DEEP RESEARCH AGENTS: A SYSTEMATIC EXAMINATION AND ROADMAP

Yuxuan Huang[†1], Yihang Chen[†2], Haozheng Zhang[†2], Kang Li[3], Meng Fang[1], Linyi Yang[4], Xiaoguang Li[2], Lifeng Shang[2], Songcen Xu[2], Jianye Hao[2], Kun Shao[‡2], and Jun Wang[‡4]

[1] University of Liverpool [2] Huawei Noah's Ark Lab [3] University of Oxford [4] University College London

ABSTRACT

The rapid progress of Large Language Models (LLMs) has given rise to a new category of autonomous AI systems, referred to as Deep Research (DR) agents. These agents are designed to tackle complex, multi-turn informational research tasks by leveraging a combination of dynamic reasoning, adaptive long-horizon planning, multi-hop information retrieval, iterative tool use, and the generation of structured analytical reports. In this paper, we conduct a detailed analysis of the foundational technologies and architectural components that constitute Deep Research agents. We begin by reviewing information acquisition strategies, contrasting API-based retrieval methods with browser-based exploration. We then examine modular tool-use frameworks, including code execution, multimodal input processing, and the integration of Model Context Protocols (MCPs) to support extensibility and ecosystem development. To systematize existing approaches, we propose a taxonomy that differentiates between static and dynamic workflows, and we classify agent architectures based on planning strategies and agent composition, including single-agent and multi-agent configurations. We also provide a critical evaluation of current benchmarks, highlighting key limitations such as restricted access to external knowledge, sequential execution inefficiencies, and misalignment between evaluation metrics and the practical objectives of DR agents. Finally, we outline open challenges and promising directions for future research. A curated and continuously updated repository of DR agent research is available at: https://github.com/ai-agents-2030/awesome-deep-research-agent.

1 Introduction

Recent advances in large language models (LLMs) have led to the rapid emergence of sophisticated AI agents capable of autonomous research. Early models such as GPT-3 [10] primarily addressed isolated tasks, including question answering and machine translation. Subsequently, integration with external tools enabled models such as WebGPT [66] to navigate the web and synthesize information from diverse sources autonomously. Most recently, a new class of advanced autonomous systems, termed Deep Research (DR) agents, has emerged, exemplified by industry-leading solutions such as OpenAI DR [69], Gemini DR [30], Grok DeepSearch [107], and Perplexity DR [72]. These deep research agents significantly extend LLMs by incorporating advanced reasoning, dynamic task planning, and adaptive interaction with web resources and analytical tools.

Formally, we define **"Deep Research Agents"** as:

AI agents powered by LLMs, integrating dynamic reasoning, adaptive planning, multi-iteration external data retrieval and tool use, and comprehensive analytical report generation for informational research tasks.

Specifically, DR agents leverage LLMs as their cognitive core, retrieving external knowledge in real-time through web browsers and structured APIs, and dynamically invoking analytical tools via customized toolkits or standardized interfaces such as the Model Context Protocol (MCP). This architecture enables DR agents to autonomously manage complex, end-to-end research workflows by seamlessly integrating reasoning processes with multimodal resources.

Compared with traditional Retrieval-Augmented Generation (RAG) methods [85], which primarily enhance factual accuracy but lack sustained reasoning capabilities [16], and conventional Tool Use (TU) systems [76] that heavily

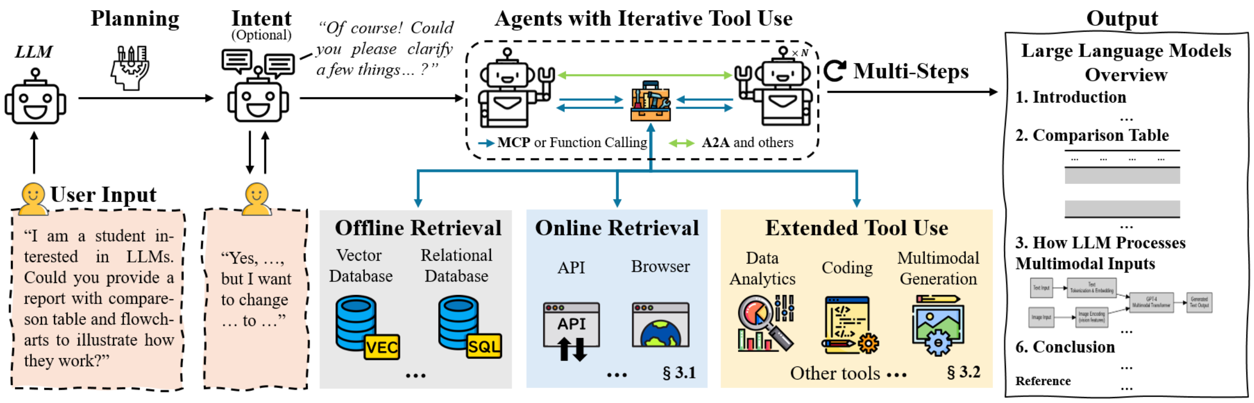


Figure 1: A structural overview of a DR agent in an multi-agent architecture for ease of illustration.

depend on pre-defined workflows [100], DR agents offer significantly greater autonomy, continual and deep reasoning abilities, dynamic task planning, and adaptive real-time interaction. These advanced capabilities uniquely position DR agents to handle complex, evolving, and knowledge-intensive research scenarios. A representative example of such a DR agent architecture is illustrated in Figure 1, which demonstrates the complete workflow from user input through optional planning and intent clarification, to iterative tool utilization encompassing offline retrieval (vector and relational databases), online retrieval (APIs and browsers), and extended capabilities including data analytics, coding (etc.), and multimodal generation, ultimately producing comprehensive structured report.

Contribution. This survey systematically reviews recent advancements in DR agents, providing a comprehensive analysis of core technologies, methodologies, optimization pipelines, and representative implementations. Specifically, the contributions of this survey include:

- A thorough analysis of representative DR systems, explicitly examining their system architectures, retrieval mechanisms, tool invocation methods, and performance characteristics, alongside optimization and tuning paradigms.
- A unified classification framework (Figure 4) that systematically categorizes DR systems based on workflow characteristics (static versus dynamic), planning strategies, and agent-based architectures (single-agent versus multi-agent), bridging diverse technical methodologies and current industrial solutions.
- A systematic review and categorization of existing benchmarks utilized to evaluate DR systems, highlighting how these benchmarks assess critical capabilities, such as retrieval accuracy, reasoning depth, and adaptive tool invocation proficiency.
- A systematic analysis of critical open challenges and research directions, focusing on expanding retrieval scope beyond traditional methods, enabling asynchronous parallel execution, developing comprehensive multi-modal benchmarks, and optimizing multi-agent architectures for enhanced robustness and efficiency.

Survey Organization. This survey methodically explores recent advancements in DR agents, organized as follows: Section 2 provides foundational concepts, examining recent progress in reasoning, retrieval-augmented generation, and agent communication protocols. Section 3 comprehensively analyzes key DR agent components, including search engine integration (Section 3.1), tool invocation strategies (Section 3.2), architectural workflows (Section 3.3), and optimization methodologies (Section 3.4). Section 4 reviews major industrial applications and practical implementations of DR agents by leading organizations. Section 5 surveys benchmarks used for evaluating DR systems, categorizing them into question-answering and task execution scenarios. Section 6 highlights critical challenges and outlines promising directions for future research, focusing on enhancing information acquisition, asynchronous parallel execution, benchmark alignment, and optimizing multi-agent architectures. Finally, Section 7 concludes with a summary and provides insights into the broader implications and opportunities within DR agent research.

2 Background and Preliminaries

2.1 Advances in Reasoning and Tool Integration

Recent advancements in large reasoning models (LRMs) have greatly enhanced the ability of language models to tackle complex and abstract tasks. These models have shown significant improvements in tasks such as arithmetic, common-sense reasoning, and symbolic problem-solving, largely due to innovations in model architectures and training techniques. One such advancement is Chain-of-Thought (CoT) prompting, introduced by Wei et al. [102], which explicitly guides models to articulate intermediate logical steps, decomposing complex problems into simpler, sequential stages. This has led to notable improvements in both the interpretability and accuracy of LLMs on various reasoning benchmarks. Building upon CoT, subsequent research has introduced methods to further enhance LLM reasoning, particularly in handling lengthy textual contexts. Approaches such as positional interpolation and sparse attention mechanisms [8, 99] have been proposed to extend the effective context window. Furthermore, specialized benchmarks like LongBench [9] and LongFinanceQA [55] have been developed to rigorously evaluate and improve the performance of these models in extended-context reasoning.

To address reasoning tasks that require real-time or specialized external knowledge, frameworks like Toolformer [77] and MultiTool-CoT [41] have been proposed, enabling LLMs to autonomously incorporate external computational resources and APIs directly within reasoning workflows. These approaches effectively enhance performance in tasks dependent on precise numerical calculations and dynamic information retrieval. Maintaining reasoning coherence across multiple conversational turns also poses distinct challenges. Techniques such as Dialogue CoT [12] and Structured CoT (SCoT) [89] explicitly integrate dialogue states and conversational contexts within reasoning chains, significantly improving coherence, context-awareness, and the ability to manage iterative interactions and clarify complex user queries. However, despite substantial improvements, existing reasoning frameworks still encounter critical issues, including hallucinations, static or outdated internal knowledge, and insufficient responsiveness to rapidly changing information needs. These limitations highlight the necessity of integrating external information sources, real-time retrieval mechanisms, and adaptive reasoning strategies, which are core motivations driving recent advances toward more comprehensive and robust reasoning frameworks suitable for DR Agent applications.

