

Review

Not peer-reviewed version

A Comprehensive Survey of the LLM-Based Agent: The Contextual Cognition Perspective

Mingyue Cheng , Daoyu Wang , Shuo Yu , Qingchuan Li , Jie Ouyang , Yucong Luo , Yiju Zhang , Qi Liu * , Enhong Chen

Posted Date: 14 April 2026

doi: 10.20944/preprints202604.0935.v1

Keywords: contextual cognition; LLM agent; agentic AI



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Review

A Comprehensive Survey of the LLM-Based Agent: The Contextual Cognition Perspective

Mingyue Cheng, Daoyu Wang, Shuo Yu, Qingchuan Li, Jie Ouyang, Yucong Luo, Yiju Zhang, Qi Liu * and Enhong Chen

State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China

* Correspondence: qiliuql@ustc.edu.cn

Abstract

Large language model (LLM)-based agents have given rise to phenomenal applications (e.g., Open-Claw, Claude Code), transitioning from fixed text processing to complex task execution. However, most existing works conceptualize the LLM-based agent by decomposing the whole system into modules such as planning, action, reflection, and memory, thereby lacking a unified perspective to explain the emergence of agentic intelligence. In this survey, we present a novel perspective by framing agentic intelligence through the lens of contextual cognition. We propose that an advanced agent fundamentally relies on a unified framework comprising four core processes: contextual encoding, perception, interaction, and reasoning. Within this framework, we reveal that the emergence of agentic intelligence stems not merely from the organization of diverse modules, but from how the agent manages and, especially, interacts with contextuality, where contextuality is defined as the dynamic integration of external observations and the LLMs' internal states. Furthermore, we systematically review current methods for constructing agents from the contextual cognition perspective, encompassing agent runtime orchestration and foundation LLM training. We also revisit corresponding benchmarks and applications, such as deep research, coding, GUI, and scientific agents. Finally, we discuss critical open challenges and outline future research trends, providing a roadmap for overcoming current cognitive bottlenecks and fostering contextualized agentic systems. We hope this perspective serves as an alternative framework to analyze agent construction through contextual cognition and guide the future development of LLM-based agents.

Keywords: contextual cognition; LLM agent; agentic AI

1. Introduction

"Without context, ... actions have no meaning at all."

— Gregory Bateson

Large language model (LLM)-based agents have given rise to phenomenal applications such as OpenClaw and Claude Code [1,2], transitioning from fixed text processing to complex task execution [3,4]. Currently, they demonstrate proficiency across diverse domains, including software engineering [5], scientific discovery [6], web navigation [7], and embodied control [8]. As a result, understanding how LLM agents work and how their intelligence emerges become an increasingly important research question [9–11].

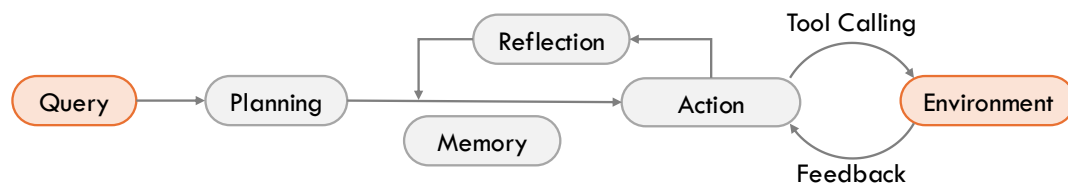
Several recent surveys attempt to systematize this field [12–14]. Most characterize LLM agents through modular architectures comprising planning, action, reflection, and memory [6,15]. While these works offer a clear taxonomy of functional capabilities, they often rely on a prevalent assumption: agent capacity emerges from stacking additional modules or optimizing components in isolation [16,17]. Although this perspective is sufficient for constructing static engineering frameworks, it falls short in explaining why agents fail to generalize in dynamic, open-ended environments. Real-world environments exhibit dynamics and ambiguity, requiring agents to make decisions by integrating evolving

contextual signals and accumulated contextual information [18,19]. In such settings, the simple integration of modules does not guarantee that the agent can understand the evolving situational environments [20–22].

This leads to a deeper question that current surveys do not answer: *Where does agentic intelligence actually emerge?* Despite capabilities in tool use, memory construction, and long-trace reasoning, why do agents still fail at tasks requiring self-adaptation, long-horizon coordination, or mistake recovery? We argue that it expose a critical gap: what agents lack is not another module, but the capacity to manage *contextuality* as tasks unfold. We strictly define contextuality as the dynamic integration of external observations and the LLMs' internal states, distinguishing it from static prompt windows or external environments. This dynamic contextuality governs signal interpretation, decision-making, and long-horizon stability, yet it remains overlooked in paradigms that prioritize the mere organization of diverse modules.

To bridge this gap, we introduce the perspective of contextual cognition. As shown in Figure 1, rather than defining an agent as a mere organization of modules, we characterize it as a system that manages contextuality through an iterative cycle of encoding, perception, interaction, and reasoning. Compared to a modern computer, the LLM acts as the central CPU, engaging with the outside world through tools like sandboxes. While prompts remain the underlying carrier of interaction, this process differs essentially from single-turn question-answering due to its high degree of dynamism and adaptability. It requires contextual information to be digitized, semanticized, and capable of forming an closed-loop process. Consequently, we argue that agentic intelligence emerges from the capacity to manage the contextuality.

(a) Traditional Perspective: Module-Centric LLM Agents



(b) Our Perspective: Contextuality-Centric LLM Agents

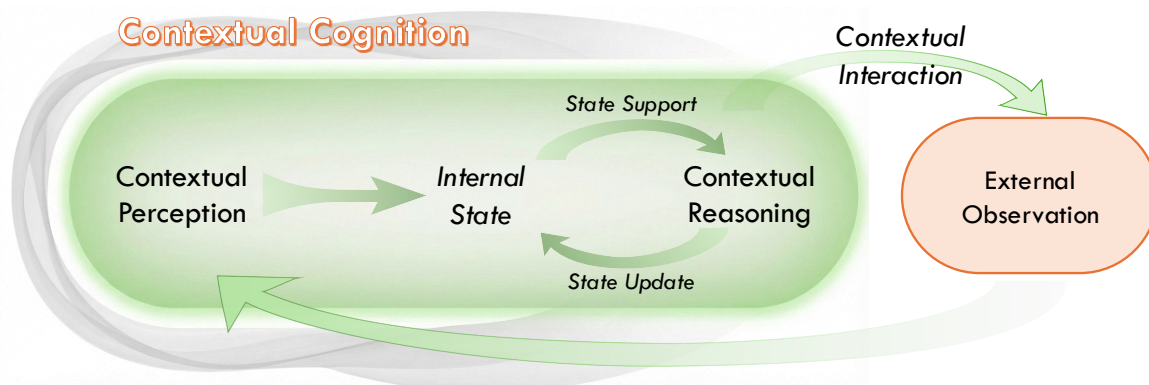


Figure 1. Comparison of traditional module-centric and proposed contextuality-centric agent perspectives. Our *contextuality* represents the dynamic integration of an agent's internal state with its external observations.

Based on this perspective, we propose a unified framework structured around four interdependent processes: (1) contextual encoding, which involves the formal presentation of contextuality; (2) contextual perception, where the agent identifies relevant signals from external observations; (3) contextual interaction, through which the agent leverages these observations to dynamically update its internal states; and (4) contextual reasoning, which guides decision-making based on the formed contextuality. This integrated loop enables coherent understanding and adaptive behavior, clarifying

why the mere organization of modules is insufficient. It highlights that agentic intelligence relies on the continuous, interaction-driven refinement of contextuality.

Guided by this framework, we review the recent progress of LLM-based agents. We analyze agent construction mechanisms by categorizing current methods into agent runtime and foundation LLM training, examining how each manages the contextual lifecycle. Furthermore, we revisit corresponding benchmarks and applications, such as deep research, coding, GUI, and scientific agents, to evaluate varying task demands on contextual interaction. Finally, we identify critical open challenges and provide a roadmap for overcoming cognitive bottlenecks and fostering contextualized agentic systems. By shifting the focus from the mere organization of modules to the cognitive dynamics of contextuality, this survey establishes an alternative view to analyze and drive the future development of LLM-based agents.

The remainder of this paper is organized as follows. Section 2 reviews recent progress in LLM agents, while Section 3 articulates the necessity of the contextual cognition perspective to address current technical bottlenecks. Section 4 formalizes our unified framework, detailing the cognitive closed loop. This framework is then applied in Section 5 to analyze construction mechanisms like agent runtime and foundation LLM training. Section 6 evaluates framework utility across corresponding benchmarks and applications. Finally, Sections 7 and 8 discuss open challenges, outline future trends, and summarize our contributions.

2. Background

As shown in Figure 2, the evolution of LLM agents represents a paradigm shift from static text generation to autonomous agents capable of navigating complex contextuality. This section delineates the modern conceptualization of agents through the lens of contextual cognition and critically reviews the burgeoning survey literature to position our unified framework as a departure from traditional architectural modularity.

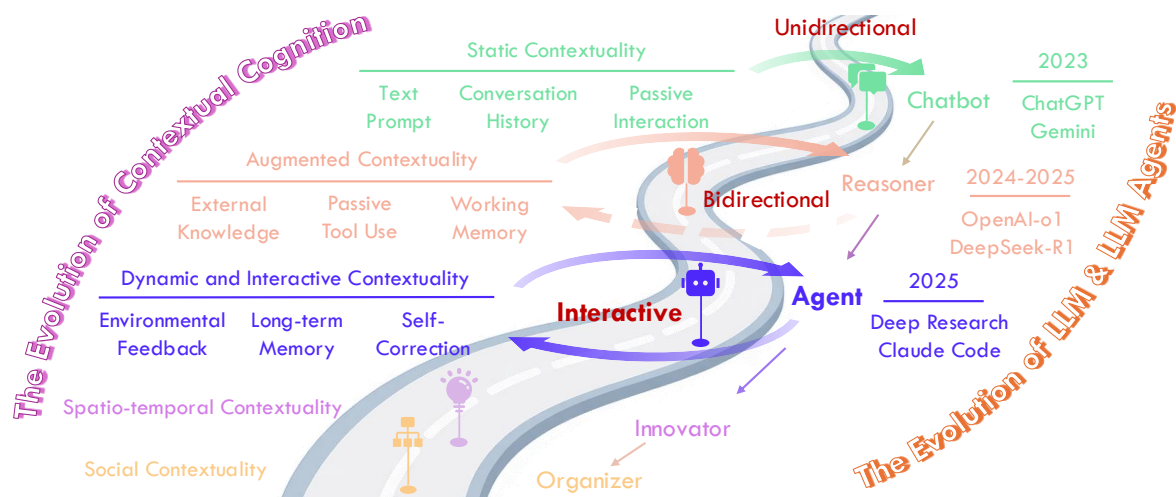


Figure 2. The evolution of LLM Agents is fundamentally a journey of deepening contextual cognition. This roadmap illustrates the shift from static context, where early Chatbots passively process text history, to augmented context, where Reasoners began utilizing external tools and working memory. We are now entering the era of dynamic and interactive context, where Agents actively engage in environmental feedback loops and self-correction.

2.1. Concept of LLM Agents

The conceptualization of agents has evolved from classical entities acting upon environments to complex systems driven by sophisticated and dynamic agentic workflows [23]. In this evolving landscape, intelligence emerges not from static prompting but from the autonomous shaping of contextuality, as systems actively integrate perception, interaction, representation, and reasoning to navigate the unstructured nature of real-world complexity [24,25]. Rather than relying solely on scalar

rewards, modern agents leverage interaction and verbal feedback to refine their cognitive loops and strategies [13,26]. This unified framework, supported by versatile agentic platforms like OpenClaw [2], encompasses diverse manifestations ranging from symbol-manipulating language agents [27,28] to embodied systems, since both require grounded reasoning that aligns dynamic internal representations with specific external affordances.

2.2. Existing Surveys

The rapid proliferation of agent research has produced a comprehensive body of literature, which we categorize into four distinct streams: General Architectures, Agentic Learning, Memory Systems, and Domain Applications. A comparative summary is provided in Table 1. While these works offer valuable structural taxonomies, they predominantly view agent capabilities in isolation, treating the agent as a mere aggregation of modules.

Table 1. Comparison of recent surveys on LLM agents (2024–2025) highlighting the gap in contextual cognition.

Key Focus	Representative Work	Year	Categories	Perspective on Contextuality
Comprehensive Architecture	Xi et al. [12]	2023	General Arch.	Viewed primarily as static prompt/input
Module Construction	Wang et al. [18]	2023	General Arch.	Viewed primarily as static prompt/input
Multi-Agent Collaboration	Tran et al. [10]	2025	General Arch.	Limited to conversational history
MAS Progress	Chawla et al. [19]	2024	General Arch.	Limited to conversational history
Long-term Optimization	Zhang et al. [8]	2025	Agentic Learning	Focus on scalar rewards over context
Self-Evolution	Gao et al. [9]	2025	Agentic Learning	Context treated as training data
Planning	Huang et al. [7]	2024	Memory Systems	Focus on storage rather than interaction
Retrieval Augmentation	Zhu et al. [20]	2023	Memory Systems	Focus on storage rather than interaction
Evaluation	Hassan et al. [11]	2025	Domain Apps.	Identifies context loss in long turns
Scientific Discovery	Ren et al. [4]	2025	Domain Apps.	Context as domain-specific knowledge
Contextual Cognition	This Work	2025	Unified Framework	Contextuality as the core

As illustrated in Table 1, foundational surveys initially established the standard design of agents, breaking them down into separate components like profile, memory, and planning [17,24]. More recent reviews have expanded this scope to Multi-Agent Systems, analyzing how agents cooperate or debate to solve problems [15,29]. However, these studies predominantly focus on the arrangement of agents or the history of their messages. They tend to overlook the shared cognitive contextuality, which effectively serves as the underlying understanding that allows agents to interact meaningfully beyond simple text exchange.

Parallel advancements in Agentic Reinforcement Learning and Self-Evolving Agents have shifted attention toward long-term optimization [13,14]. These works discuss how agents improve through

feedback or even update their own prompts and parameters. Yet, the primary focus remains on optimizing rewards or training data, often neglecting how the agent perceives and interprets the changing situation during the task. Similarly, surveys on memory and retrieval systems [12,30] typically treat contextuality as a static storage problem, focusing on how to save and retrieve tokens, rather than how agents actively use that information to drive reasoning. Even in high-stakes domains like science or finance [6,31], contextuality is frequently reduced to fixed domain knowledge or data streams.

A critical gap remains across this literature. Existing studies largely treat contextuality as a passive input window, a database to be queried, or an isolated external environment. In contrast, this survey introduces the perspective of contextual cognition. We strictly define contextuality not as a static prompt, but as the dynamic integration of an agent's internal state and external observations. Rather than framing agent operations as sophisticated single-turn question-answering, we argue that contextuality must be digitized, semanticized, and driven by an iterative, closed-loop cycle of encoding, perception, interaction, and reasoning, which establishes a unified foundation for understanding how agent intelligence fundamentally emerges from continuous contextual optimization.

3. Motivation

LLM-based agents are now entering tasks that require long-term planning, tool use, multi-step decision-making, and active interaction with changing environments [12,24,32]. While these systems have become more capable, they often exhibit unstable, inconsistent, or incoherent behavior when operating in real tasks. These failures reveal a deeper issue: current agents process information, but they do not truly form or maintain contextuality as the task unfolds. To understand why this happens and how to improve agent intelligence, we revisit the cognitive basis of contextuality and introduce a perspective centered on contextual cognition [33].

3.1. *Why Contextuality Matters for LLM-Based Agents*

Intelligent behavior fundamentally depends on contextuality. This concept has deep roots in cognitive science, psychology, and artificial intelligence. Contextual cognition shows that thinking emerges from the coupling between an agent and its environment. Understanding is shaped by how agents interpret and act within the world rather than by internal computation alone. Ecological approaches similarly highlight perception-action loops and environmental affordances that define what an agent can do in a specific setting. Classical AI adds a complementary insight [33]. Coherent behavior under uncertainty requires an internal representation of contextuality, such as a world model that integrates past experiences, current observations, and future goals. These perspectives indicate that contextuality is not merely background information, but a core part of how intelligent systems perceive, decide, and act [34].

