu, Y., Zhang, J., Subramaniam, S., Loomba, A.R., Zhang, S., Sun, Y., Wang, W.: Scibench: Evaluating college-level scientific problem-solving abilities of large language models. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research. PMLR, Vienna, Austria (2024)

[202] Li, C., Jain, V., Nakkiran, P.: A simple yet challenging benchmark for browsing agents. arXiv preprint arXiv:2504.12516 (2025)

[203] Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., Su, Y.: MIND2WEB: Towards a generalist agent for the web. In: Proceedings of the 37th Conference on Neural Information Processing Systems (Datasets and Benchmarks Track) (2023)

[204] Jing, L., Huang, Z., Wang, X., Yao, W., Yu, W., Ma, K., Zhang, H., Du, X., Yu, D.: Dsbench: How far are data science agents from becoming data science experts? In: The Thirteenth International Conference on Learning Representations (ICLR) (2025)

[205] Ma, Z., Zhang, B., Zhang, J., *et al.*: Spreadsheetbench: Towards challenging real-world spreadsheet manipulation. In: Proceedings of the 41st International Conference on Machine Learning (2024)

[206] Epoch AI: FrontierMath: Benchmarking AI on Expert-Level Mathematics. https://epoch.ai/frontiermath. Accessed August 3, 2025 (2024)

[207] Styles, O., Xu, M., Li, S., *et al.*: WorkBench: A benchmark dataset for agents

in a realistic workplace setting. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (2024)

[208] Mialon, G., Fourrier, C., Swift, C., *et al.*: GAIA: A benchmark for general ai assistants. In: Advances in Neural Information Processing Systems (2023)

[209] Xu, Y., Patil, S., Wei, Y., *et al.*: ToolBench: Generating and evaluating tool-augmented tasks for language models. In: Proceedings of the 30th ACM SIGKDD Conference (2023)

[210] Zhuang, J., Zhou, H., Zhang, M., *et al.*: ToolQA: A dataset for question answering with external tools. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (2023)

[211] Li, X., Chen, R., Liu, Q., *et al.*: API-Bank: A benchmark for tool-augmented llms with thousands of real-world apis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2023)

[212] Liu, X., Zhang, T., Gu, Y., *et al.*: Visualagentbench: Towards large multimodal models as visual foundation agents. In: International Conference on Learning Representations (2024)

[213] Zhu, K., Du, H., Hong, Z., *et al.*: Multiagentbench: Evaluating the collaboration and competition of llm agents. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (2025)

[214] Wang, S., Narasimhan, K., Yao, S., *et al.*: Scienceworld: Is your agent smarter than a 5th grader? In: Advances in Neural Information Processing Systems (2022)

[215] Ma, C., Zhang, J., Zhu, Z., Yang, C., Yang, Y., Jin, Y., Lan, Z., Kong, L., He, J.: AgentBoard: An analytical evaluation board of multi-turn LLM agents. In: Advances in Neural Information Processing Systems (Datasets and Benchmarks Track) (2024)

[216] Yao, S., Shinn, N., Razavi, P., Narasimhan, K.R.: τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In: The Thirteenth International Conference on Learning Representations (ICLR) (2025)

[217] Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., Kambhampati, S.: PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In: Advances in Neural Information Processing Systems, vol. 36, pp. 38975–38987 (2023)

[218] Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K.: Let's verify step by step. In: The Twelfth International Conference on Learning Representations (2024)

[219] Ruan, Y., Dong, H., Wang, A., Pitis, S., Zhou, Y., Ba, J., Dubois, Y., Maddison, C.J., Hashimoto, T.: Identifying the risks of LM agents with an LM-emulated sandbox. In: The Twelfth International Conference on Learning Representations (2024)

[220] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., *et al.*: Guidelines for human-AI interaction. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2019)

[221] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., *et al.*: Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: Advances in Neural Information Processing Systems, vol. 36, pp. 46595–46623 (2023)