2.2 Advances in Retrieval-Augmented Generation and Agentic Retrieval

Retrieval-augmented Generation (RAG), leveraging external knowledge bases (e.g., webs, APIs), has emerged as an effective strategy to mitigate hallucination problems and enhance the accuracy of web information search [22, 26, 85]. Early RAG architectures typically involved a static pipeline, where retrievers fetched relevant documents from external sources such as Wikipedia or search engines, and generators (e.g., LLMs) produced answers based solely on these retrieved passages. However, static approaches were limited in handling complex or multi-step queries, motivating recent advances toward iterative and interactive retrieval mechanisms to generate richer and more relevant responses, including FLARE [115], Self-RAG [7], IAG [116], and ToC [49]. In addition, studies [43, 56] expanded retrieval sources from structured databases (e.g., Wikipedia) to large-scale, diverse web corpora such as the Common Crawl dump preprocessed via the CCNet pipeline [25]. Further improvements of RAG include hybrid approaches that combine internal LLM knowledge and external retrievals for better accuracy and coherence [6]. Recently, Huang et al. [39] proposed RAG-RL, introducing reinforcement learning and curriculum learning techniques, enabling reasoning language models (RLMs) to more effectively identify and utilize relevant contexts.

Despite these advancements in retrieval methods and reasoning-enhanced models, RAG approaches still face limitations in effectively managing complex reasoning workflows and dynamically adapting to varied task requirements. To address these challenges, recent research extends RAG into an agentic paradigm, integrating additional reasoning and decision-making layers atop conventional RAG pipelines [85]. Agentic RAG approaches leverage iterative retrieval, adaptive querying, and dynamic workflow adjustments, significantly enhancing multi-step reasoning capabilities. For example, RL-based query refinement techniques (e.g., Hsu et al. [37]) improve retrieval for complex queries, while graph-based retrieval (e.g., GeAR [83]) further enhances the processing of multi-hop queries. Despite these advancements, agentic RAG still faces critical challenges, including balancing computational overhead from dynamic reasoning processes [85], aligning agent behaviors with user intentions [114], and ensuring interpretability in adaptive workflows [37, 85]. Moreover, even advanced agentic RAG approaches remain constrained by their reliance on pre-existing or periodically updated corpora, limiting their ability to handle real-time, rapidly changing, or long-tail information needs effectively. Addressing this challenge requires integrating external APIs and web browsing capabilities into RAG architectures, motivating recent DR methods aimed at further enhancing retrieval comprehensiveness and adaptability.

2.3 Model Context Protocol and Agent-to-Agent Policy

Model Context Protocol (MCP) and Agent-to-Agent (A2A) have been proposed to address interoperability challenges in LLM-based agent systems, enabling efficient tool access and effective multi-agent collaboration. **MCP**: Traditional Tool Use (TU) agents face significant challenges, including inconsistent APIs, high maintenance costs, and redundant development efforts, severely limiting interoperability across systems [77]. To address these issues, Anthropic introduced the MCP, a unified communication layer allowing LLM-based agents to interact securely and consistently with external services and data sources via standardized interfaces. MCP mitigates data silo problems by providing dynamic service discovery and uniform access patterns. **A2A**: Google's A2A protocol facilitates decentralized multi-agent collaboration through structured, task-oriented dialogues. Agents from diverse vendors and model architectures can discover peers, delegate responsibilities, and collaboratively manage complex tasks as equal participants [29]. By abstracting agent discovery into Agent Cards, and task coordination into Tasks and Artifacts, A2A supports flexible, incremental, multi-modal workflows, ideally suited to sophisticated collaborative scenarios.

MCP and A2A complement each other by clearly separating responsibilities: MCP serves as a standardized interface for accessing external tools, while A2A orchestrates collaborative agent interactions. Together, they establish a modular and scalable foundation for open, interoperable agent ecosystems, significantly enhancing the practical capabilities of AI systems in tackling complex real-world challenges.

3 Deep Research: Search Engine, Tool Use, Workflow, Tuning, Non-parametric Continual Learning

Comparison with Coventional RAG-based Approaches. DR agents expand the capabilities of traditional RAG methods by integrating dynamic retrieval, real-time TU, and adaptive reasoning into a unified system. RAG-based approaches typically rely on fixed pipelines, limiting their flexibility in handling complex, multi-step queries or rapidly changing contexts. In contrast, DR agents provide greater autonomy, context-awareness, and accuracy by dynamically engaging with external tools and managing multi-stage research tasks in real time.

In this section, we explore five core components essential for the development and optimization of DR agents: (3.1) **search engine integration**, which compares API-based interfaces with browser-based exploration to enhance dynamic knowledge acquisition; (3.2) **Tool Use capabilities**, which investigate the integration of code execution, mathematical computation, file manipulation, and multimodal processing modules within the agent's inference pipeline; (3.3) **workflow architecture**, analyzing foundational designs, the balance between multi-agent and single-agent paradigms, memory mechanisms, and auxiliary components that facilitate the orchestration of complex research workflows; (3.4) **tuning methodologies**, which examine prompt-driven structured generation, LLM-driven prompting, fine-tuning strategies, and reinforcement learning approaches aimed at optimizing agent performance, and (3.5) **Non-parametric continual learning**, which enables LLM agents to self-evolve by dynamically adapting external tools, memory, and workflows without updating internal model weights, offering scalable optimization for complex tasks.

3.1 Search Engine: API vs. Browser

To enhance reasoning depth and accuracy for handling evolving tasks, DR agents employ search engines (SE) to update their knowledge through interaction with the external environment. In Table 1, we present a comparative overview of SEs, base models, and evaluation benchmarks employed by existing DR agents. The SEs can be broadly categorized into two types:

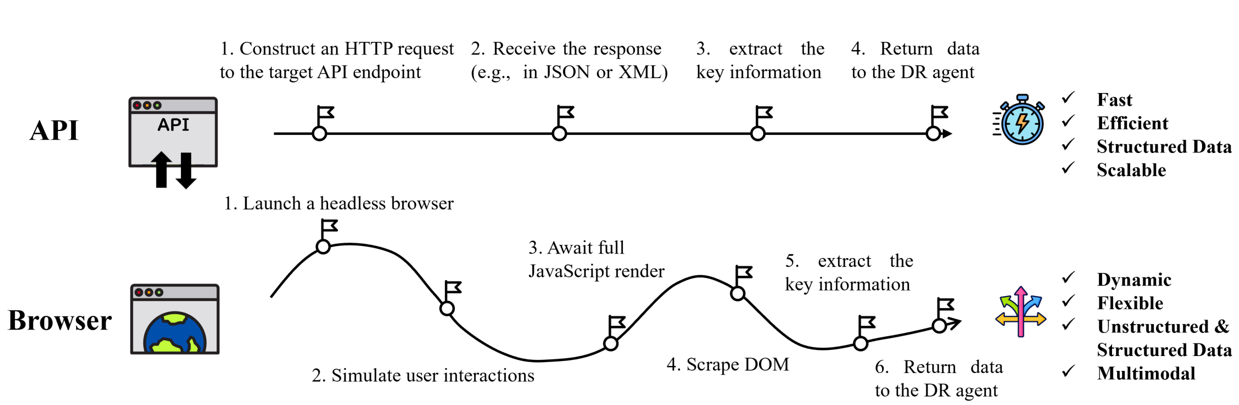


Figure 2: General Comparison of API-Based and Browser-Based Retrieval Workflow.

Table 1: Comparison of DR Agents with Search Engine Details

■ = Primary focus, ▨ = Secondary/minor focus, □ = Not present

DR Agent	Search Engine API	Search Engine Browser	Benchmark GAIA	Benchmark HLE	Benchmark Other QA	Base Model	Release
Avatar [106]	▨	□	□	□	Stark	Claude-3-Opus, GPT-4	Feb-2024
CoSearch-Agent [28]	■	□	□	□	□	GPT-3.5-turbo	Feb-2024
MMAC-Copilot [87]	■	□	■	□	□	GPT-3.5, GPT-4	Mar-2024
Storm [81]	▨	□	□	□	FreshWiki	GPT-3.5-turbo	Jul-2024
OpenResearcher [118]	■	□	□	□	Privately Collected QA Data	DeepSeek-V2-Chat	Aug-2024
The AI Scientist [57]	■	□	□	□	MLE-Bench	GPT-4o, o1-mini, o1-preview	Aug-2024
Gemini DR [30]	■	■	□	■	GPQA	Gemini-2.0-Flash	Dec-2024
Agent Laboratory [79]	■	□	□	□	MLE-Bench	GPT-4o, o1-preview	Jan-2025
Search-o1 [52]	■	□	□	□	GPQA, NQ, TriviaQA	QwQ-32B-preview	Jan-2025
Agentic Reasoning [105]	■	□	□	□	GPQA	DeepSeek-R1, Qwen2.5	Feb-2025
AutoAgent [91]	□	■	■	□	□	Claude-Sonnet-3.5	Feb-2025
Grok DeepSearch [107]	■	■	□	□	GPQA	Grok3	Feb-2025
OpenAI DR [69]	□	■	■	■	■	GPT-o3	Feb-2025
Perplexity DR [72]	■	▨	□	■	SimoleQA	Flexible	Feb-2025
AgentRxiv [78]	■	□	□	□	GPQA, MedQA	GPT-4o-mini	Mar-2025
Agent-R1 [71]	■	□	□	□	HotpotQA	Qwen2.5-1.5B-Inst	Mar-2025
AutoGLM Rumination[119]	□	■	□	□	GPQA	GLM-Z1-Air	Mar-2025
Copilot Researcher [63]	□	■	□	□	□	o3-mini	Mar-2025
H2O.ai DR [35]	■	■	■	□	□	h2ogpt-oasst1-512-12b	Mar-2025
Manus [60]	■	■	□	□	□	Claude3.5, GPT-4o	Mar-2025
Openmanus [54]	■	■	□	□	□	Claude3.5, GPT-4o	Mar-2025
OWL [11]	■	■	■	□	□	Deepeek-R1, Gemini2.5-Pro, GPT-4o	Mar-2025
R1-Searcher [86]	▨	□	□	□	2WikiMultiHopQA, HotpotQA	Llama3.1-8B-Inst, Qwen2.5-7B	Mar-2025
ReSearch [15]	▨	□	□	□	2WikiMultiHopQA, HotpotQA	Qwen2.5-7B, Qwen2.5-7B-Inst	Mar-2025
Search-R1 [47]	▨	□	□	□	2WikiMultiHopQA, HotpotQA, NQ, TriviaQA	Llama3.2-3B, Qwen2.5-3B/7B	Mar-2025
DeepResarcher [117]	□	■	□	□	HotpotQA, NQ, TriviaQA	Qwen2.5-7B-Inst	Apr-2025
Genspark Super Agent [94]	■	■	■	□	□	Mixture of Agents*	Apr-2025
WebThinker [53]	■	□	■	■	GPQA, WebWalkerQA	QwQ-32B	Apr-2025
SWIRL [27]	□	■	□	□	HotQA, BeerQA	Gemma 2-27b	Apr-2025
SimpleDeepSearcher [90]	□	■	■	□	2WikiMultiHopQA	Qwen-2.5-7B-In, Qwen-2.5-32B-In, DeepSeek-Distilled-Qwen-2.5-32B, QwQ-32B	Apr-2025
Suna AI [4]	■	■	□	□	□	GPT-4o, Claude	Apr-2025
AgenticSeek [61]	□	■	□	□	□	GPT-4o, DeepSeek-R1, Claude	May-2025
Alita [75]	■	■	■	□	PathVQA	GPT-4o, Claude-Sonnet-4	May-2025
DeerFlow [19]	■	□	□	□	□	Doubao-1.5-Pro-32k, DeepSeek-R1, GPT-4o, Qwen	May-2025
PANGU DEEPDIVER [84]	■	□	□	□	C-SimpleQA, HotpotQA, ProxyQA	Pangu-7B-Reasoner	May-2025