Most current LLM-based agents use modular pipelines that treat intelligence as the sum of independent components. Planners, memory modules, and reflection routines are wired together, leaving the formation and updating of contextuality implicit. While this module-centric view works for static tasks, dynamic environments expose failures like losing goals, misinterpreting results, or contextuality collapse. As outlined in Table 2, the evolution of agent frameworks has shifted from prompt [35] and context engineering [36] toward harness engineering [37], where contextual cognition becomes the core focus. In earlier stages, interaction was treated merely as an interface layer. Our perspective treats contextual cognition and interaction as central. Agents must form, maintain, and update contextuality through ongoing environmental interaction to behave adaptively over long horizons. This motivates the layered definition of contextuality in the following subsections.

Table 2. Evolution of AI Agent Paradigms.

Agent Paradigm	Core Problem	Applications
Prompt Engineering	Prompt Optimization	ChatGPT, Claude
Context Engineering	Context Augmentation	Perplexity, Copilot
Harness Engineering	Contextual Cognition	OpenClaw, Claude Code

3.2. Definition of Contextuality in LLM-Based Agents

Instead of viewing LLM agents merely as tool-calling systems, we formalize them as systems for *sequential decision-making and search optimization under resource constraints*. This distinguishes them from classical Reinforcement Learning (RL), a learning-centric paradigm focused on *policy optimization*. Unlike RL, which amortizes decisions into a fixed policy, LLM agents are search-centric, dynamically constructing and optimizing multi-step trajectories during inference.

Formally, let an interaction trajectory be denoted as $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$, where s_t represents the environment state and a_t is the action generated by the agent. We define the agentic process as an objective to find an optimal trajectory τ^* that maximizes the expected utility $U(\tau)$, subject to an inference resource budget B (e.g., task constraints):

$$\tau^* = \arg \max_{\tau \in \mathcal{T}} \mathbb{E}[U(\tau)] \quad \text{s.t.} \quad a_t \sim \pi_{\text{LLM}}(\cdot | \mathcal{C}_t), \quad \sum_{t=0}^T c(a_t) \leq B \quad (1)$$

where $c(a_t)$ is the step-wise inference cost, π_{LLM} denotes the LLM, and \mathcal{C}_t represents the *contextuality* at decision step t .

To clarify these conceptual boundaries, Table 3 contrasts our formulation with standard RL. Classical RL focuses on policy optimization, updating model weights (π_θ) to maximize returns based on fully Markovian states (s_t) and environmental primitives. Conversely, LLM agents rely on search optimization. They utilize a pre-trained model (π_{LLM}) as a learned prior to conduct heuristic search within a strict inference budget (B). Furthermore, LLM agents operate in partially observable environments through contextual interactions, where decisions depend on an evolving historical context (\mathcal{C}_t). Thus, the critical challenge shifts from learning static weights during training to dynamically managing context during inference.

Table 3. Conceptual Comparison: RL vs. LLM Agents.

Dimension	RL	LLM Agents
Formulation	Policy Optimization	Search Optimization
State	Fully Markovian (s_t)	Partially Observable (\mathcal{C}_t)
Action	Environmental Primitives	Contextual Interaction
Policy	Learned Weights (π_θ)	Prior (π_{LLM}) with Search
Constraint	Training Efficiency	Inference Budget (B)

As illustrated in Figure 3, contextuality \mathcal{C}_t is not merely a static input prompt or raw memory material. It is the dynamic integration of internal states (\mathcal{I}_t) and external observations (\mathcal{O}_t) during interactive search, formalized as $\mathcal{C}_t = \mathcal{I}_t \oplus \mathcal{O}_t$. These components become true contextuality only when aligned to evaluate trajectories and optimize task success.

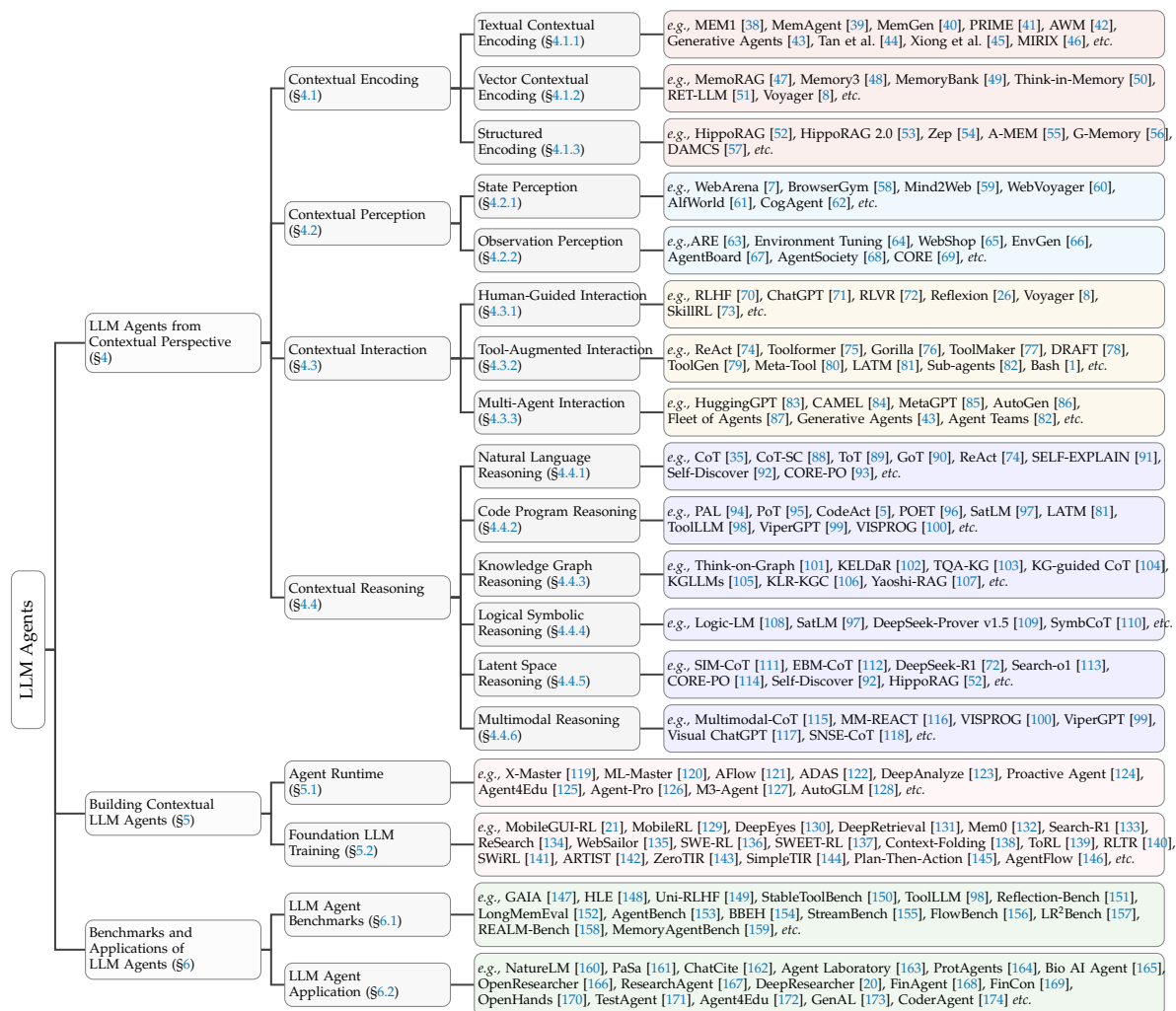


Figure 3. An overview of our proposed taxonomy for LLM-based Agents from a contextual cognition perspective. This figure illustrates the core components (§4), which include contextual encoding, perception, interaction, and reasoning; followed by building methodologies for contextual LLM agents (§5) and the supporting landscape of benchmarks and applications (§6).

Internal States of LLMs (\mathcal{I}_t).

This layer serves as a dynamic cognitive architecture organizing execution variables. It incorporates short-term working memory (e.g., intermediate reasoning traces, chain-of-thought pathways, and subgoal decomposition) as well as persistent episodic records. Furthermore, it integrates meta-level signals like self-estimated confidence and reflection markers to regulate inference costs. Collectively, these elements provide the internal grounding for evaluating explored branches and selecting optimal hybrid actions.

External Observation (\mathcal{O}_t).

External observations constitute the environmental feedback driving sequential decision-making. This layer encompasses direct human instructions, systemic constraints, and dynamic tool outputs, such as ranging from search results to code execution logs and error messages. By integrating historical interactions with these signals, the agent constructs a working view of the partially observable environment, effectively bridging the gap between user objectives and actual state transitions.

The fundamental challenge of this theoretical framework is contextual cognition. Its primary objective is the effective management of this evolving contextuality to navigate the search space within budget constraints.

4. LLM Agents from Contextual Perspective

Following Figure 4, we explore how this management unfolds within the cognitive architecture of LLM agents through four interconnected dimensions: contextual encoding, perception, interaction, and reasoning. We analyze how these components synergize to transform external signals into adaptive actions, establishing a closed-loop framework for grounded decision-making in dynamic environments.

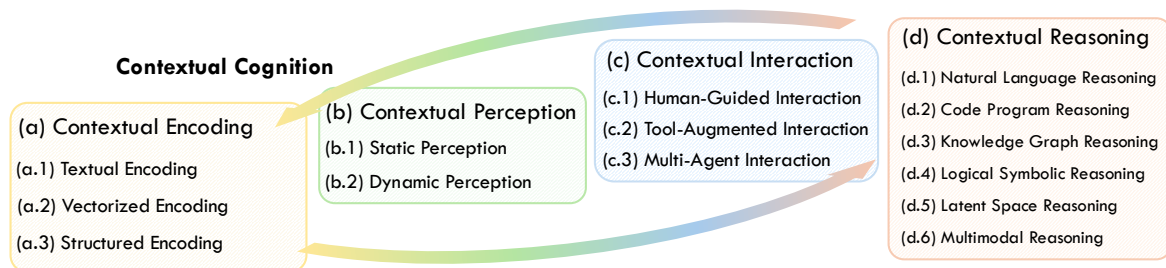


Figure 4. The overall proposed framework. The framework commences with contextual encodings (textual, vectorized, and structured), which tells about the presentation of contextuality. contextual perception extracts semantic content from dynamic environments. contextual interaction then orchestrates human, tool, and multi-agent exchanges. These inputs support diverse contextual reasoning strategies for adaptive decision-making.

4.1. Contextual Encoding

Contextual encoding structures acquired external context for agent reasoning. Moving beyond the short windows of traditional LLMs, it handles dynamic, long-range information to maintain contextual consistency in long-horizon tasks. Therefore, an effective encoding system must carefully balance capacity and efficiency, as shown in Figure 5.

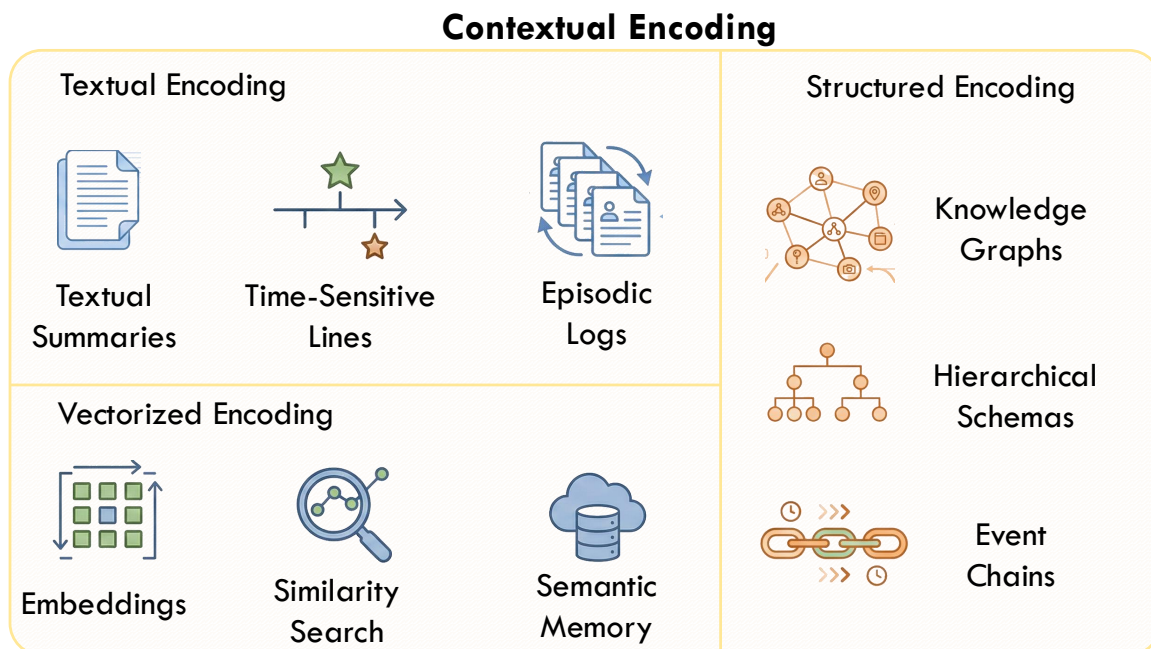


Figure 5. The Contextual Encoding. The architecture encodes information in three formats: Textual for narratives, Structured for relational dependencies, and Vector for semantic retrieval.

4.1.1. Textual Encoding

Textual contextual encoding stores context in readable, interpretable text forms such as summaries, key events, or episodic logs. Systems like MEM1 [38], MemAgent [39], and AWM [42] use multi-level summaries or event records to represent user preferences, task progress, or interaction history. This form aligns well with language-based reasoning and is easy for humans to inspect and edit. However, it tends to grow long and redundant, and its quality depends heavily on the model's extraction and writing abilities, which limits scalability as tasks become more complex.

4.1.2. Vector Encoding

As task demands increase, vector contextual encoding is adopted to improve capacity and retrieval efficiency. This approach encodes context into embeddings and retrieves information via similarity search. Systems such as Memory3 [48] and MemoRAG [47] build large semantic memory pools capable of storing long histories or external knowledge bases. Vector contextual encoding scales well and supports semantic matching in latent space, but it lacks explicit structure and interpretability, making it difficult to encode relations, dependencies, or task logic directly.

4.1.3. Structured Encoding

To address these limitations, recent work introduces structured contextual encoding that uses graphs, hierarchical schemas, event chains, or knowledge trees to explicitly organize context. Examples include HippoRAG [52], Zep [54], A-MEM [55], and G-Memory [56], which capture entities, relations, temporal order, and task dependencies. This form supports more controlled and verifiable reasoning, enables symbolic-style planning, and facilitates multi-agent coordination. Its effectiveness, however, depends on reliable extraction and structure construction, and errors in structure can propagate through the reasoning process.

Despite these advances, contextual encoding still faces significant challenges. Real environments contain heterogeneous sources such as web pages, GUIs, user feedback, sensors, and multimodal signals that are difficult to unify within a single representation. Scalability and continual updating remain problematic as context grows rapidly during long tasks, leading to redundancy, drift, or forgetting. Moreover, aligning structured forms with accurate semantics is nontrivial: errors in extraction can constrain or mislead reasoning. There is also inherent tension between interpretability and control: textual contextual encoding is interpretable but weakly structured, vector contextual encoding is scalable but opaque, and structured contextual encoding is expressive but expensive to build. Current systems lack flexible mechanisms that combine these advantages, and the interaction between contextual encoding and reasoning remains underdeveloped. Progress will require better multimodal integration, more adaptive update strategies, and tighter coupling between encoding and reasoning, enabling agents to form robust context-driven cognitive systems.

4.2. Contextual Perception

As shown in Figure 6, contextual perception processes raw environmental signals into clear features. In complex settings [7], agents face diverse and changing data. To manage this efficiently, we divide perception into two types: observation perception, which captures the current snapshot, and state perception, which tracks changes and feedback over time.