[222] Zhuge, M., Zhao, C., Ashley, D.R., Wang, W., Khizbullin, D., Xiong, Y., Liu, Z., Chang, E., Krishnamoorthi, R., Tian, Y., Shi, Y., Chandra, V., Schmidhuber, J.: Agent-as-a-judge: Evaluate agents with agents. In: Proceedings of the 42nd International Conference on Machine Learning. PMLR, vol. 267, pp. 80569–80611 (2025)

[223] Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., et al.: Toward trustworthy AI development: Mechanisms for supporting verifiable claims. arXiv preprint arXiv:2004.07213 (2020)

[224] Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

[225] Seshia, S.A., Sadigh, D., Sastry, S.S.: Toward verified artificial intelligence. Communications of the ACM 65(7), 46–55 (2022)

[226] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference (TCC), pp. 265–284 (2006). Springer

[227] Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 3645–3650 (2019)

[228] Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green AI. Communications of the ACM **63**(12), 54–63 (2020)

[229] Shavit, Y., Agarwal, S., Brundage, M., Adler, S., O'Keefe, C., Campbell, R., Lee, T., Mishkin, P., Eloundou, T., Hickey, A., et al.: Practices for governing agentic ai systems. Research Paper, OpenAI (2023)

[230] Kasirzadeh, A., Gabriel, I.: Characterizing ai agents for alignment and governance. arXiv preprint arXiv:2504.21848 (2025)

[231] Scassellati, B.: Theory of mind for a humanoid robot. Autonomous Robots **12**(1), 13–24 (2002)

[232] Baker, C.L., Jara-Ettinger, J., Saxe, R., Tenenbaum, J.B.: Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. Nature Human Behaviour **1**(4), 0064 (2017)

[233] Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S.A., Botvinick, M.: Machine theory of mind. In: International Conference on Machine Learning, pp. 4218–4227 (2018). PMLR

[234] Newell, A.: Unified Theories of Cognition. Harvard University Press, Cambridge, MA (1994)

[235] Fodor, J.A.: The Modularity of Mind. MIT Press, Cambridge, MA (1983)

[236] Coecke, B.: In: Plotnitsky, A., Haven, E. (eds.) Compositionality as We See It, Everywhere Around Us, pp. 247–267. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-12986-5_12

[237] Si, C., Yang, D., Hashimoto, T.: Can LLMs generate novel research ideas? a large-scale human study with 100+ nlp researchers. In: The Thirteenth International Conference on Learning Representations (ICLR) (2025)

[238] Su, H., Chen, R., Tang, S., Zheng, X., Li, J., Yin, Z., Ouyang, W., Dong, N.: Two heads are better than one: A multi-agent system has the potential to improve scientific idea generation. arXiv preprint arXiv:2410.09403 (2024)

[239] Cai, W., Jiang, J., Wang, F., Tang, J., Kim, S., Huang, J.: A survey on mixture of experts in large language models. IEEE Transactions on Knowledge and Data Engineering **37**(7), 3896–3915 (2025) https://doi.org/10.1109/TKDE.2025.3554028

[240] Sandini, G., Sciutti, A., Morasso, P.: Artificial cognition vs. artificial intelligence for next-generation autonomous robotic agents. Frontiers in Computational Neuroscience **18**, 1349408 (2024)

[241] Jiang, F., Pan, C., Wang, K., Michiardi, P., Dobre, O.A., Debbah, M.: From large ai models to agentic ai: A tutorial on future intelligent communications. IEEE Journal on Selected Areas in Communications **44**, 3507–3540 (2026) https://doi.org/10.1109/JSAC.2026.3660010

[242] Nguyen, D.T., Kumar, A., Lau, H.C.: Credit assignment for collective multiagent rl with global rewards. Advances in neural information processing systems **31**

(2018)

[243] Clatterbuck, H., Castro, C., Morán, A.M.: Risk alignment in agentic ai systems. arXiv preprint arXiv:2410.01927 (2024)