1) **API-Based SEs**, which interact with structured data sources, such as search-engine APIs or scientific database APIs, enabling efficient retrieval of organized information.
2) **Browser-Based SEs**, which simulate human-like interactions with web pages, facilitating real-time extraction of dynamic or unstructured content, improving the comprehensiveness of the external knowledge.

API-based retrieval is a fast, efficient, structured, and scalable method that allows DR agents to access external knowledge sources with relatively less time and computational cost. For instance, Gemini DR leverages multi-source interfaces, most notably the Google Search API and the arXiv API, to perform large-scale retrieval across hundreds to thousands of web pages, thereby significantly expanding its information coverage. Grok DeepSearch [107] claims to ensure both the freshness and depth of its knowledge base by maintaining a continuous index via news-outlet feeds, the Wikipedia API, and X's native interface, and by activating a query-driven agent on demand to generate targeted sub-queries and fetch relevant pages in real time. AgentLaboratory [79] uses the arXiv API to extract paper metadata

*Mixture of Agents refers to an ensemble of nine base models comprising GPT-4.1, GPT-o3, GPT-o4-mini-high, Claude-Sonnet-3.7-Thinking, Claude-Sonnet-3.7, Gemini-2.0-Flash, Gemini-2.5-Pro, DeepSeek-V3, DeepSeek-R1

Table 2: Comparison of DR Agents with Tool Use Capabilities

■ = Involved, ▩ = Non Disclosure, □ = Not present

DR Agent	Code Interpreter	Data Analytics	Multimodal	Release
CoSearchAgent [28]	□	■	□	Feb-2024
Storm [81]	■	□	□	Jul-2024
The AI Scientist [57]	■	□	□	Aug-2024
Agent Laboratory [79]	■	□	□	Jan-2025
Agentic Reasoning [105]	■	□	□	Feb-2025
AutoAgent [91]	■	□	■	Feb-2025
Genspark DR [94]	■	■	■	Feb-2025
Grok DeepSearch [107]	■	■	■	Feb-2025
OpenAI DR [69]	■	■	■	Feb-2025
Perplexity DR [72]	■	■	■	Feb-2025
Agent-R1 [71]	■	□	□	Mar-2025
AutoGLM Romination [119]	■	□	■	Mar-2025
Copilot Researcher [63]	■	■	▩	Mar-2025
Manus [60]	■	■	■	Mar-2025
OpenManus [54]	■	■	□	Mar-2025
OWL [11]	■	■	■	Mar-2025
H2O.ai DR [35]	■	■	■	Mar-2025
Genspark Super Agent [94]	■	■	■	Apr-2025
WebThinker [53]	■	□	□	Apr-2025
Suna Ai [4]	■	■	□	Apr-2025
AgenticSeek [61]	■	■	□	May-2025
Alita [75]	■	▩	▩	May-2025
DeerFlow [19]	■	■	□	May-2025

and abstracts for automated literature reviews. AI Scientist [57]issues requests to the Semantic Scholar API to validate the novelty and citation relationships of model-generated research ideas, and CoSearchAgent [28] integrates SerpApi to deliver Slack-based, real-time search responses. DeepRetrieval [44], operating within a reinforcement-learning framework, optimizes queries against the PubMed and ClinicalTrials.gov APIs to maximize recall on biomedical tasks, and Search-o1 [52] combines the Bing Search API with the Jina Reader API to dynamically extract and refine passages for downstream reasoning. Whilst these API-driven methods excel at structured, high-throughput data acquisition, they generally struggle when faced with deeply nested client-side JavaScript-rendered content, interactive components, or authentication barriers, thereby motivating the development of browser-based search mechanisms capable of comprehensively extracting and analyzing dynamic or unstructured information.

Browser-based retrieval provides DR agents with dynamic, flexible, and interactive access to multimodal and unstructured web content through simulated human-like browser interactions. For example, Manus AI's browsing agent operates a sandboxed Chromium instance for each research session, programmatically opening new tabs, issuing search queries, clicking through result links, scrolling pages until content thresholds are met, filling out form elements when necessary, executing in-page JavaScript to reveal lazily loaded sections, and downloading files or PDFs for local analysis [60]. Although OpenAI DR, Grok DeepSearch, and Gemini 2.5 DR do not publicly disclose the implementation details of their browsing capabilities, their ability to handle interactive widgets, dynamically rendered content, and multi-step navigation strongly suggests that they too employ comparable headless-browser frameworks behind the scenes. Among open-source studies, AutoAgent [113] operates within a BrowserGym environment to scroll, interact with page components, and download files when APIs are unavailable [113]; DeepResearcher [117] employs a dedicated Web Browsing Agent that, upon receiving a browse request, processes each segment of a webpage in turn, decides whether to continue to subsequent segments based on relevance, and incrementally aggregates pertinent information into a short-term memory buffer before returning it for reasoning. While browser-based retrieval excels at capturing real-time and deeply nested content that API calls cannot reach, it also incurs greater latency, resource consumption, and complexity in handling page variability and errors, suggesting that DR agents may benefit from hybrid architectures that combine the efficiency of API-based methods with the comprehensiveness of browser-driven exploration.

3.2 Tool Use: Empowering Agents with Extended Functionalities

To expand DR agents' capacity to interact with external environments in complex research tasks, specifically by actively invoking and handling diverse tools and data sources, various DR agents have introduced three core tool modules: code interpreters, data analytics, multimodal processing, along with the Model Context Protocol.

Code Interpreter. The code interpreter capability enables DR agents to execute scripts during inference, allowing them to perform data processing, algorithm verification and model simulation. Most DR agents, except CoSearchAgent, embed a script execution environment. They typically rely on Python utilities such as Aider and Java utilities to orchestrate dynamic scripting, conduct literature-driven analysis and carry out real-time computational reasoning.

Data Analytics. By integrating data analytics modules, DR agents transform raw retrievals into structured insights by computing summary statistics, generating interactive visualizations and conducting quantitative model evaluations, thereby accelerating hypothesis testing and decision-making. Many commercial DR agents have implemented analytics features such as charting, table generation and statistical analysis, either locally or via remote services. However, most of these systems have not publicly disclosed technical details of their implementations. In contrast, academic studies often provide concrete examples: CoSearchAgent integrates SQL-based queries within team communication platforms to run aggregate analyses and produce reports; AutoGLM extracts and analyzes structured datasets directly from table-based web interfaces; and Search-o1's Reason-in-Documents component refines lengthy retrieved texts before extracting key metrics for downstream evaluation.

Multimodal Processing and Generation. Multimodal processing and generation tools enable DR agents to integrate, analyze and generate heterogeneous data such as text, images, audio and video within a unified reasoning pipeline, thereby enriching their contextual understanding and broadening the range of their outputs. Only a subset of mature commercial and open-source projects, for example Manus, OWL, AutoAgent, AutoGLM, OpenAI, Gemini, Perplexity and Grok DeepSearch, support this capability, whereas most academic prototypes have not implemented it, often due to the high computational cost. As the typical open source studies, OWL and Openmanus extend their pipelines to include interactions with platforms such as GitHub, Notion and Google Maps and to leverage numerical libraries such as Sympy and Excel for combined data analysis and multimodal media processing [11, 54].