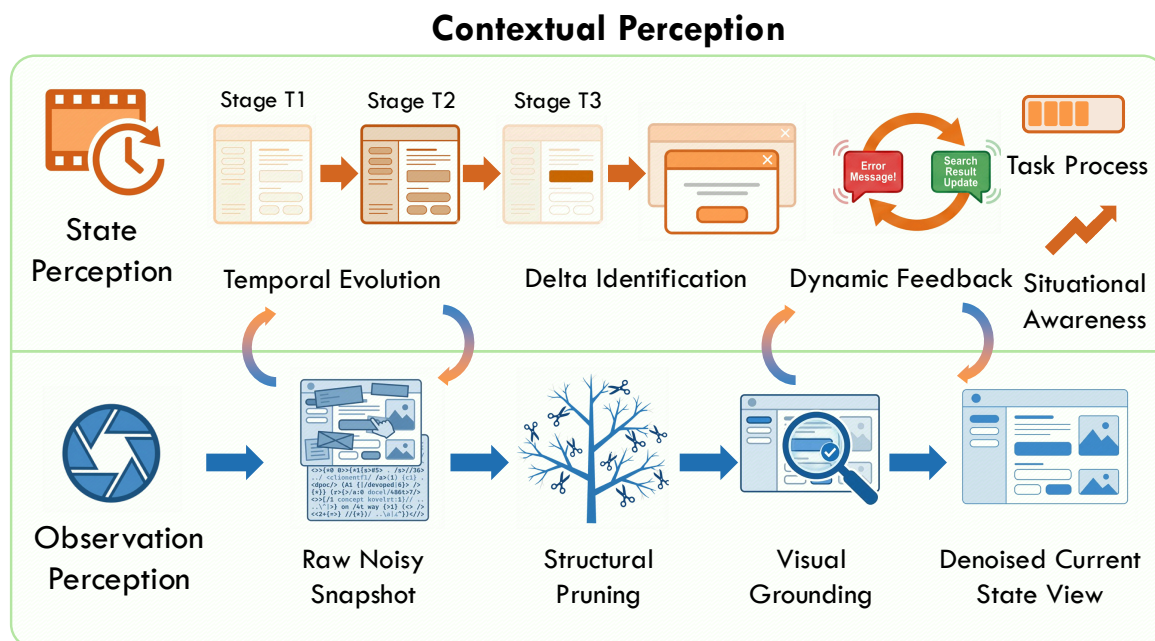


Figure 6. The Contextual Perception. It decomposes sensory processing into observation perception for snapshot filtering and grounding, and state perception for monitoring temporal deltas and feedback loops to enhance awareness.

4.2.1. State Perception

State perception shifts the focus from the snapshot to the stream, monitoring the temporal evolution of the environment. As emphasized by AgentBoard [67], real-world tasks are long-horizon and partially observable; the agent must perceive the "delta" between states to understand the consequences of its actions. This acts as a change detector mechanism. For instance, in e-commerce scenarios within WebShop [65], the agent must actively monitor dynamic feedback loops, such as search result updates or error messages, to verify if the environment has transitioned as intended [175].

This dimension is critical for maintaining situational awareness in evolving contexts. CORE [69] suggests that in scenarios with limited visibility, agents must aggregate fragmentary observations over time to perceive the global status, effectively reconstructing the whole from dynamic parts. Similarly, in social simulations like AgentSociety [68], the environment is a flowing stream of dialogue and social interactions. Here, perception involves monitoring the discourse history to detect shifts in topic or sentiment. Furthermore, research on EnvGen [66] indicates that robust state perception requires adaptability and agents must learn to tune their sensitivity to environmental feedback through repeated interaction. Through state perception, the agent captures what is changing, providing the reasoning core with crucial information about task progress and environmental responsiveness.

4.2.2. Observation Perception

Observation perception addresses the challenge of identifying salient objects and functional affordances within a single, noisy observation frame. In web-based agents, raw inputs ranging from massive HTML trees to high-resolution screenshots contain significant redundancy. Mind2Web [59] demonstrates that efficient perception requires a structural filter to prune irrelevant tags from the raw HTML, extracting only the functional sub-tree necessary for the current task. This filtering extends to the visual domain, CogAgent [62] highlights the necessity of using specialized visual encoders to identify GUI artifacts such as icons and layout boundaries that are not explicitly defined in the code, ensuring the agent perceives the interface similar to a human user.

Beyond mere detection, observation perception involves grounding, defined as the alignment of abstract concepts with concrete environmental elements. AlfWorld [61] formalizes this by requiring agents to map textual object references, such as a target fridge, to specific visual regions in the

scene. Advanced agents like WebVoyager [60] further enhance this by performing semantic validation during the perceptual phase, such as detecting whether a perceived button is disabled or active. By synthesizing these signals, observation perception reconstructs a clean, denoised view of the current snapshot, isolating what exists before any action is taken.

4.3. Contextual Interaction

Contextual interaction integrates perception, reasoning, and action into an iterative decision-making loop. Agents in this loop actively interpret and adapt to the environment rather than passively receiving information. This process requires the agent to interleave immediate state analysis with historical context processing dynamically. As shown in Figure 7, we categorize these interactions into three modes based on interacting entities: Human-Guided Interaction, Tool-Augmented Interaction, and Multi-Agent Interaction.

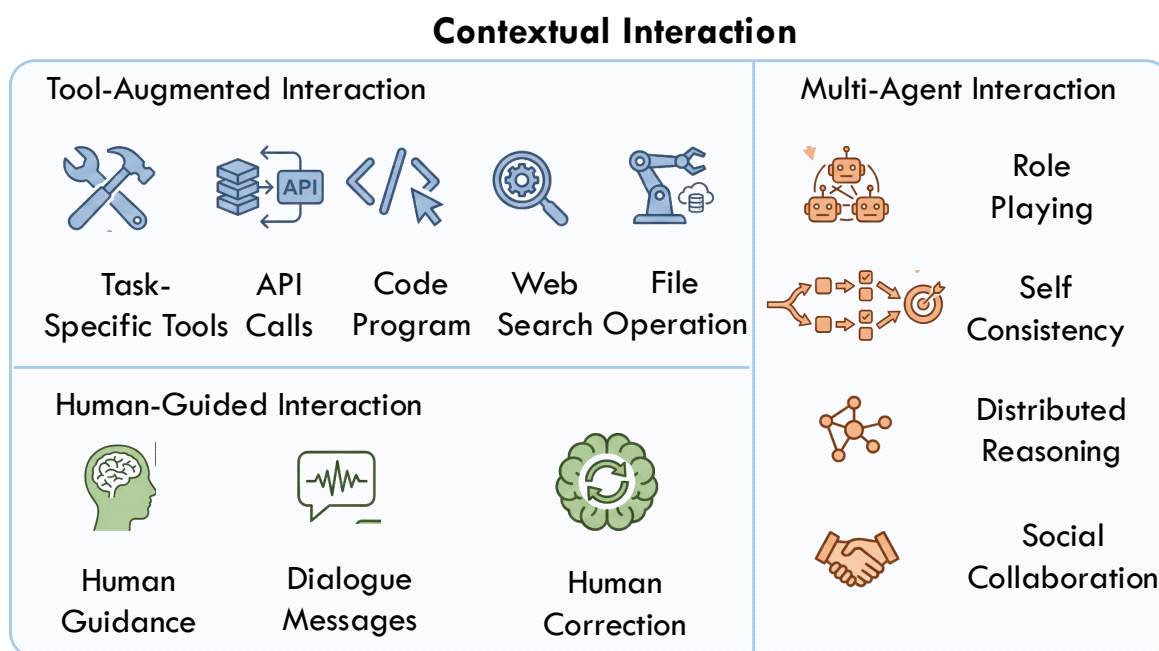


Figure 7. The Contextual Interaction. Anchored on a central Perceive-Reason-Act loop, the process extends into Human-Guided alignment, Tool-Augmented functional expansion, and Multi-Agent collaborative intelligence.

4.3.1. Human-Guided Interaction

Human-guided interaction focuses on alignment and feedback reinforcement through natural language, rule-based and model-based signals. Agents utilize these signals to refine strategies and achieve continuous improvement with or without parameter updates. Reflexion [26] introduces a verbal reinforcement learning framework. The agent generates self-reflective text after each trial and stores this feedback in a memory buffer to induce better decision-making in subsequent attempts. Similarly, Voyager [8] demonstrates lifelong learning in embodied environments. It utilizes an iterative prompting scheme where the agent incorporates environmental feedback and execution errors to refine its code library continuously. Recent advancements such as skills [73] encapsulate complex, domain-specific workflows into modular components, enabling humans to direct AI agents through high-level abstractions rather than low-level execution details. These approaches illustrate that linguistic and environmental feedback serves as a critical driver for agent evolution.

4.3.2. Tool-Augmented Interaction

Tool-augmented interaction extends agent capabilities by treating external tools as effectors for high-precision environmental manipulation. The interaction medium primarily involves executable code, API calls, and structured protocols. Early works like ReAct [74] synergize reasoning and acting

by interleaving thought generation with tool execution. To enhance reliability, Toolformer [75] and Gorilla [76] employ self-supervised learning and retrieval-augmented fine-tuning. These methods ensure the generation of syntactically correct and functionally accurate API calls. Current research shifts from mere usage to autonomous tool creation and unified retrieval. LATM [81] and ToolMaker [77] enable agents to generate reusable tools for new tasks and reduce reliance on human-developed libraries. Simultaneously, DRAFT [78] refines tool documentation through trial-and-error feedback to bridge the understanding gap. ToolGen [79] and Meta-Tool [80] further integrate tool retrieval directly into the generation process or open-world environments. Evaluation benchmarks such as ToolHop [176] highlight the necessity for robust multi-step reasoning in complex tool chains.

4.3.3. Multi-Agent Interaction

Multi-agent interaction facilitates distributed reasoning and collective intelligence through standardized communication protocols. This mode enables task decomposition and role-based collaboration among heterogeneous agents. HuggingGPT [83] utilizes an LLM as a central controller to plan tasks and orchestrate expert models for multi-modal problem solving. To reduce human intervention, CAMEL [84] employs inception prompting to establish autonomous role-playing dialogues between communicative agents. Addressing logic inconsistency in complex workflows, MetaGPT [85] encodes standard operating procedures into agent prompts. This method assigns specific professional roles to agents and enforces a structured assembly line for output verification. AutoGen [86] generalizes this by providing a flexible infrastructure for orchestrating conversational patterns among agents and humans. Beyond task solving, Generative Agents [43] simulate credible human social behaviors through memory retrieval and reflection in a sandbox environment. Recently, sub-agents [82] act as subordinate executors managed by a centralized controller, whereas agent teams embody a decentralized or semi-structured paradigm where multiple agents negotiate and synchronize their actions.

The transition to dynamic contextual interaction empowers agents to handle open-ended problems but introduces significant challenges. First, precise alignment with dynamic intent poses difficulties as user goals often drift during multi-turn interactions. Second, maintaining consistency of interaction state proves critical to prevent self-contradiction over long horizons. Third, the efficiency bottleneck in long context limits the effective utilization of extensive interaction history. Fourth, robustness in tool use requires agents to handle long-tail distribution of tools and ensure syntactic validity. Finally, multi-agent systems face issues regarding coordination costs and the cascading propagation of errors. Future research must address these bottlenecks to build agents that are truly adaptive and collaborative.

4.4. Contextual Reasoning

Contextual reasoning refers to the process by which an agent uses the contextual information it has already obtained and represented to make decisions, solve problems, and complete tasks. After contextual Perception, the agent must reason on top of these internal structures, selecting, transforming, and integrating contextual signals into coherent decision trajectories. As the final stage of the contextual cognition loop, contextual reasoning links structured context to goal-directed actions, determines how the agent interprets evolving evidence, and shapes the stability of long-horizon behaviors.

Prior surveys typically classify reasoning by prompting strategies, architectural designs, or inference procedures. In contrast, as shown in Figure 8, we adopt a complementary view that organizes reasoning by the space in which it operates. From this perspective, contextual reasoning spans six substrates: natural language, code programs, knowledge graphs, logical symbolic systems, latent implicit spaces, and multimodal environments, each offering distinct structures and constraints for manipulating context. This taxonomy shifts attention from the form of reasoning traces to the representational substrate that shapes how reasoning unfolds.

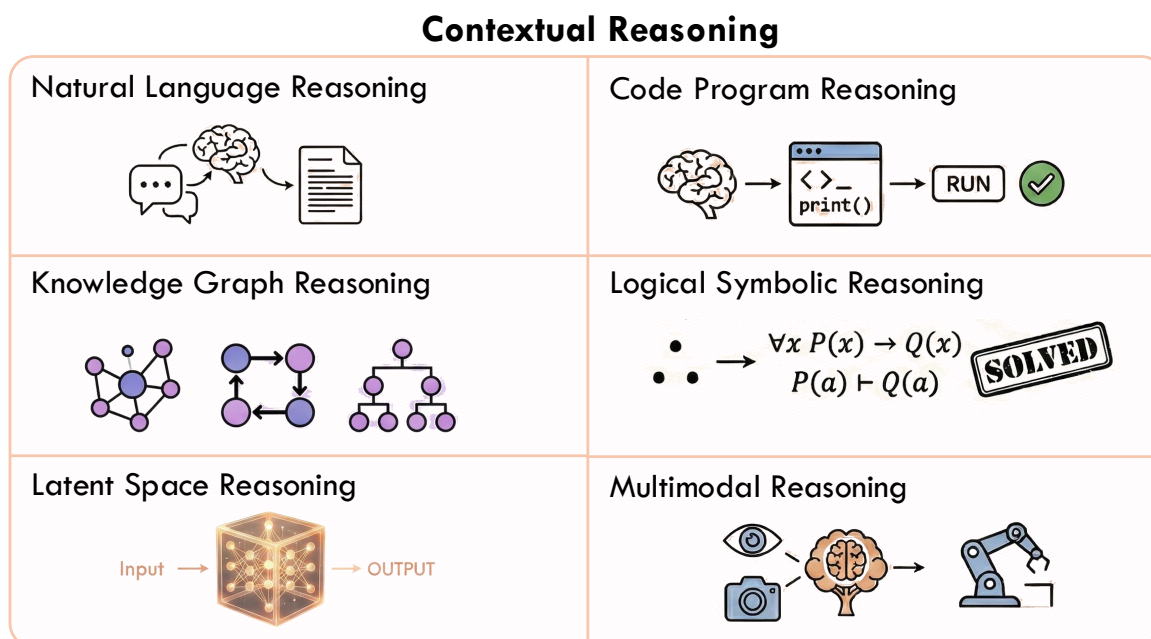


Figure 8. The Contextual Reasoning. The framework divides cognitive processes into six paradigms: language, code program, knowledge graph, logical, multimodal and latent space reasoning.

4.4.1. Natural Language Reasoning

Natural language reasoning situates the entire reasoning process within the linguistic space, where contextual information is transformed through explicit or semi-explicit reasoning traces. Methods such as Chain-of-Thought [35], self-consistency decoding [88], decomposition-based prompting, Tree-of-Thought [89], and ReAct [74] exemplify this paradigm. These approaches allow step-by-step alignment, provide interpretability, and flexibly incorporate retrieved evidence or user feedback. In contextual settings, natural language reasoning operates directly over summaries, episodic logs, and historical observations. However, its reliance on free-form text makes it vulnerable to hallucination, context drift, and inconsistencies during long-horizon tasks [177,178].

4.4.2. Code Program Reasoning

Code program reasoning leverages executable programs as the medium of inference. Instead of producing purely linguistic reasoning traces, the agent translates contextual information into programmatic expressions, such as Python programs, DSL specifications, or tool-centric action code, that are executed and verified by external runtimes. Representative systems include PAL [94], Program-of-Thoughts [95], ViperGPT [99], VISPROG [100], and tool-integrated agents [98]. This paradigm offers deterministic computation, modularity, and verifiability, making it well-suited for tasks requiring grounded interaction with APIs, environments, or external toolchains. Yet, program generation is brittle: minor contextual misinterpretation may yield syntactically or semantically incorrect code, and the reliance on external execution environments introduces additional points of failure [179].

4.4.3. Knowledge Graph Reasoning

Knowledge graph reasoning embeds inference within explicit graph structures consisting of entities, relations, events, or temporal dependencies. Techniques such as Think-on-Graph [101], graph exploration agents [102], and KG-augmented QA [105] transform contextual encodings into graph spaces that support multi-hop reasoning, constraint propagation, and symbolic grounding. Compared to unstructured linguistic reasoning, KG reasoning benefits from relational consistency and structured memory, which are valuable in long-horizon tasks requiring stable world models, domain knowledge integration, or persistent state tracking. However, its performance heavily depends on accurate

graph extraction; errors in mapping contextual information into graph form can severely constrain downstream reasoning.