[244] Kaur, D.P., Singh, N.P., Banerjee, B.: A review of platforms for simulating embodied agents in 3d virtual environments. Artificial Intelligence Review **56**(4), 3711–3753 (2023)

[245] Zhong, B., Cao, H., Zamani, M., Caccamo, M.: Towards safe ai: Sandboxing dnns-based controllers in stochastic games. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 15340–15349 (2023)

[246] Zhou, Y., Lou, J., Huang, Z., Qin, Z., Yang, S., Wang, W.: Don't say no: Jailbreaking LLM by suppressing refusal. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Findings of the Association for Computational Linguistics: ACL 2025, pp. 25224–25249. Association for Computational Linguistics, Vienna, Austria (2025). https://doi.org/10.18653/v1/2025.findings-acl.1294 . https://aclanthology.org/2025.findings-acl.1294/

[247] Zhao, W., Queralta, J.P., Westerlund, T.: Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 737–744 (2020). IEEE

[248] Collins, J., Chand, S., Vanderkop, A., Howard, D.: A review of physics simulators for robotic applications. IEEE Access **9**, 51416–51431 (2021)

[249] Eversberg, L., Lambrecht, J.: Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization. Sensors **21**(23), 7901 (2021)

[250] Ajani, O.S., Hur, S.-h., Mallipeddi, R.: Evaluating domain randomization in deep reinforcement learning locomotion tasks. Mathematics **11**(23), 4744 (2023)

[251] Batty, M.: Digital twins in city planning. Nature Computational Science **4**(3), 192–199 (2024)

[252] Li, W., Deka, D., Wang, R., Paternina, M.R.A.: Physics-constrained adversarial training for neural networks in stochastic power grids. IEEE Transactions on Artificial Intelligence **5**(3), 1121–1131 (2023)

[253] Feffer, M., Sinha, A., Deng, W.H., Lipton, Z.C., Heidari, H.: Red-teaming for generative ai: Silver bullet or security theater? In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, vol. 7, pp. 421–437 (2024)

[254] Abeysirigoonawardena, Y., Xie, K., Chen, C., Khorasgani, S.H., Chen, R., Wang,

R., Shkurti, F.: Generating transferable adversarial simulation scenarios for self-driving via neural rendering. In: Tan, J., Toussaint, M., Darvish, K. (eds.) Proceedings of the 7th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 229, pp. 3710–3731. PMLR, Atlanta, Georgia, USA (2023). https://proceedings.mlr.press/v229/abeysirigoonawardena23a.html

[255] Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135 (2017). PMLR

[256] Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and brain sciences **40**, 253 (2017)

[257] Hammond, K.J., Leake, D.B.: Large language models need symbolic ai. In: NeSy, pp. 204–209 (2023)

[258] Wu, X., Li, Y.-L., Sun, J., Lu, C.: Symbol-llm: leverage language models for symbolic system in visual human activity reasoning. Advances in neural information processing systems **36**, 29680–29691 (2023)

[259] Sansford, H., Richardson, N., Maretic, H.P., Saada, J.N.: Grapheval: A knowledge-graph based llm hallucination evaluation framework. In: Proceedings of the Workshop on Knowledge-infused Learning (KiL 2024), Co-located with the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). CEUR Workshop Proceedings, vol. 3894. CEUR-WS, Barcelona, Spain (2024)

[260] Aeronautiques, C., Howe, A., Knoblock, C., McDermott, I.D., Ram, A., Veloso, M., Weld, D., Sri, D.W., Barrett, A., Christianson, D., et al.: Pddl—the planning domain definition language. Technical Report, Tech. Rep. (1998)

[261] Velasquez, A., Bhatt, N., Topcu, U., Wang, Z., Sycara, K., Stepputtis, S., Neema, S., Vallabha, G.: Neurosymbolic ai as an antithesis to scaling laws. PNAS nexus **4**(5), 117 (2025)