Deep Research Agent with Computer Use. Most recently, the boundaries of DR agents have been progressively expanded through integrating computer-assisted task execution capabilities (i.e., computer use). For example, Zhipu AI introduced AutoGLM Rumination [119], a RL-based system incorporating self-reflection and iterative refinement mechanisms, which significantly enhances multi-step reasoning and advanced function-calling abilities. Specifically, AutoGLM Rumination autonomously interacts with web environments, executes code, invokes external APIs, and effectively accomplishes sophisticated tasks, including data retrieval, analysis, and structured generation of comprehensive reports. Comparison with OpenAI's DR: While OpenAI DR primarily focus on intricate reasoning and information retrieval, AutoGLM Rumination exhibits superior autonomy in practical execution. This enhanced autonomy allows it to transform abstract analytical insights into concrete operational tasks, such as automated interactions with web interfaces and real-time data processing. Moreover, AutoGLM Rumination addresses and resolves limitations inherent in simulated browsing environments by seamlessly integrating advanced reasoning capabilities with authentic browser-based interactions. Therefore, the agent gains reliable access to user-authenticated resources, including platforms such as CNKI, Xiaohongshu, and WeChat official accounts. Such integration significantly elevates the agent's autonomy and adaptability in both information acquisition and execution of real-world tasks.

3.3 Architecture and Workflow

As shown in Figure 4, this section systematically analyzes the construction of DR systems, focusing on workflows categorized into **static** and **dynamic** types. We then discuss planning strategies, which enhance task allocation and execution through **three distinctive user interaction types** to clarify intent: planning-only (direct planning without clarifying user intent), intent-to-planning (clarifying intent before planning to align the task with user goals), and unified intent-planning (generating a plan and requesting user confirmation). The distinction between **single-agent** and **multi-agent** systems is examined in the context of dynamic workflows, emphasizing specialization in task management. Additionally, we examine memory mechanisms for managing and integrating retrieved information, which enhance the performance and adaptability of DR systems.

3.3.1 Static vs. Dynamic Workflows

Static Workflows. Static workflows rely on manually predefined task pipelines, decomposing research processes into sequential subtasks executed by dedicated agents. These workflows follow explicitly structured procedures, making them particularly suitable for well-defined, structured research scenarios. For instance, AI Scientist [57] automates scientific discovery through distinct sequential phases, including ideation, experimentation, and reporting. Similarly, Agent Laboratory [79] segments research activities into formalized stages, such as literature review, experimentation, and synthesis of findings. Extending this static paradigm further, AgentRxiv [78] incorporates inter-agent collaboration mechanisms, enabling incremental knowledge reuse through sharing intermediate research outcomes among specialized

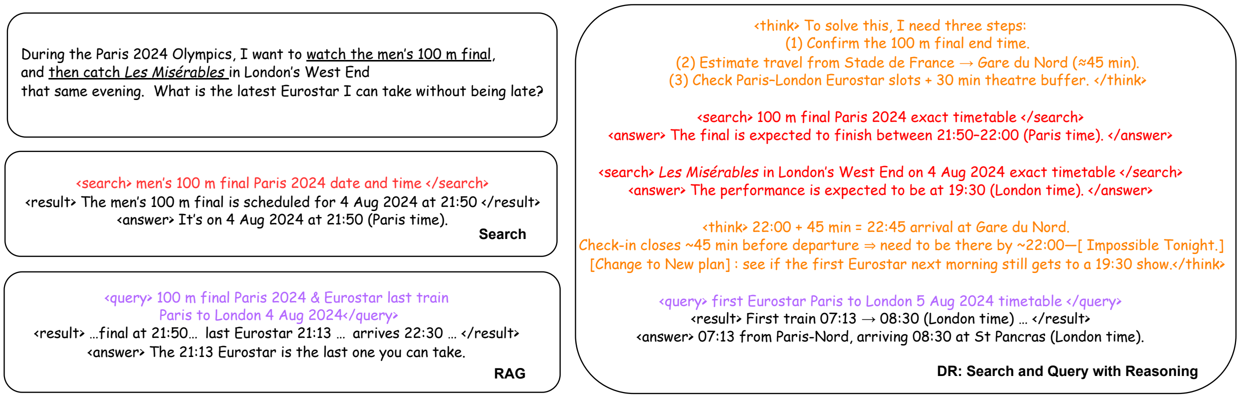


Figure 3: **Comparison of Information Retrieval Methods.** The upper left corner (Search) represents the searching methods, which can use the browser or API; the lower left corner (RAG, Query) represents Retrieval-Augmented Generation, combining retrieval and generative models to output natural language answers; the right side (Deep Research) represents the deep research process, generating complex decisions or analyses through retrieval and explicit reasoning.

agents. Whist their ease of implementation and structured clarity, static workflows suffer from limited generalization capabilities, as each distinct task necessitates a specifically tailored pipeline.

Dynamic Workflows. To overcome the limitations in flexibility and generalizability inherent in static workflows, dynamic workflows support adaptive task planning, allowing agents to dynamically reconfigure task structures based on iterative feedback and evolving contexts. Dynamic architectures leverage advanced mechanisms including automated planning, iterative refinement, and interactive task allocation, enabling tasks to evolve in real-time as new knowledge or external inputs become available. Consequently, dynamic workflows exhibit superior generality and adaptability, making them highly suitable for complex, knowledge-intensive tasks commonly encountered in AI-driven research scenarios.

3.3.2 Dynamic Workflows: Planning Strategies

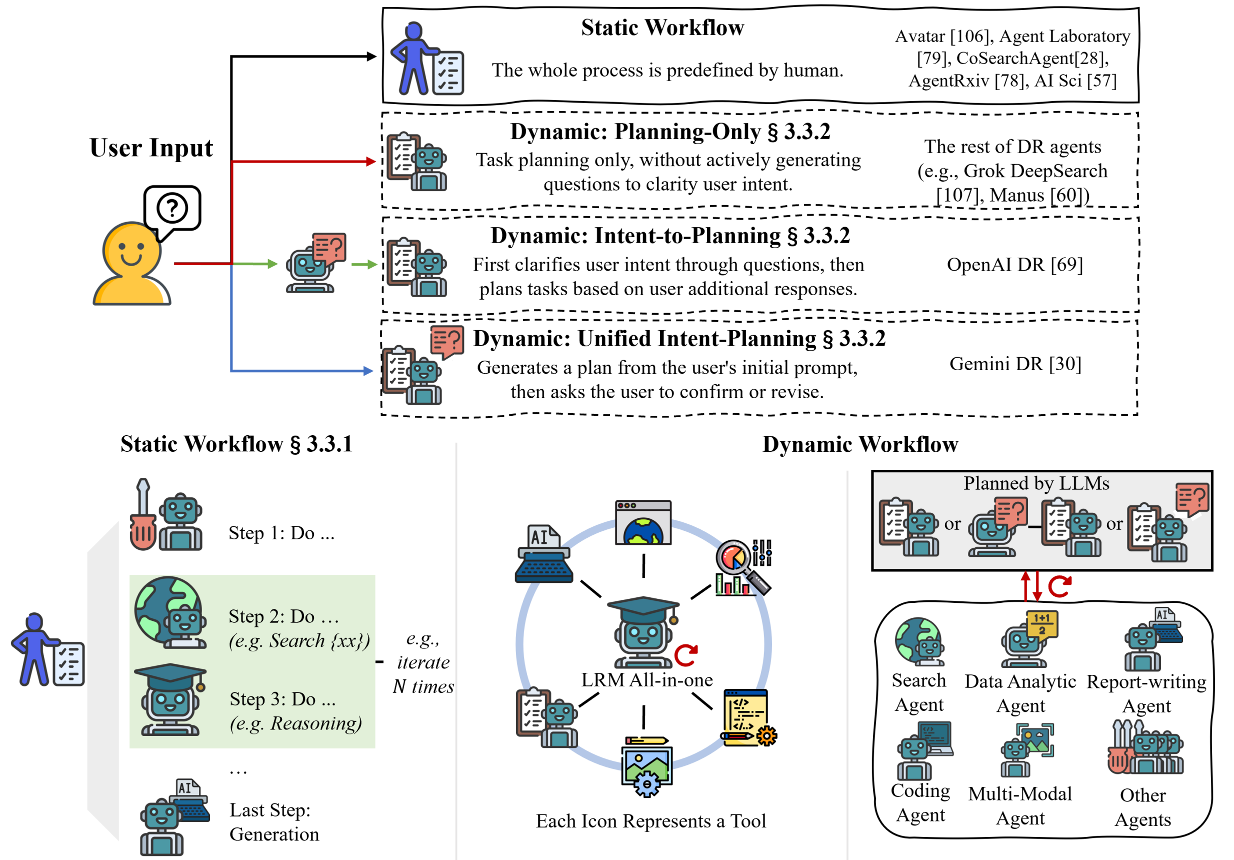
To enhance DR agents' adaptability in response to evolving user requirements and contexts, existing studies propose three distinctive LLM-based planning strategies, each differing in whether and how they interact with the user to clarify intent:

1) The **Planning-Only** approach directly generates task plans based solely on initial user prompts without actively engaging in further clarification, adopted by the majority of existing DR agents, including Grok [107], H2O [35] and Manus [60].
2) The **Intent-to-Planning** strategy actively clarifies user intent prior to planning through targeted questions, subsequently generating tailored task sequences based on clarified user inputs; this method is utilized by OpenAI DR [69].
3) The **Unified Intent-Planning** approach synthesizes these methods by generating a preliminary plan from the initial prompt, together with interactively engaging the user to confirm or revise the proposed plan. Gemini DR [30] is representative of this strategy, effectively adopts the strength of user-guided refinement.

3.3.3 Dynamic Workflows: Single-Agent vs. Multi-Agent

Dynamic workflows of DR agents can be differentiated based on agent architectures into single-agent and multi-agent frameworks, each exhibiting distinct characteristics concerning task specialization, coordination complexity, and scalability of execution.

Dynamic Single-Agent Systems. Dynamic single-agent systems **integrate planning, tool invocation, and execution within a unified LRM**, streamlining task management into a cohesive cognitive loop. Single-agent architectures autonomously refine task plans and invoke appropriate tools based on evolving contexts, typically without explicit inter-agent coordination. Compared to multi-agent architectures, single-agent systems enable direct end-to-end reinforcement learning (RL) optimization across the entire workflow, facilitating smoother and more coherent integration of reasoning,



(a) Static Workflow: Manually predefined pipeline; fixed agent roles; simple yet limited adaptability, requiring tailored designs for each distinct task.