4.4.4. Logical Symbolic Reasoning

Logical symbolic reasoning situates contextual inference in formal symbolic systems such as first-order logic, constraint satisfaction problems, and SAT/SMT solvers. Methods like Logic-LM [108], SatLM [97], solver-augmented CoT [110], and neuro-symbolic hybrid systems convert contextual inputs into logical formulas, constraints, or proof steps that can be processed by symbolic solvers with correctness guarantees. This paradigm is particularly useful in tasks requiring strict consistency, rule adherence, or formal verification. Within the contextual cognition loop, symbolic reasoning can validate or refine contextual encodings by enforcing invariants. However, the translation from natural language to logical form remains brittle, and small semantic errors may lead to unsatisfiable or incorrect symbolic encodings [180].

4.4.5. Latent Space Reasoning

Latent space reasoning refers to implicit reasoning processes occurring entirely within the model's internal states, without explicit intermediate steps. Recent large reasoning models, implicit Chain-of-Thought frameworks, reinforcement-learned reasoning policies, and latent-memory systems exemplify this paradigm [72,111,112]. These models perform multi-step inference through internal computations such as attention routing and hidden-state transitions. Latent reasoning is efficient and avoids the verbosity and instability of explicit text-based reasoning. However, it sacrifices interpretability and controllability: contextual misalignment or reasoning errors become harder to diagnose, and ensuring safety or consistency in long-horizon latent reasoning remains an open challenge.

4.4.6. Multimodal Reasoning

Multimodal reasoning extends contextual inference to environments involving images, videos, GUI states, visual affordances, or embodied sensory inputs. Frameworks such as Multimodal Chain-of-Thought [115], MM-ReAct [116], VISPROG [100], and ViperGPT [99] integrate textual reasoning with perceptual grounding to bridge the gap between language and sensory data. In this paradigm, contextual information may originate from complex sensory observations, requiring spatial, semantic, and temporal integration. Multimodal reasoning enables situated decision-making, visual planning, and perception-informed action. Yet, it inherits uncertainty from perception pipelines: errors in visual grounding can cascade through the reasoning process, and aligning multimodal signals with textual context remains a significant challenge.

Despite rapid advances across these reasoning spaces, achieving stable, verifiable, and context-grounded reasoning in open environments remains difficult. A central problem is context grounding: reasoning steps must remain faithful to earlier contextual encodings, yet models frequently hallucinate premises, ignore context, or drift from task goals. Dynamic environments require continuous updating of reasoning states, while long-horizon tasks exacerbate inconsistencies across steps. Explicit reasoning offers transparency but is brittle; implicit reasoning offers efficiency but lacks interpretability; symbolic reasoning provides correctness but relies on fragile semantic parsing; multimodal reasoning compounds noise from perception; and graph- or program-based reasoning depends on accurate structured extraction. Addressing these challenges requires advances in context verification, reasoning control, cross-space consistency, and adaptive reasoning capable of operating reliably in real-world settings.

5. Building Contextual LLM Agents

In this section, we explore two primary construction mechanisms for contextual agents: runtime runtime and foundation LLM training, including their convergence in Figure 9. We contrast how foundation LLM training optimizes end-to-end policies, while runtime runtime structures cognitive processes. Together, these pathways demonstrate how the contextual lifecycle is managed and refined.

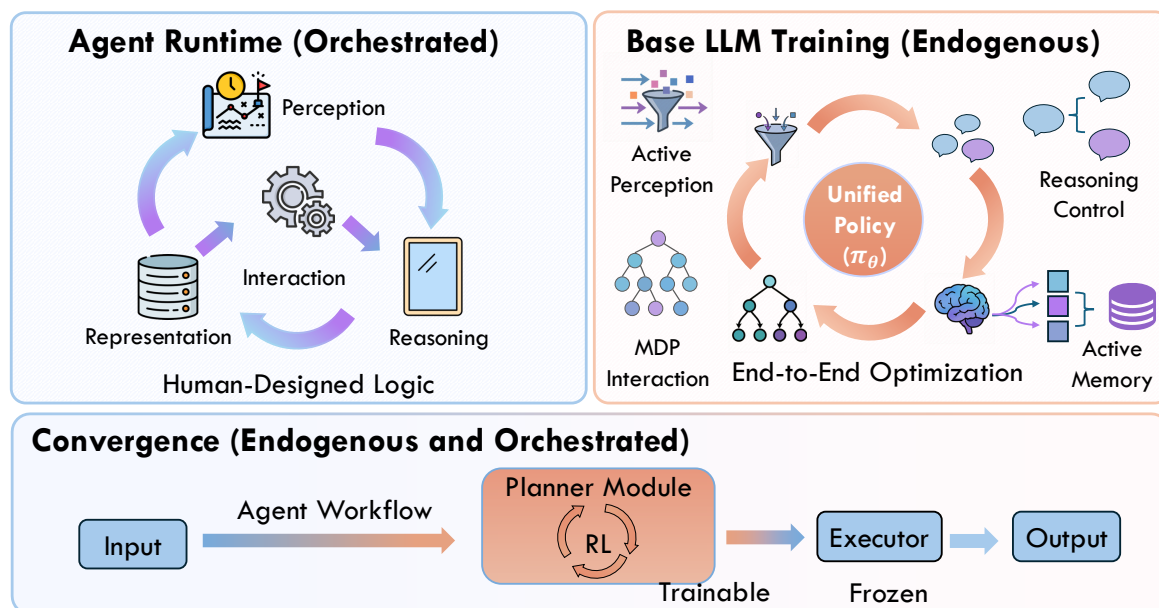


Figure 9. Architectural paradigms in Contextual Cognition. We contrast end-to-end Agentic RL with modular Workflow Orchestration, leading to a convergent hybrid that injects learnable policies into structured workflows.

5.1. Agent Runtime

Unlike learning-based paradigms that implicitly optimize policies via end-to-end training [61,181], workflow-based agent runtime focuses on the explicit orchestration of cognitive architectures [17,27]. In this paradigm, the agent is governed by structured execution flows, which essentially serve as the control logic for contextual flow [85,86]. This modular design allows developers to decompose the complex lifecycle of contextual cognition into deterministic, interpretable steps. To systematically align with the agent's cognitive lifecycle established in this survey, we categorize these orchestrated modules into four consecutive phases: contextual perception, interaction, representation, and reasoning [182].

5.1.1. Agent Runtime for Contextual Encoding

Contextual encoding governs how the agent structurally encodes both anticipated future states and historical experiences to guide current behavior. To explicitly decouple global planning from local execution, dedicated planners generate high-level, abstract guidance, termed "meta plans," which outline the trajectory of context evolution independent of specific environmental details [183,184]. This explicit foresight is critical for long-horizon tasks, preventing myopic behavior. Recently, to-do lists have emerged as dynamic closed-loop encodings that track the real-time progression of tasks from initiation to completion, ensuring planned workflows are verifiable and fully realized [185]. To bridge transient processing and lifelong learning [43], the workflow relies on hierarchical long-term memory (episodic, semantic, procedural) powered by vector databases. This encoding enables retrieval-augmented generation [186] to dynamically fetch relevant past experiences. Frameworks like Reflexion further convert transient feedback into persistent verbal reinforcement [26], allowing the agent to weave historical lessons into current contexts for continuous self-improvement and consistency across long-term interactions [8,49,187].

5.1.2. Agent Runtime for Contextual Perception

Contextual perception establishes the agent's active awareness of its current environment, available capabilities, and the inherent gap between its current state and global objectives [188]. Rather than reacting blindly, this workflow optimizes the agent's active workspace via short-term working memory, employing information density optimization to retain critical task states within the finite context window. This ensures the agent is always grounded in the most relevant aspects of its environment. Beyond environmental states, perception critically extends to tool awareness. To

manage the complexity of heterogeneous tool ecosystems, the Model Context Protocol (MCP) serves as a standardized architecture [189]. MCP abstracts diverse data sources and tools into standardized primitives (resources, prompts, and tools), ensuring the agent remains universally aware of how to discover and connect to local or remote utilities without bespoke glue code [98]. This standardized perception mechanism is the prerequisite for any meaningful action.

5.1.3. Agent Runtime for Contextual Interaction

Contextual interaction functions as the interface where internal cognition is grounded in the external environment, enabling the agent to actively deploy its capabilities to modify the context [75]. Fundamental execution operations are realized through explicit function calling mechanisms [76]. This modularity allows developers to seamlessly transfer complex "action awareness" and tool-use trajectories, initially established by large models, down to smaller, more specialized models [190]. Furthermore, interaction encompasses a dynamic generate-evaluate-refine cycle to engage with environmental feedback [191]. When an action is executed, the workflow synthesizes multi-source feedback to reconcile discrepancies between the intended context (the plan) and the actual outcome [192]. Instead of immediately halting upon error, the agent actively interacts with this feedback to diagnose root causes and syntax errors from logic flaws. This process is supported by defensive layers that ensure these active contextual updates do not compromise overall integrity.

5.1.4. Agent Runtime for Contextual Reasoning

Contextual reasoning serves as the central computational engine, determining how the agent dynamically manipulates and processes represented information to derive conclusions. This workflow orchestrates reasoning across two orthogonal dimensions: depth and breadth. To enhance reasoning depth, agents employ test-time scaling strategies like Chain-of-Thought [35], engaging in a step-by-step dialogue with the current context to break down complex queries into solvable sub-problems via decomposition [193]. Additionally, structured thought paradigms like Tree of Thoughts [89] are utilized to simulate potential future contexts and reasoning paths. To enhance reasoning breadth, the workflow incorporates uncertainty quantification as a mechanism for contextual verification [194]. By utilizing techniques like self-consistency [88], the agent samples multiple diverse reasoning paths to assess the stability of its conclusions. This breadth-oriented mechanism acts as a cognitive brake: when uncertainty is high, it triggers defensive behaviors, such as asking for clarification, ensuring the reasoning process remains robust. Finally, the root causes diagnosed during the interaction phase are fed back into this reasoning engine as refined prompts, forcing the model to systematically regenerate corrected solutions.

5.2. Foundation LLM Training

While traditional reinforcement learning (e.g., RLHF) treats LLMs as passive, single-turn generators relying on static, user-provided contexts, agentic reinforcement learning shifts this focus by embedding the model as an interactive policy within a partially observable environment (POMDP). In this dynamic setting, context is not passively received but actively constructed: the agent executes actions and processes feedback to shape its own input distribution. Unlike workflow-based methods (§5.1) that depend on human-engineered pipelines, Agentic RL pursues the end-to-end learning of contextual cognition, enabling the agent to autonomously optimize its entire cognitive lifecycle based on task rewards.

5.2.1. RL for Contextual Encoding

Contextual encoding focuses on how an agent structures acquired information into a usable internal state for downstream reasoning. Rather than treating the context window as a passive buffer, agentic RL frames memory management as an active decision process involving retrieval, maintenance, and compression [195]. While systems like Mem0 [132] demonstrate the value of structured memory architectures, recent RL approaches make the memory operations themselves learnable. For example, Memory-R1 optimizes a memory manager that decides distinct actions (ADD, UPDATE, DELETE) via

end-to-end reinforcement to autonomously curate a concise knowledge bank. Furthermore, addressing long-horizon challenges, frameworks such as AgentFold [196] and Context-Folding [138] treat the context history as a flexible workspace, training the agent to proactively fold or compress past trajectory segments. By doing so, these methods transform encoding from a static storage problem into a policy-optimized resource management task.

5.2.2. RL for Contextual Perception

Contextual perception concerns how an agent actively filters vast, noisy environmental signals into a concise internal observation tailored for reasoning. Unlike passive data pipelines, RL-driven perception optimizes an active filtering policy to maximize the signal-to-noise ratio before reasoning begins. We structure this design space along three core strategies: (i) active sensing to bring relevant information into view, as seen in GUI agents learning to scroll or navigate apps to locate task-critical widgets [21,197,198]; (ii) attentional filtering to distill specific regions and suppress noise, demonstrated by visual agents using tools like cropping to isolate informative areas [130,199]; and (iii) perception budgeting to weigh the marginal gain of acquiring new information against the inherent interaction costs [179,200]. By mastering these behaviors, the agent transforms perception from a fixed front-end into a dynamic, context-shaping process.

5.2.3. RL for Contextual Interaction

Contextual interaction concerns how an agent sequences tool calls and environment actions to acquire useful context over time. Instead of following rigid, pre-defined pipelines (e.g., search-then-answer), agentic RL formulates information seeking as a dynamic interaction policy, deciding when to issue a query, how to refine search terms based on partial feedback, and when to terminate exploration. This transforms interaction into an optimizable trade-off between information gain and interaction steps. In web research, models like Search-R1 [133], R1-Searcher [201], and Tongyi DeepResearch [202] utilize outcome-based RL to learn iterative browsing strategies that autonomously determine search depth and specificity. Similarly, in software engineering, DeepSWE and SWE-RL [136] treat coding as a multi-step debugging loop. By using compiler feedback and test execution as rewards, these agents learn trial-and-repair policies, acquiring just enough high-quality context to solve tasks reliably without unnecessary exploration.

5.2.4. RL for Contextual Reasoning

Contextual reasoning concerns the optimization of the structure and control of the thought process itself. Beyond simply generating an answer, agentic RL aims to learn the optimal topology of reasoning, determining how many steps to take, when to branch, and how to interleave internal thought with external tool use. This shift from imitating fixed chain-of-thought templates to discovering robust reasoning patterns, such as strategic backtracking, maximizes correctness in complex environments [203]. By treating intermediate reasoning steps and tool invocations as actions, methods like SWiRL [141], ARTIST [142], and ToRL [139] reinforce locally useful sub-goals and coherent tool usage. Furthermore, RL enables meta-cognitive control over the planning process; frameworks like RLTR [140] and Learning-When-to-Plan [204] reward agents for deciding when to engage in explicit planning versus acting directly, thereby dynamically balancing computational expenditure with reasoning depth [205].

5.3. Convergence: RL-Optimized Agentic Workflows

While Agentic RL (§5.2) typically trains monolithic policies and Workflow Orchestration (§5.1) relies on static, human-designed logic, a rapidly emerging paradigm seeks to converge these approaches by injecting learnability into modular workflows. This hybrid direction retains the structural interpretability of engineered pipelines while leveraging reinforcement learning to optimize critical decision-making nodes, such as planners or routers, within the system.

Recent frameworks demonstrate two distinct pathways for this integration. AgentFlow [146] introduces module optimization in the flow, where a Planner is explicitly trained using trajectory-level

feedback. By broadcasting outcomes to intermediate turns, it solves credit assignment in long-horizon tasks while keeping tool-use and verification modules frozen. Conversely, Agent-Lightning [206] proposes a framework-agnostic approach, decoupling reinforcement learning training from agent execution. By formulating any agentic chain as a Markov Decision Process through a unified data interface, it enables the selective optimization of specific components without modifying the underlying workflow code.

The core of this approach is integrating modular steps—like tool-calling and memory retrieval—into a unified reinforcement learning loop. By treating workflows as stochastic policies rather than fixed sequences, agents can adaptively explore reasoning paths to maximize long-term rewards. A prime example is Claw-R1 [207], an advanced framework that empowers general agents (like OpenClaw) through Agentic RL. It utilizes a middleware architecture to decouple agent execution from training, allowing models to learn from environmental feedback and optimize high-variance decision points without retraining the entire foundation. This targeted optimization ensures sample efficiency and maintains safety.

In summary, the transition from prompt engineering to flow engineering with reinforcement learning signifies the rise of trainable cognitive systems. In these systems, structural priors ensure safety and reliability while reinforcement learning continuously refines reasoning policies within those constraints. This white-box optimization combines symbolic structure with neural adaptability, paving the way for autonomous agents that self-evolve through environmental interaction while remaining fundamentally interpretable to human designers.

6. Benchmarks and Applications

This section surveys the evaluation landscape and practical frontiers of contextual agents. We first categorize benchmarks across encoding, interaction and reasoning dimensions to assess core competencies. Subsequently, we examine applications in science, deep research, and finance, illustrating the transition toward autonomous, full-cycle problem solving.

6.1. Datasets and Benchmarks

To examine the core abilities of contextual agents, existing benchmarks are organized into three dimensions, contextual encoding, interaction, and reasoning, each capturing a layer of an agent’s ability to engage with, internalize, and reason over context. Table 4 summarizes representative benchmarks, illustrating how current frameworks probe different facets of contextual intelligence.

Table 4. Comprehensive Benchmark Landscape for Contextual Agent Evaluation.