[106],[79],[28],[78],[57]

(b) Dynamic-Single-Agent Workflow § 3.3.3: A powerful LRM iteratively generates and updates plans, dynamically invokes tools (e.g., search, coding), and produces reports.

[118],[52],[105],[71],[86],[47],[117],[52],[53],[69],[30],[107],[72],[15],[27],[90],[61],[4],[75],[61],[119]

(c) Dynamic-Multi-Agent Workflow § 3.3.3: LLMs generate plans, assign tasks to agents, refine plans from real-time response, iteratively reassign tasks until produce final reports.

[81],[91],[54],[60],[11],[84],[35],[63],[94],[19],[87]

Figure 4: Comparison of DR Workflows: (1) **Static vs. Dynamic Workflows**: Static workflows rely on predefined task sequences, while dynamic workflows allow LLM-based task planning. (2) **Planning Strategies**: Three types include: planning-only (direct planning without clarifying user intent), intent-to-planning (clarifying intent before planning), and unified intent-planning (generating a plan and requesting user confirmation). (3) **Single-Agent vs. Multi-Agent**: Dynamic workflows can be categorized to dynamic-multi-agent systems (tasks distributed across specialized agents) or dynamic-single-agent systems (a LRM autonomously updates and executes tasks).

planning, and tool invocation. Systems such as Agent-R1 [71], ReSearch [15], and Search-R1 [47] exemplify this paradigm through iterative cycles of explicit reasoning, action, and reflection, aligning with the ReAct framework [110]. However, this streamlined approach places significant demands on the foundation model's reasoning capabilities, contextual understanding, and autonomous selection and invocation of tools. Additionally, the tightly integrated nature of single-agent systems may limit modular flexibility, complicating independent scaling or optimization of individual functional components.

Dynamic Multi-Agent Systems. Dynamic multi-agent systems leverage multiple specialized agents to collaboratively execute subtasks generated and dynamically allocated through adaptive planning strategies. These systems typically employ hierarchical or centralized planning mechanisms, wherein a coordinator agent continuously assigns and redistributes tasks based on real-time feedback and replanning. Representative frameworks include OpenManus [54] and Manus [60], both adopting hierarchical planner-toolcaller architectures. Similarly, OWL [11] includes a workforce-oriented model, utilizing a central manager agent to orchestrate task distribution among specialized execution agents. Furthermore, Alita [75] incorporates a self-evolution mechanism into DR agents, allowing the agent to online instantiate and configure new MCP servers tailored to specific tasks and environmental conditions. Such multi-agent configurations effectively handle complex, parallelizable research tasks, thereby enhancing flexibility and scalability in open-ended research scenarios. Nevertheless, a major current challenge of multi-agent systems lies in the inherent complexity of coordinating multiple independent agents, making it difficult to conduct effective end-to-end reinforcement learning optimization.

3.3.4 Memory Mechanism for Long-Context Optimization

Memory mechanisms empower DR agents to persistently capture, organize, and recall relevant information across multiple retrieval rounds, thereby reducing redundant queries and improving both the efficiency and coherence of DR tasks. During the DR process, agents typically perform extensive multi-round retrieval, generating hundreds of thousands of tokens (or even millions). Although recent advances in LLMs have significantly expanded context window sizes, current limits still constrain tasks involving extremely long contexts. To address these challenges, DR systems have implemented various optimizations for processing extended contexts. Broadly, these optimizations can be categorized into three main strategies: **(i) Expanding the Context Window Length; (ii) Compressing Intermediate Steps; (iii) Utilizing External Structured Storage for Temporary Results**.

Extending the context window length is the **most intuitively effective** approach, exemplified by Google's Gemini model [30], which supports a context window of up to one million tokens, supplemented by a RAG setup. Despite its straightforwardness, this method often incurs high computational costs and may lead to inefficiencies in resource utilization during practical deployments.

An alternative strategy involves compressing or summarizing intermediate reasoning steps, significantly reducing the number of tokens processed by the model and thereby improving both efficiency and output quality. Representative frameworks such as The AI Scientist [57] and CycleResearcher [103] pass summarized intermediate results between workflow phases. Further, Li et al. [52] introduced the concept of "Reason-in-Documents," utilizing LRMs to compress documents, substantially reducing token volume and enhancing model decision-making efficiency. However, a potential drawback of this approach is the loss of detailed information, potentially impacting the precision of subsequent reasoning.

Utilizing external structured storages for preserving and retrieving historical information enables DR agents to persistently and efficiently store vast amounts of past context beyond the constraints of the context window, improving memory capacity, retrieval speed, and semantic relevance. Popular open-source frameworks such as Manus [60], OWL [11], Open Manus [54], and Avatar [106] utilize external file systems to store intermediate outcomes and historical data for subsequent retrieval. Frameworks like WebThinker [53] and AutoAgent [91] have developed self-managing modules that leverage vector databases to support scalable memory storage and fast similarity-based lookup. Beyond plain text or vector stores, some works propose more semantically structured memory frameworks: for instance, Wu et al. [105] employ knowledge graphs to capture intermediate reasoning processes and thereby enhance the precision of information reuse, while Agentrxiv [78] simulates an academic repository akin to arXiv for storing and retrieving relevant outcomes from other agents. Although these structured approaches offer superior semantic retrieval efficiency and accuracy, they typically entail higher development and maintenance costs due to the need for meticulous data structure design and management.

3.4 Tuning: Beyond Prompting toward Capability Enhancement

Parametric Approaches. Prompt-based methods directly leverage the capabilities of pre-trained LLMs, enabling complex functionalities without expensive fine-tuning or additional training. However, it remains challenging to

Table 3: Comparison of DR Agents with Tuning Methods

■ = Yes, ▨ = Yes but details unknown, □ = Not present

DR Agent	SFT	RL	Base Model	Data	Reward Design	Release
Gemini DR [30]	▨	▨	Gemini-2.0-Flash	□	▨	Dec-2024
Grok DeepSearch [107]	□	▨	Grok3	□	▨	Feb-2025
OpenAI DR [69]	□	▨	GPT-o3	□	▨	Feb-2025
Agent-R1 [71]	□	PPO [80], Reinforce++ [38], GRPO [82]	Qwen2.5-1.5B-Inst	HotpotQA	Rule-Outcome	Mar-2025
AutoGLM Romination [119]	▨	▨	GLM-Z1-Air	□	▨	Mar-2025
H2O.ai DR [35]	■	▨	h2ogpt-oasst1-512-12b	□	▨	Mar-2025
Copilot Researcher [63]	▨	▨	o3-mini	□	□	Mar-2025
ReSearch [15]	□	GRPO	Qwen2.5-7B-Inst, Qwen2.5-32B-Inst	2WikiMultiHopQA	Rule-Outcome	Mar-2025
R1-Searcher [86]	■	Reinforce++, GRPO	Qwen2.5-7B-InSt, LLaMA-3.1-8B-Inst	2WikiMultiHopQA, HotpotQA	Rule-Outcome	Mar-2025
Search-R1 [47]	■	PPO, GRPO	Qwen2.5-3B/7B, LLaMA3.2-3B-Inst	NQ, HotpotQA	Rule-Outcome	Mar-2025
DeepResearcher [117]	□	GRPO	Qwen2.5-7B-Inst	NQ, HotpotQA	Rule-Outcome	Apr-2025
Genspark Super Agent [94]	□	□	Mixture of Agents	□	□	Apr-2025
WebThinker [53]	■	Iterative Online DPO	QwQ-32B	Expert Dataset	Rule-Outcome	Apr-2025
SWIRL [27]	□	Offline-RL	Gemma-2-27B	HotPotQA	□	Apr-2025
SimpleDeepSearcher [90]	■	PPO	Qwen-2.5-7B-In, Qwen-2.5-32B-In, Deepseek-Distilled-Qwen-32B, QwQ-32B	NQ, HotpotQA, 2WikiMultiHopQA, Musique, SimpleQA, MultiHop-RAG	Process-based reward	Apr-2025
PANGU DEEPDIVER [84]	■	GRPO	Pangu-7B-Reasoner	WebPuzzle	Rule-Outcome	May-2025

systematically optimize prompt structures and workflows. Moreover, since an agent's performance is inherently limited by its backbone LLM, increasing the complexity of decision-making processes quickly reaches the model's performance ceiling. To overcome these limitations, it is essential to incorporate advanced optimization techniques such as fine-tuning, reinforcement learning (RL) or hybrid training paradigms to further extend the model's inherent capabilities. Below, we discuss the two main tuning paradigms, supervised fine-tuning (SFT) and RL, and highlight how each extends agent capabilities beyond prompt-only methods.

3.4.1 SFT-based Optimization

Prompt-based approaches, while effective for rapid adaptation, are fundamentally constrained by the intrinsic generalization capacity of backbone LLMs and often exhibit limited robustness in complex task settings. In order to address these limitations, researchers have increasingly explored fine-tuning methodologies aimed at systematically optimizing LLMs for critical components of deep research agents. These components include search query formulation, structured report generation, and external tool utilization. These efforts aim to enhance retrieval quality, mitigate hallucinations, and enable more reliable long-form and evidence-grounded generation.

An early milestone in this research direction is Open-RAG [42], which augments data construction with diverse supervisory signals, including retrieval tokens, relevance tokens, grounding tokens, and utility tokens. Through adversarial training, Open-RAG improves the model's capability to filter irrelevant information, thereby enhancing both retrieval accuracy and the quality of downstream tasks. Building upon this foundation, AUTO-RAG [113] enhances the autonomous iterative retrieval capabilities of LLMs. In contrast to earlier multi-hop retrieval approaches that relied on few-shot prompting or hand-crafted templates [45, 23, 97], AUTO-RAG constructs reasoning-grounded instruction datasets, enabling models to autonomously plan retrieval queries and engage in multi-round interactions with retrievers. The model dynamically refines its retrieval strategy during generation, gathering sufficient evidence before synthesizing a final answer. Extending these retrieval-centric innovations, DeepRAG [32] proposes a binary tree search mechanism that recursively generates sub-queries and constructs multi-turn retrieval trajectories. This mechanism enables the model to judiciously balance between internal parametric knowledge and external retrieval-based rollouts. Consequently, it enhances search efficiency and mitigates redundant external queries.

In order to further reduce reliance on manually constructed supervised fine-tuning (SFT) datasets, recent work has sought to reduce dependence on manually constructed supervised fine-tuning datasets by developing fine-tuning strategies based on rejection sampling. CoRAG [98] uses rejection sampling to extract intermediate retrieval chains from standard question answering datasets, allowing for stepwise retrieval augmentation and dynamic reformulation of subqueries as context evolves instead of supervising only final outputs. Li et al. [51] propose a hint-infer mechanism that monitors token patterns during generation and triggers external computational tools, such as Python executors or hint libraries when specific cues are detected. After an initial supervised fine-tuning phase, the model undergoes a rejection sampling fine-tuning process that teaches it to generate its own prompts and invoke tools autonomously without reliance on hand-curated demonstrations. ATLAS [17] proposes a novel approach for LLM-based agents that trains exclusively on selected critical steps from expert trajectories, significantly improving generalization performance.

Although these SFT methods enhance the generalization of deep research agents by supporting dynamic retrieval planning, structured information synthesis, and integrated tool use, they remain confined to offline, static retrieval pipelines characteristic of retrieval-augmented systems. In contrast, reinforcement learning offers a more adaptive solution for online query generation and tool invocation. By learning from real-time reward signals, reinforcement learning agents acquire the ability to formulate effective search queries and determine the optimal timing for tool calls. This approach addresses the limitations of synthetic demonstration data and distributional shifts, yielding more robust and adaptive performance in open-ended research environments.

3.4.2 Reinforcement Learning-based Optimization

RL-based methods optimize DR agents by directly enhancing their adaptive capabilities and generalization across diverse tasks, surpassing conventional instruction-following or pattern learning approaches. Recent advances have demonstrated that end-to-end RL training significantly strengthens iterative information retrieval, dynamic tool invocation, and integrated reasoning capabilities within DR agents. See comparative analysis in Table 3.

Early RL-based approaches such as DeepRetrieval [44] optimized query generation for improved information retrieval quality, effectively enhancing downstream text generation by producing more relevant search results. Building on query optimization, ReSearch [15] extended RL to adaptive reasoning over retrieved information. The model dynamically refined search strategies and iteratively updated results based on continuous feedback, significantly improving task-solving accuracy. Subsequently, R1-Searcher [86] further optimized retrieval interactions, explicitly training models to refine search strategies through carefully designed reward functions. This allowed better exploitation of external information and improved search result relevance.

Search-R1 [47] advanced RL-based retrieval by structurally integrating sophisticated search interactions with complex reasoning processes. The method systematically bridged query generation and information reasoning, enabling nuanced responses through refined integration of retrieved content. Finally, this research line culminated in the development of Agent-R1 [71], a comprehensive DR framework integrating RL into end-to-end training of LLM agents. Agent-R1 leveraged diverse tools such as APIs, search engines, and databases, achieving autonomous multi-step task execution and dynamic tool coordination. Through RL-driven optimization across its entire pipeline, Agent-R1 demonstrated advanced capabilities in adaptive planning, iterative execution, and task refinement. Moreover, WebThinker [53] integrates a Web Explorer module for dynamic multi-hop web exploration and employs Iterative Online Direct Pre ference (DPO) Optimization to seamlessly interleave search, navigation, and report drafting during reasoning, while Pangu DeepDiver [84] builds on the 7B Pangu model pretrained on Huawei's Ascend NPUs by introducing Search Intensity Scaling (SIS) through a two-phase SFT and RL curriculum, enabling adaptive adjustment of search depth and frequency in open-web environments.

Table 3 reveals three key RL implementation patterns in DR systems: 1) Industrial systems like Gemini DR [30] and Grok DeepSearch [107] employ proprietary RL implementations with undisclosed details, 2) Academic approaches [15, 86] favor modular RL optimization using GRPO [82] and Reinforce++ [38] with transparent reward designs, and 3) Emerging hybrid systems like SimpleDeepSearcher [90] combine process-based rewards with multi-task training across 6 QA datasets. The table also highlights the prevalence of Qwen2.5 and LLaMA3 model families as preferred base architectures for RL optimization.

Reward Model and Policy Model. Most current open-source RL implementations of DR agents, including the methods discussed above, commonly adopt rule-based reward models that explicitly define task-specific objectives such as retrieval relevance, information accuracy, or successful tool invocation. To efficiently perform policy optimization, recent systems have increasingly utilized Proximal Policy Optimization (PPO) [80] and Group Relative Policy Optimization (GRPO) [82]. In particular, GRPO fundamentally reconfigures the advantage estimation paradigm by replacing traditional value functions with group-relative advantage computation. It expands reward space through intra-group normalization, sparse binary rewards are transformed into continuous advantage values spanning wider ranges. This expanded signal space provides richer gradient information for policy updates, as evidenced higher high-reward response density compared to PPO. In addition, GRPO provides a variance suppression mechanism by constraining advantage estimation within dynamically clustered response groups, such as grouping by reasoning depth or tool usage patterns, reducing policy gradient variance through local standardization. In contrast to PPO, GRPO eliminates separate value networks, removing conflicting optimization objectives between policy and value functions. Empirical measurements show GRPO reduces gradient direction conflicts from 12 to 3 per training epoch, significantly accelerating convergence. As a result, GRPO outperforms conventional PPO in wider reward distribution coverage, enhancing exploration capacity and faster KL divergence stabilization during alignment.

3.5 Non-parametric Continual Learning

DR agents depend heavily on LRMs and often utilize complex hierarchical workflows. Parameter-based learning approaches such as SFT and RL encounter significant obstacles in this context, including the need to scale model parameters, manage extensive volumes of structured experience data, and design increasingly intricate training algorithms. In contrast, non-parametric continual learning approaches offer a scalable alternative: agents refine their capabilities at runtime by optimizing external memory, workflows, and tool configurations through continuous interaction with the external environment rather than by updating internal weights. This non-parametric continual learning paradigm enables efficient online adaptation with minimal data and computational overhead, making it well-suited to DR agents with complex architectures.

Non-parametric continual learning approaches, most notably case-based reasoning (CBR), are currently a mainstream method in LLM-driven agent systems. The CBR-based method enables agents to retrieve, adapt, and reuse structured problem-solving trajectories from an external case bank dynamically. Unlike traditional RAG-based methods, which rely on static databases, CBR facilitates online contextual adaptation and effective task-level generalisation. Such flexibility underscores its potential as a scalable and practical optimization solution for DR agents with complex architecture. DS-Agent [33] is a pioneering LLM-driven agent that introduced CBR into automated data science workflows, employing approximate online retrieval from a constructed case bank. Similarly, LAM [34] applies CBR techniques to functional test generation, combining trajectory-level retrieval with LLM planning in a modular system design. Although DS-Agent itself does not include a learning phase, Agent K [31]advances this paradigm with dynamic external case retrieval and reuse guided by a reward-based memory policy, which exemplifies genuine self-evolution enabling continual adaptation and optimization without updating model parameters. Focusing on DR agents, AgentRxiv [78] further extends this paradigm by enabling autonomous research agents to collaboratively share and access a centralized repository of prior research outputs. This framework allows LLM agent laboratories to upload and retrieve reports from a shared preprint server, simulating an online-updating arXiv-like platform which can be seen as a comprehensive case bank. Such a system empowers agents to enhance their capabilities and knowledge through contextual adaptation without modifying their model parameters.

Compared to prompt-based methods, which encode fixed demonstrations or task heuristics into static input templates, Non-parametric methods enable dynamic retrieval and adaptation of structured trajectories, thereby facilitating continual task generalization without manual prompt engineering. Relative to RAG, which typically retrieves unstructured textual content from static corpora, CBR operates at the trajectory level and emphasizes reasoning-centered memory organization. A notable example is the Kaggle Grandmaster Agent [31], which demonstrates how LLMs equipped with modular reasoning components and persistent memory can achieve expert-level structured problem solving, aligning closely with the CBR paradigm. These characteristics make CBR particularly well-suited for agents requiring procedural adaptation and context-sensitive optimization across tasks. Except memory-based method, self-evolution can also arise from dynamic infrastructure adaptation. For example, Alita [75] monitors task requirements and environmental signals to provision and configure new MCP servers at runtime, seamlessly extending and refining its toolset on demand.

In summary, these self-evolution paradigms in LLM-driven DR agent systems offer substantial promise for structured reasoning and dynamic retrieval and open new pathways for efficient knowledge reuse and continual learning. Although these methods have not yet achieved widespread attention, they address the high data and computational demands inherent to parameter-based approaches and therefore represent an attractive direction for future research and practical deployment.