Category	Benchmark	Task Type	Key Focus
Contextual Encoding	BBEH [154]	Hard reasoning set	Stress-testing world-model consistency.
	LongMemEval [152]	Long-term interactive dialogue	Long-term memory stability and drift resistance.
	MemoryAgentBench [159] StreamBench [155]	Incremental multi-turn tasks Streaming input-feedback tasks	Accurate recall and cross-turn integration. Continuous improvement via streaming feedback.
Contextual Interaction	GAIA [147]	Real-world multi-step tasks	Open-world tasks with intention following and tool use.
	HLE [148]	Human-level exams	Broad situational cognition in human-level tasks.
	Uni-RLHF [149]	Feedback-driven adaptation	Adapting behavior based on user feedback.
	ToolLLM [98] StableToolBench [150] AgentBench [153]	API/function calling Virtual API environment Multi-environment agent tasks	Reliable API calling and tool invocation. Stable, reproducible large-scale tool use. Acting across diverse interactive environments.
Contextual Reasoning	REALM-Bench [158] FlowBench [156] Reflection-Bench [151] LR ² Bench [157]	Real-world planning Workflow-guided planning Cognitive psychology tasks Consistency-based reasoning	Dynamic planning and replanning. Workflow-guided multi-step reasoning. Epistemic reasoning and belief updates. Long-chain reflective reasoning.

Contextual encoding concerns an agent’s ability to build, maintain, and update internal models of its environment, memories, and evolving tasks, central to situational intelligence. BBEH [154] exposes the limits of an agent’s world-model consistency in cross-domain reasoning tasks, revealing how fragile internal encodings become under high cognitive load. Long-range memory formation is examined by LongMemEval [152], which stresses stability, selective forgetting, cross-turn coherence, and resilience to context drift during extended interactions. MemoryAgentBench [159] further decomposes the encoding problem in incremental multi-turn tasks, testing accurate recall, information accumulation, and integration across unfolding contexts. Extending beyond static memory, StreamBench [155] evaluates whether agents can continuously improve by updating prompts, memory, or retrieval mechanisms over streaming sequences, measuring an agent’s ability to construct encodings that evolve with experience rather than being fixed at inference time. These benchmarks capture how contextual encodings support situationally aware behavior, showing that while agents can form task-specific internal states, persistent and updateable encodings remain a major bottleneck in open-ended environments.

Evaluating contextual interaction focuses on how effectively an agent engages with humans, tools, and environments, reflecting its situational awareness and ability to act in real-world contexts. Benchmarks such as GAIA [147] and HLE [148] assess an agent’s competence in multi-step tasks involving intention understanding, tool-mediated actions, and open-world knowledge integration, highlighting whether an agent can follow human goals across long-horizon procedures. Human-aligned improvement is further examined by Uni-RLHF [149], which evaluates how agents update contextual understanding based on user feedback, a core requirement for adaptive human–AI collaboration. Tool-based interaction is captured by ToolLLM [98] and StableToolBench [150], which analyze an agent’s reliability in API calling, error recovery, and large-scale tool usage under realistic constraints. Meanwhile, AgentBench [153] extends evaluation into multi-environment settings, from operating systems and databases to web and embodied games, probing whether agents can ground their actions in dynamic environments with strict interfaces and partial observability. Together, these benchmarks portray contextual interaction as the foundation of situational intelligence, revealing both the strengths and fragilities of agents.

Contextual reasoning measures how effectively an agent leverages situational information to perform structured decision-making, long-horizon inference, and adaptive strategy formation. REALM-Bench [158] evaluates agents in planning and scheduling tasks that incorporate disturbances, multi-agent coordination, and logistical constraints, revealing their ability to re-plan under changing environments. FlowBench [156] examines workflow-guided planning by comparing text-, code-, and diagram-based representations, assessing how different forms of contextual knowledge shape multi-step reasoning and conversational planning. Reflection-Bench [151] evaluates epistemic agency, such as belief revision, counterfactual reasoning, prediction, and meta-reflection, probing whether agents can form reflective chains that regulate their own reasoning. LR2Bench [157] targets long-chain reflective reasoning, measuring multi-step thinking and the capacity to self-correct across extended trajectories. These benchmarks characterize contextual reasoning as the apex of situational intelligence: the ability not just to act or remember, but to deliberate, adapt strategies, revise beliefs, and construct long-horizon plans grounded in evolving context [208,209].

6.2. LLM Agent Applications

Agentic AI is transitioning toward autonomous, full-cycle problem solving, exemplified by systems like AI Co-Scientist [210] and Manus [211]. Figure 10 highlights this shift across deep research, software engineering, graphical user interfaces, and scientific discovery, where agents now integrate retrieval, planning, and execution into end-to-end workflows. This evolution marks the rise of contextual agents as collaborators capable of independent reasoning and action within complex environments [172].

Deep research agents aim to enable models to perform multi-hop retrieval, information integration, and in-depth reasoning across open web environments and massive document repositories. Industrial systems like Google’s NotebookLM accelerate this by grounding dynamic interactions in user-provided

contexts, while academic frameworks such as OpenResearcher [166], ResearchAgent [167], and DeepResearcher [20] equip models with robust behaviors for literature processing, cross-validation, and reflection. Crucially, these capabilities extend beyond academic inquiry into complex commercial analysis, where systems like FinAgent [168] and FinCon [169] integrate multimodal perception, hierarchical memory, and multi-tool orchestration to maintain consistent decision-making. Overall, deep research agents are rapidly evolving from retrieval-augmented assistants into comprehensive analysts capable of autonomously synthesizing external knowledge across diverse real-world landscapes.

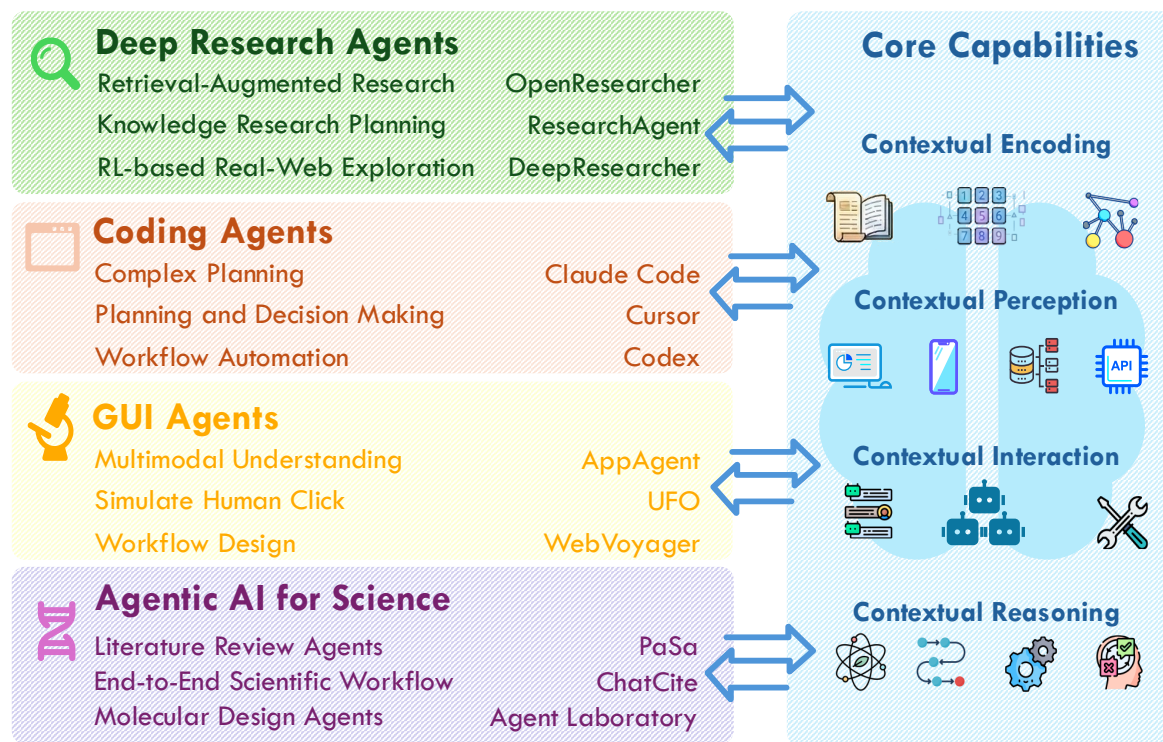


Figure 10. Overview of how LLM agents leverage core capabilities like perception and interaction to automate complex workflows across deep research, coding, GUI and scientific agents.

Coding agents represent a significant leap in converting natural language intent into functional software through iterative interaction with development environments. MetaGPT [85] models a complete software company by assigning specialized roles to multiple agents to streamline collaborative development, whereas SWE-agent [212] focuses on single-agent efficiency via an Agent-Computer Interface tailored for code editing and execution. Furthermore, autonomous frameworks like OpenHands [170] demonstrate the capability to handle end-to-end software engineering tasks, from environment setup to testing and debugging. By maintaining a continuous contextual understanding of the codebase and execution feedback, these agents significantly reduce human intervention in the development lifecycle.

GUI agents interact with digital ecosystems by perceiving visual interfaces and executing human-like actions, bridging the gap between natural language and operating systems. AppAgent [213] enables language models to operate smartphone applications through a simplified action space, learning UI navigation via exploration and human demonstration. In desktop environments, UFO [214] acts as a dual-agent framework for Windows OS, seamlessly operating across multiple applications to fulfill user requests, while WebVoyager [60] processes web layouts and visual elements to execute complex transactional workflows. These agents rely on sophisticated visual grounding to translate dynamic screen states into actionable external observations, expanding the reach of LLMs beyond text-based interfaces.

In scientific-agent systems, recent work shows a clear evolution from foundation models to multi-agent architectures capable of autonomous discovery. PaSa [161] and ChatCite [162] enhance literature

discovery and reflective incremental writing, while Agent Laboratory [163] integrates data preparation, experimental execution, and manuscript writing into a reproducible end-to-end workflow. Beyond literature, NatureLM [160] constructs a unified sequence-based foundation model spanning diverse scientific domains, providing a backbone for systems like ProtAgents [164] and Bio AI Agent [165], which coordinate multi-agent reasoning across physical simulation, target identification, and molecular design. Overall, these systems reflect a growing trend in which contextual agents augment the full cycle of “search–understand–design–validate–decide,” driving scientific discovery toward increasingly autonomous collaboration.

6.3. Case Study: Contextual Cognition for OpenClaw

Personal AI assistants, such as OpenClaw, serve as a prototypical application for contextual cognition, where multi-turn communication requires dynamic state management across diverse messaging platforms. In this setting, the agent’s external observation space, comprising persistent memory and accessible skills, must be continuously internalized. Through contextual encoding, OpenClaw translates multimodal inputs from various message channels and user preferences into a structured internal representation. This encoded state forms the foundation for contextual perception, allowing the agent to actively monitor its environment, parse task contexts, and evaluate available plugins. Rather than treating context as a static prompt, these initial cognitive stages ensure the evolving personal state remains retrievable and actionable across multiple interaction sessions.

As shown in Figure 11, the operational lifecycle of OpenClaw is driven by a continuous cognitive loop. Moving beyond passive perception, the agent engages in contextual interaction by mapping its perceived environment to deliberate internal states, like intent recognition and action planning. This interactive phase utilizes execution logs and API feedback to orchestrate workflows dynamically across platforms. Finally, the system employs contextual reasoning to synthesize these interactions into robust decision-making, enabling critical functions such as state updating and error recovery. By integrating encoding, perception, interaction, and reasoning, OpenClaw transforms a fragmented cross-platform environment into a cohesive cognitive architecture, elevating the agent from a response generator to a resilient system capable of multi-step task completion.

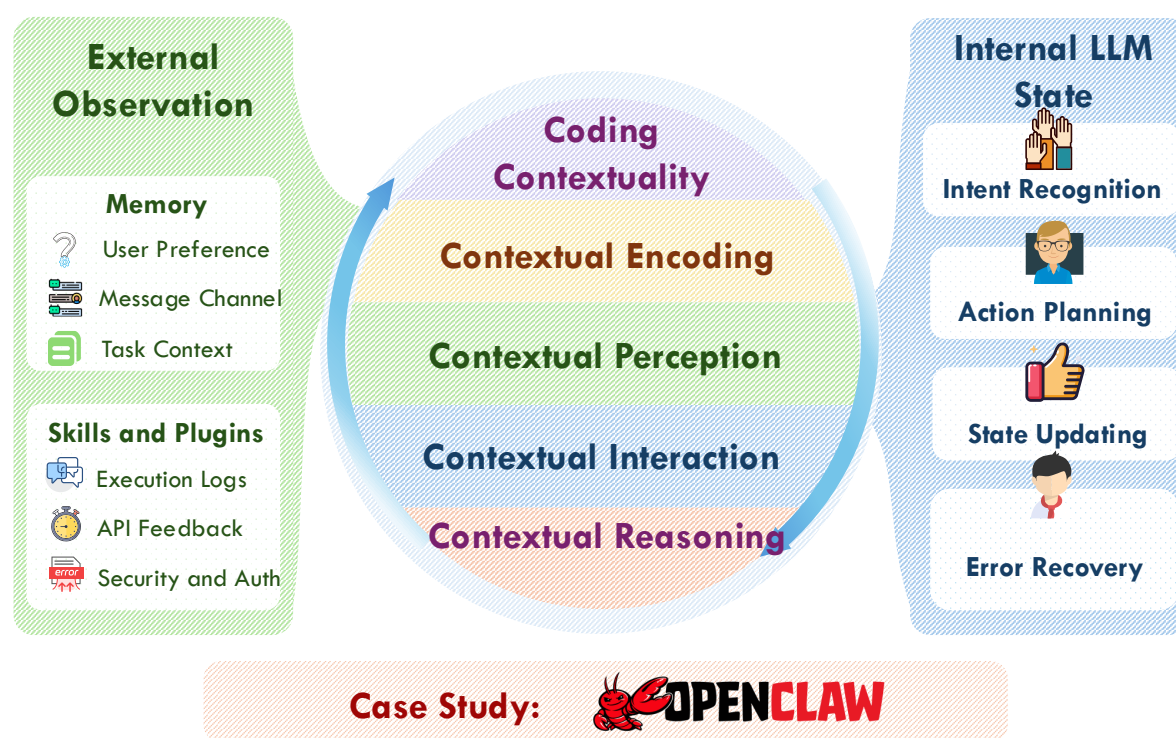


Figure 11. Case Study: coding agents adaptively build context from codebase states and runtime feedback to action like requirement inquiry, code generation, refactoring, and self-debugging for automated software development.

7. Future Research Directions

The transition from static module optimization to dynamic contextual cognition presents significant opportunities. We identify six critical directions to enhance the adaptability and reliability of LLM agents: evolving always-on runtimes, natively aligning models for continuous interaction, advancing dynamic state management. Finally, broadening applications into expert domains and pioneering dynamic evaluation frameworks will be essential.

7.1. Evolving Agentic Runtimes

The emergence of tools like Claude Code highlights a shift toward persistent, agent-centric execution environments, yet current systems remain transitional. Future agentic runtimes must evolve beyond static sandboxes into dynamic, lifelong operating systems that inherently support contextual cognition. This requires infrastructures capable of continuous, asynchronous state synchronization, where the agent seamlessly aligns its evolving internal state with real-time external observations. Developing these human-like, always-on runtimes will enable agents to proactively monitor environments, manage long-term background processes, and maintain situational awareness across extended interaction lifecycles.

7.2. Aligning LLM for Contextual Cognition.

Currently, most large language models are optimized for single-turn generation rather than continuous environmental interaction. To truly realize contextual cognition, the underlying foundation models must be natively aligned with agentic behaviors. This involves shifting the training focus from static text completion to optimizing interaction policies within partially observable environments. Open-source reinforcement learning frameworks tailored for agentic AI, provide a critical pathway for this evolution. By integrating reward signals derived directly from environmental feedback, these frameworks enable models to autonomously construct and refine their working context, bridging the gap between passive reasoning and grounded action.

7.3. Advanced Contextual State Management

As interactions become increasingly complex, maintaining a coherent internal state over long horizons presents a significant challenge. Future research must address how agents autonomously manage their memory architectures—transitioning from passive vector stores to active, hierarchical state management. This involves learning optimal policies for encoding, compressing, and retrieving episodic and procedural experiences. Effective state management will allow agents to dynamically fold historical contextuality into real-time decision-making, ensuring continuous self-improvement while mitigating the risks of context overload and information degradation.