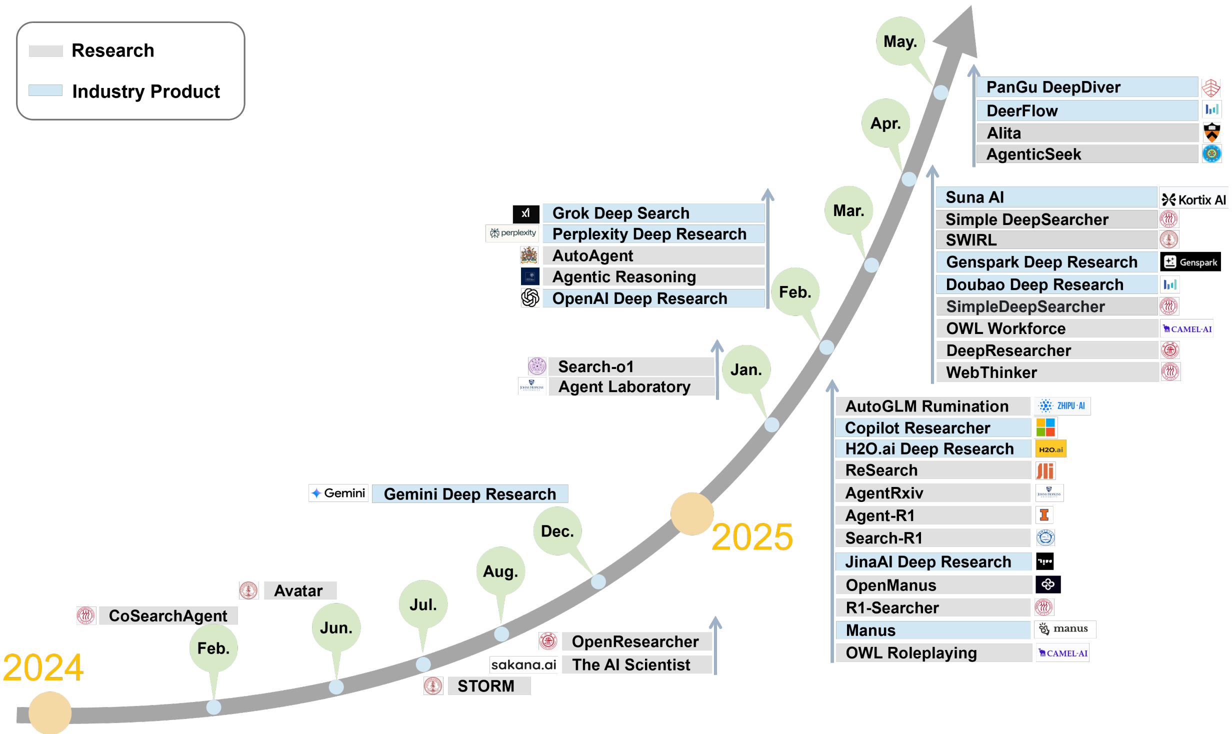


Figure 5: An overview of DR agents evolution over years.

4 Industrial Applications of Deep Research Agents

4.1 Open AI Deep Research

OpenAI recently introduced its DR capability [69], employing a single-agent architecture centered around a reinforcement learning-based, fine-tuned o3 reasoning model. Upon receiving a research query, the system initiates a concise interactive clarification step to accurately define user intent and research objectives. It then autonomously formulates and executes a sophisticated, multi-step research strategy, encompassing multimodal information retrieval, web browsing, and computational tasks such as data analysis and visualization through browser tools. Technologically, this solution delivers three significant advancements: (1) **A dynamically adaptive iterative research workflow**: Capable of refining its strategy throughout task execution. (2) **Enhanced context memory and robust multimodal processing capabilities**: Facilitating effective integration of diverse information sources. (3) **Comprehensive toolchain integration**: Combining web browsing capabilities with built-in programming tools to produce structured, authoritative reports supported by precise citations.

4.2 Gemini Deep Research

Google DeepMind recently introduced Gemini DR [30], an advanced DR agent based on its multimodal Gemini 2.0 Flash Thinking model. Gemini's reinforcement learning-driven fine-tuning, facilitated by a single-agent architecture, has been shown to enhance planning and adaptive research capabilities, enabling the system to autonomously and expeditiously complete complex tasks. Technologically, this solution delivers four significant advancements: (1) **Interactive Research Planning**: Upon receiving a research query, Gemini autonomously formulates a multi-step investigation plan for interactive user review and modification. (2) **Asynchronous Task Management**: Adopts asynchronous task management architecture to efficiently handle multiple simultaneous tasks. (3) **Large-scale context windows RAG ensembles**: Enabling effective management and coherent synthesis of multimodal data (eg. text, images)for in-depth professional research analysis. (4) **High speed adaptive retrieval**: Implements fast, multi-round adaptive web search that significantly outperforms other agents in terms of retrieval speed and amount of information per iteration.

4.3 Perplexity Deep Research

Perplexity's recently developed DR agent [72] has demonstrated an advanced capability to decompose complex queries into well-defined subtasks. The system is capable of conducting targeted web searches iteratively, critically evaluating authoritative sources, and synthesizing structured, comprehensive reports. Technologically, this solution delivers two significant advancements: (1) **Iterative Information Retrieval**: Conducts successive rounds of targeted web searches with dynamic adjustments based on interim insights, ensuring comprehensive information coverage and accuracy. (2) **Dynamic Prompt-Guided Model Selection**: Use hybrid architecture to autonomously select the optimal combination of specialized models based on the requirements and context of specific tasks, thereby enhancing adaptability and effectiveness in various research scenarios.

4.4 Grok DeepSearch

Grok DeepSearch [107], developed by xAI, is a computational framework that combines real-time information retrieval with multimodal reasoning to dynamically solve complex and information-rich problems. Technologically, this solution delivers two significant advancements: (1) **Segment-level module processing pipline**: Upon receiving a query, Grok3 initiates the credibility assessment module to identify and filter out low-quality information. Subsequently, the system's real-time data acquisition engine gathers multimodal inputs (e.g. text, images, and code) from various sources. Subsequently, employing the sparse attention mechanism, the system undertakes key reasoning subtasks, including data cleaning, cross-source verification, and multimodal integration, in a concurrent manner. Finally, the iterative optimization process culminates in the generation of structured outputs, encompassing analysis summaries, advanced visualizations (e.g. 3D trajectories), and verifiable citations. (2) **Dynamic resource allocation**: Capacity for adaptively alternating between lightweight retrieval and intensive analysis modes is noteworthy, and it is further augmented by the incorporation of a secure sandbox environment for computational verification.

4.5 Microsoft Copilot Researcher and Analyst

Microsoft recently introduced two innovative reasoning agents within Microsoft 365 Copilot: Researcher and Analyst [88]. These agents securely and compliantly access users' work data (such as emails, meeting notes, documents, and chats) as well as web information, delivering on-demand expert knowledge.

Researcher is designed to assist users in tackling complex, multi-step research tasks, delivering insights with unprecedented quality and accuracy. It combines OpenAI's advanced research models with Microsoft 365 Copilot's sophisticated orchestration and deep search capabilities. Users can employ Researcher to craft detailed market entry strategies, identify market opportunities for new products by integrating internal and external data, or prepare comprehensive quarterly reports for client reviews. Additionally, Researcher enhances its insights through connectors to third-party data sources such as Salesforce, ServiceNow, and Confluence.

Analyst is built as an advanced data analytics agent that rapidly transforms raw data into valuable insights within minutes. It leverages OpenAI's o3-mini inference model, specifically optimized for advanced analytical tasks in professional environments. Analyst uses a chain-of-thought reasoning approach, solving problems step-by-step, generating high-quality responses that closely mirror human analytical thinking.

4.6 Qwen Deep Research

Alibaba Qwen recently launched Qwen Deep Research, an advanced research agent powered by its flagship multimodal model Qwen3-235B-A22B. Through reinforcement learning-optimized task scheduling within a unified agent framework, the system demonstrates enhanced autonomous planning and adaptive execution capabilities, enabling rapid completion of complex research workflows. Key technological advancements include: (1) **Dynamic Research Blueprinting** with interactive plan refinement. (2) **Concurrent Task Orchestration** enabling parallel retrieval validation synthesis.

In addition to the pioneering DR services previously discussed, major technology corporations such as Microsoft and ByteDance, alongside emerging startups including Jina AI [3], H2O [35], and Zhipu AI [119], have also introduced their proprietary DR platforms. The advent of these solutions has spurred considerable global interest, reflected by their rapid proliferation, thereby underscoring both the technological attractiveness and substantial market potential of DR applications. Looking forward, continuous advancements in LLM reasoning, retrieval integration techniques, and multimodal generation are expected to enable DR agents to transcend traditional information retrieval and basic tool invocation tasks. Consequently, DR systems are anticipated to tackle increasingly sophisticated reasoning and complex knowledge-construction challenges, ultimately positioning DR as a foundational technological pillar for next-generation intelligent collaborative research platforms.

5 Benchmarks for DR Agent

Evaluating DR agents requires benchmarks that capture their full research workflow, including multi-step information retrieval, cross-source synthesis, dynamic tool invocation, and structured evidence-grounded report generation. Existing evaluations fall into two main categories. **Question-Answering (QA)** benchmarks range from single-turn factual queries to complex research-style problems, assessing agents' factual knowledge, domain-specific reasoning, and ability to locate and integrate relevant information. **Task Execution benchmarks** evaluate broader capabilities such as long-horizon planning, multimodal understanding, tool usage, and environment interaction by measuring how well agents carry out end-to-end research tasks. Although long-form generation datasets such as Qasper [20] and ELI5 [21] provide tests of extended output coherence, their free-form nature does not align with the structured evidence-based reporting expected of DR agents. Consequently, there is a pressing need for specialized benchmarks that reflect the multi-stage, multimodal characteristics of DR workflows and ensure rigorous and relevant assessment of agent performance across all phases of autonomous research.

Table 4: Performance of DR agents on major QA benchmarks. The best performance is highlighted in **bold**, and the second-best is indicated with an <u>underline</u>.