7.4. Expanding Contextual Encodings

For agents to interact robustly with the physical and digital world, the external observation layer must become highly digitized and semanticized. Future development requires the creation of richer toolsets capable of translating unstructured, multimodal signals into formal, closed-loop representations. By expanding the agent's perceptual bandwidth—through advanced APIs, sensory inputs, and structural filters—ambiguous environmental feedback can be converted into precise, expressible operational states. This strict digitization ensures that the dynamic contextuality remains verifiable and actionable throughout the reasoning cycle.

7.5. Broadening Agent Applications

As contextual capabilities mature, agentic applications will rapidly expand beyond constrained coding or web-navigation tasks into highly open-ended, expert-level domains. Future systems will drive complex, long-horizon workflows, such as synthesizing comprehensive commercial analyses, autonomously tracing and forecasting scientific evolution by constructing structured trend trees, and orchestrating massive multi-agent ecosystems. In these advanced applications, contextuality will not

just be maintained by a single entity, but shared, negotiated, and dynamically updated among diverse agents to solve cross-disciplinary challenges.

7.6. Evaluating Dynamic Contextual Interactions.

The dynamic and adaptive nature of contextual cognition necessitates a fundamental shift in how agentic systems are evaluated. Traditional static benchmarks, which focus on final output accuracy, are insufficient for assessing the quality of multi-step, closed-loop interactions. Future evaluation frameworks must measure the efficiency of contextual perception, the robustness of the interaction trajectory, and the agent's ability to recover from environmental shifts or tool failures. Developing dynamic, tool-augmented testing environments that quantify the trade-offs between information gain and interaction costs will be critical for accurately measuring true agentic intelligence.

8. Conclusion

In this survey, we challenge the conventional modular view of LLM-based agents by establishing a unified foundation centered on contextual cognition. We argue that the emergence of agentic intelligence stems not from the mere organization of diverse modules, but from how an agent manages and interacts with contextuality: the dynamic integration of external observations and internal states. To operationalize this insight, we propose a comprehensive framework comprising contextual encoding, perception, interaction, and reasoning. This framework serves as the lens to systematically review current agent construction methods, encompassing runtime orchestration and foundation LLM training. By revisiting corresponding benchmarks and complex applications, we highlight the necessity of shifting focus from static modules to dynamic, contextualized processes. We conclude that overcoming the open challenges rooted in this cognitive bottleneck is essential for future research, providing a roadmap to drive the development of robust, real-world agentic systems.

References

1. Anthropic. Claude Code: Build, debug, and ship from your terminal. <https://claude.ai/product/claude-code>, 2025. Accessed: 2026-02-06.
2. openclaw. openclaw: Your own personal AI assistant. Any OS. Any Platform. The lobster way. <https://github.com/openclaw/openclaw>, 2024.
3. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* 2020.
4. Hu, Y.; Liu, S.; Yue, Y.; Zhang, G.; Liu, B.; Zhu, F.; Lin, J.; Guo, H.; Dou, S.; Xi, Z.; et al. Memory in the Age of AI Agents. *arXiv preprint arXiv:2512.13564* 2025.
5. Wang, X.; Chen, Y.; Yuan, L.; Zhang, Y.; Li, Y.; Peng, H.; Ji, H. Executable code actions elicit better llm agents. In Proceedings of the Forty-first International Conference on Machine Learning, 2024.
6. Ren, J.; et al. Towards Scientific Intelligence: A Survey of LLM-based Scientific Agents. *arXiv preprint arXiv:2503.24047* 2025.
7. Zhou, S.; Xu, F.F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854* 2023.
8. Wang, G.; Xie, Y.; Jiang, Y.; Mandlkar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* 2023.
9. Wang, X.; Cui, Z.; Li, H.; Zeng, Y.; Wang, C.; Song, R.; Chen, Y.; Shao, K.; Zhang, Q.; Liu, J.; et al. PerPilot: Personalizing VLM-based Mobile Agents via Memory and Exploration. *arXiv preprint arXiv:2508.18040* 2025.
10. Huo, Y.; Lu, Y.; Zhang, Z.; Chen, H.; Lin, Y. AtomMem: Learnable Dynamic Agentic Memory with Atomic Memory Operation. *arXiv preprint arXiv:2601.08323* 2026.
11. Ding, D.; Liu, S.; Yang, E.; Lin, J.; Chen, Z.; Dou, S.; Guo, H.; Cheng, W.; Zhao, P.; Xiao, C.; et al. OctoBench: Benchmarking Scaffold-Aware Instruction Following in Repository-Grounded Agentic Coding. *arXiv preprint arXiv:2601.10343* 2026.
12. 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.
13. Zhang, G.; et al. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey. *arXiv preprint arXiv:2509.02547* 2025. Key survey for Agentic RL.

14. Gao, H.a.; et al. A Survey of Self-Evolving Agents: On Path to Artificial Super Intelligence. *arXiv preprint arXiv:2507.21046* **2025**. Key survey for Self-evolution.
15. Tran, K.T.; Nguyen, H.D.; et al. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv preprint arXiv:2501.06322* **2025**.
16. Hassan, A.; Graham, B. Evaluating LLM-based Agents for Multi-Turn Conversations: A Survey. *arXiv preprint arXiv:2503.22458* **2025**.
17. Xi, Z.; Chen, W.; Guo, X.; et al. The Rise and Potential of Large Language Model Based Agents: A Survey. *Frontiers of Computer Science* **2024**. Comprehensive architecture review.
18. Xie, T.; Zhang, D.; Chen, J.; Li, X.; Zhao, S.; Cao, R.; Hua, T.J.; Cheng, Z.; Shin, D.; Lei, F.; et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems* **2024**, *37*, 52040–52094.
19. Mon-Williams, R.; Li, G.; Long, R.; Du, W.; Lucas, C.G. Embodied large language models enable robots to complete complex tasks in unpredictable environments. *Nature Machine Intelligence* **2025**, pp. 1–10.
20. Zheng, Y.; Fu, D.; Hu, X.; Cai, X.; Ye, L.; Lu, P.; Liu, P. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160* **2025**.
21. Shi, Y.; Yu, W.; Li, Z.; Wang, Y.; Zhang, H.; Liu, N.; Mi, H.; Yu, D. MobileGUI-RL: Advancing Mobile GUI Agent through Reinforcement Learning in Online Environment. *arXiv preprint arXiv:2507.05720* **2025**.
22. Cheng, M.; Luo, Y.; Ouyang, J.; Liu, Q.; Liu, H.; Li, L.; Yu, S.; Zhang, B.; Cao, J.; Ma, J.; et al. A survey on knowledge-oriented retrieval-augmented generation. *arXiv preprint arXiv:2503.10677* **2025**.
23. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson Education, 2010.
24. Wang, L.; Ma, C.; Feng, X.; et al. A Survey on Large Language Model based Autonomous Agents. *Frontiers of Computer Science* **2024**.
25. Ng, A. Agentic workflows: The future of AI automation. *DeepLearning.AI Blog* **2024**. Concept of Agentic Workflow.
26. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* **2023**, *36*, 8634–8652.
27. Summers, T.; Yao, S.; Narasimhan, K.; Griffiths, T. Cognitive architectures for language agents. *Transactions on Machine Learning Research* **2023**.
28. Yang, B.; Xu, L.; et al. ContextAgent: Context-Aware Proactive LLM Agents with Open-World Sensory Perceptions. *arXiv preprint arXiv:2505.12345* **2025**.
29. Chawla, R.; Wiest, O.; Zhang, X. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. *Proceedings of IJCAI Survey Track* **2024**.
30. Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Chen, H.; Liu, Z.; Dou, Z.; Wen, J.R. Large language models for information retrieval: A survey. *ACM Transactions on Information Systems* **2023**.
31. Wang, Q.; Zhang, L.; Huang, Y. FinAgent: A multimodal foundation agent for financial trading. *ACM Transactions on Management Information Systems* **2023**, *15*, 1–19.
32. Xu, W.; Huang, C.; Gao, S.; Shang, S. LLM-Based Agents for Tool Learning: A Survey: W. Xu et al. *Data Science and Engineering* **2025**, pp. 1–31.
33. Mei, L.; Yao, J.; Ge, Y.; Wang, Y.; Bi, B.; Cai, Y.; Liu, J.; Li, M.; Li, Z.Z.; Zhang, D.; et al. A Survey of Context Engineering for Large Language Models, 2025, [arXiv:cs.CL/2507.13334].
34. Cao, H.; Jiang, D.; Pei, J.; He, Q.; Liao, Z.; Chen, E.; Li, H. Context-aware query suggestion by mining click-through and session data. In Proceedings of the Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2008; KDD '08, p. 875–883. <https://doi.org/10.1145/1401890.1401995>.
35. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* **2022**, *35*, 24824–24837.
36. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
37. Lopopolo, R. Harness engineering: leveraging Codex in an agent-first world. <https://openai.com/index/harness-engineering/>, 2026. OpenAI Blog.
38. Zhou, Z.; Qu, A.; Wu, Z.; Kim, S.; Prakash, A.; Rus, D.; Zhao, J.; Low, B.K.H.; Liang, P.P. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. *arXiv preprint arXiv:2506.15841* **2025**.

39. Yu, H.; Chen, T.; Feng, J.; Chen, J.; Dai, W.; Yu, Q.; Zhang, Y.Q.; Ma, W.Y.; Liu, J.; Wang, M.; et al. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent. *arXiv preprint arXiv:2507.02259* **2025**.
40. Zhang, G.; Fu, M.; Yan, S. MemGen: Weaving Generative Latent Memory for Self-Evolving Agents. *arXiv preprint arXiv:2509.24704* **2025**.
41. Zhang, X.F.; Beauchamp, N.; Wang, L. PRIME: Large Language Model Personalization with Cognitive Dual-Memory and Personalized Thought Process. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 33695–33724.
42. Wang, Z.Z.; Mao, J.; Fried, D.; Neubig, G. Agent Workflow Memory. In Proceedings of the Forty-second International Conference on Machine Learning, 2025.
43. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative agents: Interactive simulacra of human behavior. In Proceedings of the Proceedings of the 36th annual acm symposium on user interface software and technology, 2023, pp. 1–22.
44. Tan, Z.; Yan, J.; Hsu, I.H.; Han, R.; Wang, Z.; Le, L.; Song, Y.; Chen, Y.; Palangi, H.; Lee, G.; et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 8416–8439.
45. Xiong, Z.; Lin, Y.; Xie, W.; He, P.; Tang, J.; Lakkaraju, H.; Xiang, Z. How Memory Management Impacts LLM Agents: An Empirical Study of Experience-Following Behavior. *arXiv preprint arXiv:2505.16067* **2025**.
46. Wang, Y.; Chen, X. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957* **2025**.
47. Qian, H.; Liu, Z.; Zhang, P.; Mao, K.; Lian, D.; Dou, Z.; Huang, T. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In Proceedings of the Proceedings of the ACM on Web Conference 2025, 2025, pp. 2366–2377.
48. Yang, H.; Lin, Z.; Wang, W.; Wu, H.; Li, Z.; Tang, B.; Wei, W.; Wang, J.; Tang, Z.; Song, S.; et al. Memory3: Language modeling with explicit memory. *arXiv preprint arXiv:2407.01178* **2024**.
49. Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; Wang, Y. Memorybank: Enhancing large language models with long-term memory. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, pp. 19724–19731.
50. Liu, L.; Yang, X.; Shen, Y.; Hu, B.; Zhang, Z.; Gu, J.; Zhang, G. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719* **2023**.
51. Modarressi, A.; Imani, A.; Fayyaz, M.; Schütze, H. RET-LLM: Towards a General Read-Write Memory for Large Language Models. *arXiv e-prints* **2023**, pp. arXiv–2305.
52. Jimenez Gutierrez, B.; Shu, Y.; Gu, Y.; Yasunaga, M.; Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems* **2024**, *37*, 59532–59569.
53. Gutiérrez, B.J.; Shu, Y.; Qi, W.; Zhou, S.; Su, Y. From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. In Proceedings of the Forty-second International Conference on Machine Learning, 2025.
54. Rasmussen, P.; Paliychuk, P.; Beauvais, T.; Ryan, J.; Chalef, D. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956* **2025**.
55. Xu, W.; Mei, K.; Gao, H.; Tan, J.; Liang, Z.; Zhang, Y. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110* **2025**.
56. Zhang, G.; Fu, M.; Wan, G.; Yu, M.; Wang, K.; Yan, S. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems. *arXiv preprint arXiv:2506.07398* **2025**.
57. Yang, H.; Chen, J.; Siew, M.; Botran, T.L.; Joe-Wong, C. LLM-Powered Decentralized Generative Agents with Adaptive Hierarchical Knowledge Graph for Cooperative Planning. In Proceedings of the The First MARW: Multi-Agent AI in the Real World Workshop at AAAI 2025, 2025.
58. Chezelles, D.; Le Sellier, T.; Shayegan, S.O.; Jang, L.K.; Lù, X.H.; Yoran, O.; Kong, D.; Xu, F.F.; Reddy, S.; Cappart, Q.; et al. The browsergym ecosystem for web agent research. *arXiv preprint arXiv:2412.05467* **2024**.
59. Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; Su, Y. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* **2023**, *36*, 28091–28114.
60. He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; Yu, D. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919* **2024**.

61. Shridhar, M.; Yuan, X.; Cote, M.A.; Bisk, Y.; Trischler, A.; Hausknecht, M. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In Proceedings of the International Conference on Learning Representations, 2020.
62. Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. Cogagent: A visual language model for gui agents. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 14281–14290.
63. Andrews, P.; Benhalloum, A.; Bertran, G.M.T.; Bettini, M.; Budhiraja, A.; Cabral, R.S.; Do, V.; Froger, R.; Garreau, E.; Gaya, J.B.; et al. Are: Scaling up agent environments and evaluations. *arXiv preprint arXiv:2509.17158* 2025.
64. Lu, S.; Wang, Z.; Zhang, H.; Wu, Q.; Gan, L.; Zhuang, C.; Gu, J.; Lin, T. Don't Just Fine-tune the Agent, Tune the Environment. *arXiv preprint arXiv:2510.10197* 2025.
65. Yao, S.; Chen, H.; Yang, J.; Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems* 2022, 35, 20744–20757.
66. Zala, A.; Cho, J.; Lin, H.; Yoon, J.; Bansal, M. EnvGen: Generating and Adapting Environments via LLMs for Training Embodied Agents. In Proceedings of the First Conference on Language Modeling, 2024.
67. Chang, M.; 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. *Advances in neural information processing systems* 2024, 37, 74325–74362.
68. Zhang, J.; Yan, Y.; Yan, J.; Zheng, Z.; Piao, J.; Jin, D.; Li, Y. A parallelized framework for simulating large-scale llm agents with realistic environments and interactions. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), 2025, pp. 1339–1349.
69. Wang, B.; Zhang, L.; Wang, Z.; Zhao, Y.; Zhou, T. Core: Cooperative reconstruction for multi-agent perception. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 8710–8720.
70. 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* 2022, 35, 27730–27744.
71. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* 2023.
72. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* 2025.
73. Xia, P.; Chen, J.; Wang, H.; Liu, J.; Zeng, K.; Wang, Y.; Han, S.; Zhou, Y.; Zhao, X.; Chen, H.; et al. SkillRL: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning. *arXiv preprint arXiv:2602.08234* 2026.
74. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.R.; Cao, Y. React: Synergizing reasoning and acting in language models. In Proceedings of the The eleventh international conference on learning representations, 2022.
75. Schick, T.; Dwivedi-Yu, J.; Dessi, 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* 2023, 36, 68539–68551.
76. Patil, S.G.; Zhang, T.; Wang, X.; Gonzalez, J.E. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems* 2024, 37, 126544–126565.
77. Wölflein, G.; Ferber, D.; Truhn, D.; Arandjelovic, O.; Kather, J.N. Llm agents making agent tools. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 26092–26130.
78. Qu, C.; Dai, S.; Wei, X.; Cai, H.; Wang, S.; Yin, D.; Xu, J.; Wen, J.R. From Exploration to Mastery: Enabling LLMs to Master Tools via Self-Driven Interactions. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
79. Wang, R.; Han, X.; Ji, L.; Wang, S.; Baldwin, T.; Li, H. ToolGen: Unified Tool Retrieval and Calling via Generation. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2024.
80. Qin, S.; Zhu, Y.; Mu, L.; Zhang, S.; Zhang, X. Meta-Tool: Unleash Open-World Function Calling Capabilities of General-Purpose Large Language Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 30653–30677.

81. Cai, T.; Wang, X.; Ma, T.; Chen, X.; Zhou, D. Large Language Models as Tool Makers. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
82. Piskala, D.B. Agent, Sub-Agent, Skill, or Tool? A Practitioner's Guide to Extending Agentic AI Systems. *Authorea Preprints* 2026.
83. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems* 2023, 36, 38154–38180.
84. Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; Ghanem, B. Camel: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems* 2023, 36, 51991–52008.
85. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
86. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In Proceedings of the First Conference on Language Modeling, 2024.
87. Klein, L.H.; Potamitis, N.; Aydin, R.; West, R.; Gulcehre, C.; Arora, A. Fleet of Agents: Coordinated Problem Solving with Large Language Models. In Proceedings of the The Exploration in AI Today Workshop at ICML 2025, 2025.
88. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.V.; Chi, E.H.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
89. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* 2023, 36, 11809–11822.
90. Yao, Y.; Li, Z.; Zhao, H. GoT: Effective graph-of-thought reasoning in language models. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 2901–2921.
91. Jiachen, Z.; Yao, Z.; et al. SELF-EXPLAIN: Teaching large language models to reason complex questions by themselves. In Proceedings of the R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models, 2023.
92. Zhou, P.; Pujara, J.; Ren, X.; Chen, X.; Cheng, H.T.; Le, Q.V.; Chi, E.; Zhou, D.; Mishra, S.; Zheng, H.S. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems* 2024, 37, 126032–126058.
93. Zhao, J.; Xie, Y.; Kawaguchi, K.; He, J.; Xie, M. Automatic model selection with large language models for reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 758–783.
94. Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; Neubig, G. Pal: Program-aided language models. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 10764–10799.
95. Chen, W.; Ma, X.; Wang, X.; Cohen, W.W. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Transactions on Machine Learning Research* 2022.
96. Pi, X.; Liu, Q.; Chen, B.; Ziyadi, M.; Lin, Z.; Fu, Q.; Gao, Y.; Lou, J.G.; Chen, W. Reasoning Like Program Executors. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 761–779.
97. Ye, X.; Chen, Q.; Dillig, I.; Durrett, G. Satlm: Satisfiability-aided language models using declarative prompting. *Advances in Neural Information Processing Systems* 2023, 36, 45548–45580.
98. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. In Proceedings of the ICLR, 2024.
99. Surís, D.; Menon, S.; Vondrick, C. Vipergpt: Visual inference via python execution for reasoning. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 11888–11898.
100. Gupta, T.; Kembhavi, A. Visual programming: Compositional visual reasoning without training. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 14953–14962.
101. Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; Gong, Y.; Ni, L.; Shum, H.Y.; Guo, J. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.

102. Li, Y.; Song, D.; Zhou, C.; Tian, Y.; Wang, H.; Yang, Z.; Zhang, S. A framework of knowledge graph-enhanced large language model based on question decomposition and atomic retrieval. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 11472–11485.
103. He, M.; Zhou, A.; Shi, X. Enhancing Textbook Question Answering with Knowledge Graph-Augmented Large Language Models. In Proceedings of the Asian Conference on Machine Learning. PMLR, 2025, pp. 639–654.
104. Wang, R.; Vinh, T.; Xu, R.; Zhou, Y.; Lu, J.; Yang, C.; Pasquel, F. Knowledge Graph Augmented Large Language Models for Next-Visit Disease Prediction. *arXiv preprint arXiv:2512.01210* 2025.
105. Leng, X.; Liang, J.; Mauro, J.; Wang, X.; Bertozzi, A.L.; Chapman, J.; Lin, J.; Chen, B.; Ye, C.; Daniel, T.; et al. Narrative Analysis of True Crime Podcasts With Knowledge Graph-Augmented Large Language Models. *CoRR* 2024.
106. Ji, S.; Liu, L.; Xi, J.; Zhang, X.; Li, X. KLR-KGC: Knowledge-Guided LLM Reasoning for Knowledge Graph Completion. *Electronics (2079-9292)* 2024, 13.
107. Sha, H.; Gong, F.; Liu, B.; Liu, R.; Wang, H.; Wu, T. Leveraging retrieval-augmented large language models for dietary recommendations with traditional Chinese Medicine's medicine food homology: algorithm development and validation. *JMIR Medical Informatics* 2025, 13, e75279.
108. Pan, L.; Albalak, A.; Wang, X.; Wang, W. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 3806–3824.
109. Xin, H.; Ren, Z.; Song, J.; Shao, Z.; Zhao, W.; Wang, H.; Liu, B.; Zhang, L.; Lu, X.; Du, Q.; et al. Deepseek-prover-v1. 5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv preprint arXiv:2408.08152* 2024.
110. Xu, J.; Fei, H.; Pan, L.; Liu, Q.; Lee, M.L.; Hsu, W. Faithful Logical Reasoning via Symbolic Chain-of-Thought. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 13326–13365.
111. Wei, X.; Liu, X.; Zang, Y.; Dong, X.; Cao, Y.; Wang, J.; Qiu, X.; Lin, D. SIM-CoT: Supervised Implicit Chain-of-Thought. *arXiv preprint arXiv:2509.20317* 2025.
112. Chen, Z.; Cui, S.; Ye, D.; Zhang, Y.; Bian, Y.; Zhu, T. Think Consistently, Reason Efficiently: Energy-Based Calibration for Implicit Chain-of-Thought. *arXiv preprint arXiv:2511.07124* 2025.
113. Li, X.; Dong, G.; Jin, J.; Zhang, Y.; Zhou, Y.; Zhu, Y.; Zhang, P.; Dou, Z. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366* 2025.
114. Jang, H.; Jang, Y.; Lee, S.; Ok, J.; Ahn, S. Self-Training Large Language Models with Confident Reasoning. *arXiv preprint arXiv:2505.17454* 2025.
115. Zhang, Z.; Zhang, A.; Li, M.; Karypis, G.; Smola, A.; et al. Multimodal Chain-of-Thought Reasoning in Language Models. *Transactions on Machine Learning Research* 2024.
116. Yang, Z.; Li, L.; Wang, J.; Lin, K.; Azarnasab, E.; Ahmed, F.; Liu, Z.; Liu, C.; Zeng, M.; Wang, L. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381* 2023.
117. Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; Duan, N. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671* 2023.
118. Zheng, G.; Wang, J.; Zhou, X.; Zhang, X. Enhancing Semantics in Multimodal Chain of Thought via Soft Negative Sampling. In Proceedings of the LREC/COLING, 2024.
119. Chai, J.; Tang, S.; Ye, R.; Du, Y.; Zhu, X.; Zhou, M.; Wang, Y.; Zhang, Y.; Zhang, L.; Chen, S.; et al. SciMaster: Towards General-Purpose Scientific AI Agents, Part I. X-Master as Foundation: Can We Lead on Humanity's Last Exam? *arXiv preprint arXiv:2507.05241* 2025.
120. Liu, Z.; Cai, Y.; Zhu, X.; Zheng, Y.; Chen, R.; Wen, Y.; Wang, Y.; Chen, S.; et al. ML-Master: Towards AI-for-AI via Integration of Exploration and Reasoning. *arXiv preprint arXiv:2506.16499* 2025.
121. Curtarolo, S.; Setyawati, W.; Hart, G.L.; Jahnatek, M.; Chepulskii, R.V.; Taylor, R.H.; Wang, S.; Xue, J.; Yang, K.; Levy, O.; et al. AFLOW: An automatic framework for high-throughput materials discovery. *Computational Materials Science* 2012, 58, 218–226.
122. Hu, S.; Lu, C.; Clune, J. Automated Design of Agentic Systems. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
123. Zhang, S.; Fan, J.; Fan, M.; Li, G.; Du, X. Deepanalyze: Agentic large language models for autonomous data science. *arXiv preprint arXiv:2510.16872* 2025.

124. Lu, Y.; Yang, S.; Qian, C.; Chen, G.; Luo, Q.; Wu, Y.; Wang, H.; Cong, X.; Zhang, Z.; Lin, Y.; et al. Proactive Agent: Shifting LLM Agents from Reactive Responses to Active Assistance. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
125. Gao, W.; Liu, Q.; Yue, L.; Yao, F.; Lv, R.; Zhang, Z.; Wang, H.; Huang, Z. Agent4edu: Generating learner response data by generative agents for intelligent education systems. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 23923–23932.
126. Zhang, W.; Tang, K.; Wu, H.; Wang, M.; Shen, Y.; Hou, G.; Tan, Z.; Li, P.; Zhuang, Y.; Lu, W. Agent-Pro: Learning to Evolve via Policy-Level Reflection and Optimization. In Proceedings of the ICLR 2024 Workshop on Large Language Model (LLM) Agents, 2024.
127. Long, L.; He, Y.; Ye, W.; Pan, Y.; Lin, Y.; Li, H.; Zhao, J.; Li, W. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736* 2025.
128. Liu, X.; Qin, B.; Liang, D.; Dong, G.; Lai, H.; Zhang, H.; Zhao, H.; Long, I.L.; Sun, J.; Wang, J.; et al. Autoglm: Autonomous foundation agents for guis. *arXiv preprint arXiv:2411.00820* 2024.
129. Xu, Y.; Liu, X.; Liu, X.; Fu, J.; Zhang, H.; Jing, B.; Zhang, S.; Wang, Y.; Zhao, W.; Dong, Y. MobileRL: Online Agentic Reinforcement Learning for Mobile GUI Agents. *arXiv preprint arXiv:2509.18119* 2025.
130. Zheng, Z.; Yang, M.; Hong, J.; Zhao, C.; Xu, G.; Yang, L.; Shen, C.; Yu, X. DeepEyes: Incentivizing" Thinking with Images" via Reinforcement Learning. *arXiv preprint arXiv:2505.14362* 2025.
131. Jiang, P.; Lin, J.; Cao, L.; Tian, R.; Kang, S.; Wang, Z.; Sun, J.; Han, J. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223* 2025.
132. Chhikara, P.; Khant, D.; Aryan, S.; Singh, T.; Yadav, D. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413* 2025.
133. Jin, B.; Zeng, H.; Yue, Z.; Yoon, J.; Arik, S.; Wang, D.; Zamani, H.; Han, J. Search-rl: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516* 2025.
134. Chen, M.; Sun, L.; Li, T.; Sun, H.; Zhou, Y.; Zhu, C.; Wang, H.; Pan, J.Z.; Zhang, W.; Chen, H.; et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470* 2025.
135. Li, K.; Zhang, Z.; Yin, H.; Zhang, L.; Ou, L.; Wu, J.; Yin, W.; Li, B.; Tao, Z.; Wang, X.; et al. WebSailor: Navigating Super-human Reasoning for Web Agent. *arXiv preprint arXiv:2507.02592* 2025.
136. Wei, Y.; Duchenne, O.; Copet, J.; Carbonneaux, Q.; ZHANG, L.; Fried, D.; Synnaeve, G.; Singh, R.; Wang, S. SWE-RL: Advancing LLM Reasoning via Reinforcement Learning on Open Software Evolution. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025.
137. Zhou, Y.; Jiang, S.; Tian, Y.; Weston, J.; Levine, S.; Sukhbaatar, S.; Li, X. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478* 2025.
138. Sun, W.; Lu, M.; Ling, Z.; Liu, K.; Yao, X.; Yang, Y.; Chen, J. Scaling Long-Horizon LLM Agent via Context-Folding. *arXiv preprint arXiv:2510.11967* 2025.
139. Li, X.; Zou, H.; Liu, P. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383* 2025.
140. Li, Z.; Hu, Y.; Wang, W. Encouraging good processes without the need for good answers: Reinforcement learning for llm agent planning. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2025, pp. 1654–1666.
141. Goldie, A.; Mirhoseini, A.; Zhou, H.; Cai, I.; Manning, C.D. Synthetic data generation & multi-step rl for reasoning & tool use. *arXiv preprint arXiv:2504.04736* 2025.
142. Singh, J.; Magazine, R.; Pandya, Y.; Nambi, A. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441* 2025.
143. Mai, X.; Xu, H.; Li, Z.Z.; Wang, W.; Hu, J.; Zhang, Y.; Zhang, W.; et al. Agent rl scaling law: Agent rl with spontaneous code execution for mathematical problem solving. *arXiv preprint arXiv:2505.07773* 2025.
144. Xue, Z.; Zheng, L.; Liu, Q.; Li, Y.; Zheng, X.; MA, Z.; An, B. SimpleTIR: End-to-End Reinforcement Learning for Multi-Turn Tool-Integrated Reasoning. In Proceedings of the NeurIPS 2025 Fourth Workshop on Deep Learning for Code, 2025.
145. Dou, Z.; Zhao, Q.; Wan, Z.; Zhang, D.; Wang, W.; Raiyan, T.; Chen, B.; Pan, Q.; Ouyang, Y.; Gao, Z.; et al. Plan Then Action: High-Level Planning Guidance Reinforcement Learning for LLM Reasoning. *arXiv preprint arXiv:2510.01833* 2025.
146. Li, Z.; Zhang, H.; Han, S.; Liu, S.; Xie, J.; Zhang, Y.; Choi, Y.; Zou, J.; Lu, P. In-the-Flow Agentic System Optimization for Effective Planning and Tool Use. In Proceedings of the NeurIPS 2025 Workshop on Efficient Reasoning, 2025.

147. Mialon, G.; Fourrier, C.; Wolf, T.; LeCun, Y.; Scialom, T. Gaia: a benchmark for general ai assistants. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
148. Phan, L.; Gatti, A.; Han, Z.; Li, N.; Hu, J.; Zhang, H.; Zhang, C.B.C.; Shaaban, M.; Ling, J.; Shi, S.; et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249* 2025.
149. Yuan, Y.; Jianye, H.; Ma, Y.; Dong, Z.; Liang, H.; Liu, J.; Feng, Z.; Zhao, K.; ZHENG, Y. Uni-RLHF: Universal Platform and Benchmark Suite for Reinforcement Learning with Diverse Human Feedback. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
150. Guo, Z.; Cheng, S.; Wang, H.; Liang, S.; Qin, Y.; Li, P.; Liu, Z.; Sun, M.; Liu, Y. StableToolBench: Towards Stable Large-Scale Benchmarking on Tool Learning of Large Language Models. In Proceedings of the ACL (Findings), 2024.
151. Li, L.; Wang, Y.; Zhao, H.; Kong, S.; Teng, Y.; Li, C.; Wang, Y. Reflection-Bench: Evaluating Epistemic Agency in Large Language Models. In Proceedings of the Forty-second International Conference on Machine Learning, 2025.
152. Wu, D.; Wang, H.; Yu, W.; Zhang, Y.; Chang, K.W.; Yu, D. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
153. Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. AgentBench: Evaluating LLMs as Agents. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
154. Kazemi, M.; Fatemi, B.; Bansal, H.; Palowitch, J.; Anastasiou, C.; Mehta, S.V.; Jain, L.K.; Aglietti, V.; Jindal, D.; Chen, Y.P.; et al. Big-bench extra hard. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 26473–26501.
155. Wu, C.K.; Tam, Z.R.; Lin, C.Y.; Chen, Y.N.V.; Lee, H.y. Streambench: Towards benchmarking continuous improvement of language agents. *Advances in Neural Information Processing Systems* 2024, 37, 107039–107063.
156. Xiao, R.; Ma, W.; Wang, K.; Wu, Y.; Zhao, J.; Wang, H.; Huang, F.; Li, Y. FlowBench: Revisiting and Benchmarking Workflow-Guided Planning for LLM-based Agents. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 10883–10900.
157. Chen, J.; Wei, Z.; Ren, Z.; Li, Z.; Zhang, J. LR²Bench: Evaluating Long-chain Reflective Reasoning Capabilities of Large Language Models via Constraint Satisfaction Problems. In Proceedings of the ACL (Findings), 2025, pp. 6006–6032.
158. Geng, L.; Chang, E.Y. Realm-bench: A real-world planning benchmark for llms and multi-agent systems. *arXiv preprint arXiv:2502.18836* 2025.
159. Hu, Y.; Wang, Y.; McAuley, J. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257* 2025.
160. Xia, Y.; Jin, P.; Xie, S.; He, L.; Cao, C.; Luo, R.; Liu, G.; Wang, Y.; Liu, Z.; Chen, Y.J.; et al. Nature Language Model: Deciphering the Language of Nature for Scientific Discovery. *arXiv preprint arXiv:2502.07527* 2025.
161. He, Y.; Huang, G.; Feng, P.; Lin, Y.; Zhang, Y.; Li, H.; et al. Pasa: An llm agent for comprehensive academic paper search. *arXiv preprint arXiv:2501.10120* 2025.
162. Li, Y.; Chen, L.; Liu, A.; Yu, K.; Wen, L. ChatCite: LLM agent with human workflow guidance for comparative literature summary. In Proceedings of the Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 3613–3630.
163. Schmidgall, S.; Su, Y.; Wang, Z.; Sun, X.; Wu, J.; Yu, X.; Liu, J.; Liu, Z.; Barsoum, E. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227* 2025.
164. Ghafarollahi, A.; Buehler, M.J. ProtAgents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery* 2024, 3, 1389–1409.
165. Ni, Y.; Zhu, L.; Li, S. Bio AI Agent: A Multi-Agent Artificial Intelligence System for Autonomous CAR-T Cell Therapy Development with Integrated Target Discovery, Toxicity Prediction, and Rational Molecular Design. *arXiv preprint arXiv:2511.08649* 2025.
166. Zheng, Y.; Sun, S.; Qiu, L.; Ru, D.; Jiayang, C.; Li, X.; Lin, J.; Wang, B.; Luo, Y.; Pan, R.; et al. OpenResearcher: Unleashing AI for Accelerated Scientific Research. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2024, pp. 209–218.
167. Baek, J.; Jauhar, S.K.; Cucerzan, S.; Hwang, S.J. Researchagent: Iterative research idea generation over scientific literature with large language models. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 6709–6738.

168. Zhang, W.; Zhao, L.; Xia, H.; Sun, S.; Sun, J.; Qin, M.; Li, X.; Zhao, Y.; Zhao, Y.; Cai, X.; et al. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In Proceedings of the Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining, 2024, pp. 4314–4325.
169. Yu, Y.; Yao, Z.; Li, H.; Deng, Z.; Jiang, Y.; Cao, Y.; Chen, Z.; Suchow, J.; Cui, Z.; Liu, R.; et al. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems* **2024**, *37*, 137010–137045.
170. Wang, X.; Li, B.; Song, Y.; Xu, F.F.; Tang, X.; Zhuge, M.; Pan, J.; Song, Y.; Li, B.; Singh, J.; et al. Openhands: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741* **2024**.
171. Yu, J.; Zhuang, Y.; Sun, Y.; Gao, W.; Liu, Q.; Cheng, M.; Huang, Z.; Chen, E. TestAgent: An Adaptive and Intelligent Expert for Human Assessment. *arXiv preprint arXiv:2506.03032* **2025**.
172. Gao, W.; Liu, Q.; Yue, L.; Yao, F.; Lv, R.; Zhang, Z.; Wang, H.; Huang, Z. Agent4edu: Generating learner response data by generative agents for intelligent education systems. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 23923–23932.
173. Lv, R.; Liu, Q.; Gao, W.; Zhang, H.; Lu, J.; Zhu, L. GenAL: Generative Agent for Adaptive Learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 577–585.
174. Zhan, Y.; Liu, Q.; Gao, W.; Zhang, Z.; Wang, T.; Shen, S.; Lu, J.; Huang, Z. CoderAgent: simulating student behavior for personalized programming learning with large language models. In Proceedings of the Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, 2025, IJCAI '25.
175. Yu, S.; Cheng, M.; Liu, Q.; Wang, D.; Yang, J.; Ouyang, J.; Luo, Y.; Lei, C.; Chen, E. Multi-source knowledge pruning for retrieval-augmented generation: A benchmark and empirical study. In Proceedings of the Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 3931–3941.
176. Ye, J.; Du, Z.; Yao, X.; Lin, W.; Xu, Y.; Chen, Z.; Wang, Z.; Zhu, S.; Xi, Z.; Yuan, S.; et al. ToolHop: A Query-Driven Benchmark for Evaluating Large Language Models in Multi-Hop Tool Use. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 2995–3021.
177. Liu, Z.; Chen, B.; Cheng, M.; Chen, E.; Li, L.; Lei, C.; Ou, W.; Li, H.; Gai, K. Towards Context-aware Reasoning-enhanced Generative Searching in E-commerce. *arXiv preprint arXiv:2510.16925* **2025**.
178. Cheng, M.; Wang, J.; Wang, D.; Tao, X.; Liu, Q.; Chen, E. Can slow-thinking llms reason over time? empirical studies in time series forecasting. *arXiv preprint arXiv:2505.24511* **2025**.
179. Jiang, C.; Cheng, M.; Tao, X.; Mao, Q.; Ouyang, J.; Liu, Q. TableMind: An Autonomous Programmatic Agent for Tool-Augmented Table Reasoning. *arXiv preprint arXiv:2509.06278* **2025**.
180. Li, Q.; Cheng, M.; Liu, Z.; Wang, D.; Zeng, Y.; Liu, T. From Hypothesis to Premises: LLM-based Backward Logical Reasoning with Selective Symbolic Translation. *arXiv preprint arXiv:2512.03360* **2025**.
181. Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohllhart, P.; Welker, S.; Wahid, A.; et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Proceedings of the Conference on Robot Learning. PMLR, 2023, pp. 2165–2183.
182. Zhang, X.; Gao, T.; Cheng, M.; Pan, B.; Guo, Z.; Liu, Y.; Tao, X. AlphaCast: A Human Wisdom-LLM Intelligence Co-Reasoning Framework for Interactive Time Series Forecasting. *arXiv preprint arXiv:2511.08947* **2025**.
183. Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R.K.W.; Lim, E.P. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Long Papers*; Association for Computational Linguistics (ACL), 2023; pp. 2609–2634.
184. Lin, B.Y.; Fu, Y.; Yang, K.; Brahman, F.; Huang, S.; Bhagavatula, C.; Ammanabrolu, P.; Choi, Y.; Ren, X. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems* **2023**, *36*, 23813–23825.
185. Gao, J.; Juluri, S. From Idea to Co-Creation: A Planner-Actor-Critic Framework for Agent Augmented 3D Modeling. *arXiv preprint arXiv:2601.05016* **2026**.
186. 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* **2020**, *33*, 9459–9474.
187. Cheng, M.; Wang, D.; Liu, Q.; Yu, S.; Tao, X.; Wang, Y.; Chu, C.; Duan, Y.; Long, M.; Chen, E. Mind2Report: A Cognitive Deep Research Agent for Expert-Level Commercial Report Synthesis. *arXiv preprint arXiv:2601.04879* **2026**.

188. Song, C.H.; Wu, J.; Washington, C.; Sadler, B.M.; Chao, W.L.; Su, Y. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 2998–3009.
189. Karimova, S.; Dadashova, U. The model context protocol: a standardization analysis for application integration. *Journal of Computer Science and Digital Technologies* **2025**, *1*, 50–59.
190. Chen, B.; Shu, C.; Shareghi, E.; Collier, N.; Narasimhan, K.; Yao, S. FireAct: Toward Language Agent Fine-tuning. *CoRR* **2023**, *abs/2310.05915*.
191. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhunoye, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* **2023**, *36*, 46534–46594.
192. Gou, Z.; Shao, Z.; Gong, Y.; Yang, Y.; Duan, N.; Chen, W.; et al. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
193. Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.V.; et al. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
194. Lin, S.; Hilton, J.; Evans, O. Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research* **2022**.
195. Pan, T.; Ouyang, J.; Cheng, M.; Li, Q.; Liu, Z.; Pan, M.; Yu, S.; Liu, Q. PaperScout: An Autonomous Agent for Academic Paper Search with Process-Aware Sequence-Level Policy Optimization. *arXiv preprint arXiv:2601.10029* **2026**.
196. Ye, R.; Zhang, Z.; Li, K.; Yin, H.; Tao, Z.; Zhao, Y.; Su, L.; Zhang, L.; Qiao, Z.; Wang, X.; et al. AgentFold: Long-Horizon Web Agents with Proactive Context Management. *arXiv preprint arXiv:2510.24699* **2025**.
197. Lai, H.; Liu, X.; Zhao, Y.; Xu, H.; Zhang, H.; Jing, B.; Ren, Y.; Yao, S.; Dong, Y.; Tang, J. Computerrl: Scaling end-to-end online reinforcement learning for computer use agents. *arXiv preprint arXiv:2508.14040* **2025**.
198. Wang, H.; Zou, H.; Song, H.; Feng, J.; Fang, J.; Lu, J.; Liu, L.; Luo, Q.; Liang, S.; Huang, S.; et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544* **2025**.
199. Hong, J.; Zhao, C.; Zhu, C.; Lu, W.; Xu, G.; Yu, X. DeepEyesV2: Toward Agentic Multimodal Model. *arXiv preprint arXiv:2511.05271* **2025**.
200. Ouyang, J.; Yan, R.; Luo, Y.; Cheng, M.; Liu, Q.; Liu, Z.; Yu, S.; Wang, D. Training powerful llm agents with end-to-end reinforcement learning, 2025.
201. Song, H.; Jiang, J.; Min, Y.; Chen, J.; Chen, Z.; Zhao, W.X.; Fang, L.; Wen, J.R. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592* **2025**.
202. Team, T.D.; Li, B.; Zhang, B.; Zhang, D.; Huang, F.; Li, G.; Chen, G.; Yin, H.; Wu, J.; Zhou, J.; et al. Tongyi DeepResearch Technical Report. *arXiv preprint arXiv:2510.24701* **2025**.
203. Zhang, H.; Cheng, M.; Luo, Y.; Tao, X. STaR: Towards Cognitive Table Reasoning via Slow-Thinking Large Language Models. *arXiv preprint arXiv:2511.11233* **2025**.
204. Paglieri, D.; Cupiał, B.; Cook, J.; Piterbarg, U.; Tuyls, J.; Grefenstette, E.; Foerster, J.N.; Parker-Holder, J.; Rocktäschel, T. Learning When to Plan: Efficiently Allocating Test-Time Compute for LLM Agents. *arXiv preprint arXiv:2509.03581* **2025**.
205. Luo, Y.; Zhou, Y.; Cheng, M.; Wang, J.; Wang, D.; Pan, T.; Zhang, J. Time Series Forecasting as Reasoning: A Slow-Thinking Approach with Reinforced LLMs. *arXiv preprint arXiv:2506.10630* **2025**.
206. Luo, X.; Zhang, Y.; He, Z.; Wang, Z.; Zhao, S.; Li, D.; Qiu, L.K.; Yang, Y. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680* **2025**.
207. Wang, D.; Ouyang, J.; Yu, S.; Cheng, M.; Liu, Q. Claw-R1: Agentic RL for Modern Agents. <https://github.com/AgentR1/Claw-R1>, 2025. GitHub repository.
208. Ouyang, J.; Pan, T.; Cheng, M.; Yan, R.; Luo, Y.; Lin, J.; Liu, Q. Hoh: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. *arXiv preprint arXiv:2503.04800* **2025**.
209. Wang, D.; Cheng, M.; Yu, S.; Liu, Z.; Guo, Z.; Liu, Q. Paperarena: An evaluation benchmark for tool-augmented agentic reasoning on scientific literature. *arXiv preprint arXiv:2510.10909* **2025**.
210. Gottweis, J.; Weng, W.H.; Daryin, A.; Tu, T.; Palepu, A.; Sirkovic, P.; Myaskovsky, A.; Weissenberger, F.; Rong, K.; Tanno, R.; et al. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864* **2025**.
211. Manus. Introducing Manus 1.6: Max Performance, Mobile Dev, and Design View. <https://manus.im/blog/manus-max-release>, 2025. Accessed: 2026-02-06.

212. Yang, J.; Jimenez, C.E.; Wettig, A.; Lieret, K.; Yao, S.; Narasimhan, K.; Press, O. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems* **2024**, *37*, 50528–50652.
213. Zhang, C.; Yang, Z.; Liu, J.; Li, Y.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; Yu, G. Appagent: Multimodal agents as smartphone users. In Proceedings of the Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, 2025, pp. 1–20.
214. Zhang, C.; Li, L.; He, S.; Zhang, X.; Qiao, B.; Qin, S.; Ma, M.; Kang, Y.; Lin, Q.; Rajmohan, S.; et al. Ufo: A ui-focused agent for windows os interaction. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 597–622.

Short Biography of Authors



Mingyue Cheng received his PhD degree in Data Science from the University of Science and Technology of China (USTC). He is currently serving as an Associate Researcher at USTC, affiliated with the State Key Laboratory of Cognitive Intelligence and School of Computer Science and Technology. His research interests encompass time series and sequence modeling, biomedical big data, recommender systems. Dr. Cheng has contributed several papers to renowned conference proceedings and journals, including KDD, WWW, SIGIR, WSDM, ICDM, and ACM Transactions on Knowledge and Data Engineering (TKDE). He has also served on the program committees for top-tier conferences and as a reviewer for international journals.



Daoyu Wang received the B.E. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, where he is currently pursuing the master's degree with the School of Computer Science and Technology and the State Key Laboratory of Cognitive Intelligence. His research interests include retrieval-augmented generation and LLM agents. He has authored or co-authored several papers in premier conferences such as ICML and WWW.



Shuo Yu received the B.E. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, where he is currently pursuing the master's degree with the School of Artificial Intelligence and Data Science and the State Key Laboratory of Cognitive Intelligence. His research interests include retrieval-augmented generation and LLM agents. He has authored or co-authored several papers in premier conferences such as CIKM and WWW.



Qingchuan Li received the B.E. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, and is currently pursuing the master's degree with the School of Computer Science and Technology and the State Key Laboratory of Cognitive Intelligence. His research interests include LLM logical reasoning and LLM agents, and he has authored or co-authored several papers in premier conferences such as AACL and WWW.



Jie Ouyang received the B.E. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, where he is currently pursuing the master's degree with the School of Computer Science and Technology and the State Key Laboratory of Cognitive Intelligence. His research interests include retrieval-augmented generation and reinforcement learning. He has authored or co-authored several papers in premier conferences such as ACL and KDD.



Yucong Luo received his bachelor's degree in Data Science and Big Data Technology from the University of Science and Technology of China (USTC), Hefei, China. He is currently pursuing a master's degree at the School of Artificial Intelligence and Data Science, USTC, and the State Key Laboratory of Cognitive Intelligence. His research interests include large language model agents (LLM Agents) and LLM-enhanced recommendation systems (LLM4Rec). He has authored or co-authored several papers published in top-tier international conferences, including AAAI, ICML, WWW, NeurIPS, and ACL.



Yiju Zhang received the B.E. degree from Zhengzhou University. He is currently pursuing the master's degree with the School of Artificial Intelligence and Data Science, University of Science and Technology of China (USTC), and the State Key Laboratory of Cognitive Intelligence. His research interests include LLM agents and agentic RL.



Qi Liu (Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2013. He is currently a Professor with the School of Computer Science and Technology. He has authored or coauthored papers published prolifically in refereed journals and conference proceedings, including IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Information Systems, and KDD conference. His research interests include data mining and intelligent education. Dr. Liu is an Associate Editor for IEEE TRANSACTIONS ON BIG DATA and Neurocomputing. He was the recipient of KDD' 18 Best Student Paper Award, ICDM' 11 Best Research Paper Award, and China Outstanding Youth Science Foundation in 2019. He is a Member of the Alibaba DAMO Academy Young Fellow.



Enhong Chen (Fellow, IEEE) received the PhD degree from the University of Science and Technology of China (USTC), in 1996. He is currently a professor of USTC. His research areas include data mining and knowledge discovery, machine learning and artificial intelligence. His research is supported by the National Science Foundation for Distinguished Young Scholars of China. He has published more than 200 papers in refereed conferences and journals, including IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Information Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Neural Networks and Learning Systems, ICML, NeurIPS, KDD, ICLR and AAAI. He is an associate editor of IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Systems, Man, and Cybernetics, ACM Transactions on Intelligent Systems and Technology, World Wide Web Journal. He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of ICKG-2020, a program co-chair for PAKDD-2022. He received the Best Application Paper Award on KDD 2008, the Best Student Paper Award on KDD 2018 (Research), the Best Research Paper Award on ICDM 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.