☐ **= Not present**

DR Agent	Base Model	QA Benchmarks					Release
		Hotpot	2Wiki	NQ	TQ	GPQA	
Search-o1 [52]	QwQ-32B-preview	57.3	**71.4**	<u>49.7</u>	<u>74.1</u>	57.9	Jan-2025
Agentic Reasoning [105]	DeepSeek-R1, Qwen2.5	☐	☐	☐	☐	67.0	Feb-2025
Grok DeepSearch [107]	Grok3	☐	☐	☐	☐	**84.6**	Feb-2025
AgentRxiv [78]	GPT-4o-mini	☐	☐	☐	☐	41.0	Mar-2025
R1-Searcher [86]	Qwen2.5-7B-Base	71.9	63.8	☐	☐	☐	Mar-2025
ReSearch [15]	Qwen2.5-7B-Base	30.0	29.7	☐	☐	☐	Mar-2025
ReSearch [15]	Qwen2.5-7B-Inst	63.6	54.2	☐	☐	☐	Mar-2025
ReSearch [15]	Qwen2.5-32B-Base	64.3	45.6	☐	☐	☐	Mar-2025
ReSearch [15]	Qwen2.5-32B-Inst	67.7	50.0	☐	☐	☐	Mar-2025
Search-R1 [47]	Llama3.2-3B-Base	30.0	29.7	43.1	61.2	☐	Mar-2025
Search-R1 [47]	Llama3.2-3B-Inst	31.4	23.3	35.7	57.8	☐	Mar-2025
Search-R1 [47]	Qwen2.5-7B-Base	28.3	27.3	39.6	58.2	☐	Mar-2025
Search-R1 [47]	Qwen2.5-7B-Inst	34.5	36.9	40.9	55.2	☐	Mar-2025
DeepResearcher [117]	Qwen2.5-7B-Inst	64.3	<u>66.6</u>	**61.9**	**85.0**	☐	Apr-2025
WebThinker [53]	QwQ-32B	☐	☐	☐	☐	<u>68.7</u>	Apr-2025
SimpleDeepSearch [90]	Qwen2.5-7B-Inst	☐	68.1	☐	☐	☐	Apr-2025
SimpleDeepSearch [90]	Qwen2.5-32B-Inst	70.5	☐	☐	☐	☐	Apr-2025
SimpleDeepSearch [90]	DeepSeek-R1-Distill-Qwen-32B	68.1	☐	☐	☐	☐	Apr-2025
SimpleDeepSearch [90]	QwQ-32B	**73.5**	☐	☐	☐	☐	Apr-2025
SWIRL [27]	Gemma-2-27B	<u>72.0</u>	☐	☐	☐	☐	Apr-2025

QA Benckmarks. QA benchmarks span a spectrum of complexity, from simple factual recall to multi-hop reasoning and research-style question answering. At the lower end, datasets such as **SimpleQA** [101], **TriviaQA** [48], and **PopQA** [59]focus on parametric or single-hop factual recall, evaluating whether models can retrieve short factual answers from memory or minimal context. **Natural Questions (NQ)** [50] and **TELEQnA** [58] add complexity by requiring answer extraction from long documents or domain-specific sources. Benchmarks like **HotpotQA** [109], **2WikiMultihopQA** [36], and **Bamboogle** [5] emphasize multi-hop reasoning and supporting evidence selection across documents. At the highest level of difficulty lies **Humanity's Last Exam (HLE)** [73], which targets expert-level, open-domain scientific questions crafted by leading professors in various fields. These questions often require multi-turn retrieval, complex inference, and even multimodal understanding. Additionally, BrowseComp [70] is another challenging benchmark proposed by OpenAI to measure the ability of AI agents to locate hard-to-find information. It retains the answer verifiability of the Simple QA benchmark while filtering out those that can be easily solved by LLMs with web search, thus testing agents' information retrieval and synthesis capabilities. Despite recent advancements, leading DR agents still exhibit suboptimal performance on the HLE and BrowserComp benchmark compared to human experts. This highlights these two benchmarks as the most critical and unresolved challenges in the evaluation of DR agents.

Table 5: Performance of DR agents on GAIA test, validation set and HLE benchmarks. The best performance is highlighted in **bold**, and the second-best is indicated with an underline.

□ = **Not present**

DR Agent	Base Model	GAIA Level-1	GAIA Level-2	GAIA Level-3	GAIA Ave.	HLE	Release
Test set							
MMAC-Copilot [87]	GPT-3.5, GPT-4	45.16	20.75	6.12	25.91	□	Mar-2024
H2O.ai DR [35]	Claude3.7-Sonnet	89.25	**79.87**	**61.22**	**79.73**	□	Mar-2025
Alita [75]	Claude-Sonnet-4, GPT-4o	**92.47**	71.7	55.1	75.42	□	May-2025
Dev set							
AutoAgent [91]	Claude-Sonnet-3.5	71.7	53.5	26.9	55.2	□	Feb-2025
OpenAI DR [69]	GPT-o3-customized	78.7	**73.2**	58.0	67.4	**26.6**	Feb-2025
Perplexity DR [72]	Flexible	□	□	□	□	21.1	Feb-2025
Manus [60]	Claude3.5, GPT-4o	86.5	70.1	57.7	71.4	□	Mar-2025
OWL [11]	Claude-3.7-Sonnet	84.9	68.6	42.3	69.7	□	Mar-2025
H2O.ai DR [35]	h2ogpt-oasst1-512-12b	67.92	67.44	42.31	63.64	□	Mar-2025
Genspark Super Agent [94]	Claude 3 Opus	**87.8**	72.7	58.8	**73.1**	□	Apr-2025
WebThinker [53]	QwQ-32B	53.8	44.2	16.7	44.7	13.0	Apr-2025
SimpleDeepSearch [90]	QwQ-32B	50.5	45.8	13.8	43.9	□	Apr-2025
Alita [75]	Claude-Sonnet-4, GPT-4o	75.15	□	**87.27**	□	□	May-2025

Table 6: Overview of nine widely used QA benchmark datasets employed in recent DR-agent studies. The first group covers single-hop QA tasks, while the second group focuses on multi-hop and multi-turn reasoning.

Benchmark	Release	Size	Task & Context	Domain	Multi-hop Nums
TriviaQA [48]	2017	95 k	Single-hop retrieval (Long web/Wiki docs)	Open	1
Natural Questions [50]	2019	307 k	Document answer extraction (Full Wikipedia article)	Open	1
PopQA [59]	2023	14 k	Single-hop parametric recall (None)	Open	1
TELEQnA [58]	2023	10 k	Domain factual QA (Telecom standards/articles)	Telecom	1
SimpleQA [101]	2024	4.3 k	Single-hop factual recall (None / parametric)	Open	1
HotpotQA [109]	2018	113 k	Multi-hop reasoning (2 Wikipedia paragraphs)	Open	2
2WikiMultihopQA [36]	2020	192 k	Multi-hop reasoning (Retrieval across Wikipedia)	Open	2+
Bamboogle [5]	2023	125	Compositional reasoning (Online search)	Open	2–3
Humanity's Last Exam [73]	2025	2.5 k	Expert-level multi-turn (Mixed external sources)	Multi-discipline	2+

Task Execution Benchmarks. Task execution benchmarks evaluate an agent's integrated capabilities in tool use, environment perception, and information filtering. These can be grouped into two subcategories. The first category comprises general-purpose assistant tasks such as GAIA [62], AssistantBench [111], and Magentic-One [24]. These tasks require agents to plan and execute tool-based workflows (for example, searching, browsing, or form filling) within environments that are open-ended and often web-based. Among them, **GAIA** has emerged as the most important benchmark, offering diverse, realistic tasks that are easily human-solvable but remain highly challenging for current agents. The second subcategory focuses on **research and code-oriented tasks**, including **SWE-bench** [46], **HumanEvalFix** [64], **MLGym** [67], **MLE-bench** [13], **MLBench** [92], **MLAgentBench** [40], and **ScienceAgentBench** [18], which test agents on completing machine learning pipelines, repairing real-world code, or replicating scientific experiments. These tasks require long-horizon planning, precise tool invocation, and often code generation and validation. Additionally, benchmarks like **RE-Bench** [104] and **RESEARCHTOWN** [112] simulate multi-agent research environments, evaluating how well agents collaborate and iterate in multi-role scientific workflows.

As DR agents continue to integrate more interactive tools, future evaluation may expand into GUI-based manipulation environments. Benchmarks such as **OSWorld** [108], **WebArena** [120], and **SpaBench** [14] allow agents to control applications or web interfaces directly, opening new avenues for testing embodied research capabilities in realistic, user-facing scenarios.

6 Challenge and Future Directions

Despite the rapid evolution of DR agents and their demonstrated efficacy in automating multi-step information discovery and synthesis, two overarching challenges persist, defining the roadmap for future innovation. First, the breadth and depth of accessible information remain tightly constrained by reliance on static knowledge repositories or conventional search interfaces. Second, the efficiency and robustness of execution workflows and system architectures are limited by linear planning paradigms and monolithic agent designs. Addressing these challenges will be critical to enabling DR agents to function as truly autonomous, adaptable research assistants capable of navigating complex, heterogeneous data landscapes and orchestrating high-throughput, parallelized reasoning processes.

Broaden Information Source. To meet the information needs of complex tasks, current DR agents adopt static knowledge bases (such as the RAG method) or rely exclusively on search engines and browsers; the former approach is insufficiency, while the latter is confined to publicly available web content, thereby significantly constraining their information-acquisition capabilities. This inherent limitation renders them incapable of retrieving information concealed behind applications, proprietary interfaces or specialised databases. For example, conventional browsing and search techniques cannot penetrate enterprise software, mobile applications, or subscription-only services, such as the Bloomberg Terminal, thereby precluding access to critical, real-time market intelligence. In order to surmount this limitation, it is imperative to integrate a more granular and extensive range of modular tools via MCPs. This approach enables agents to dynamically access specialised tools and resources beyond the scope of standard browsers or search engines. Such resources may include proprietary applications, databases, or APIs, thereby facilitating the retrieval of previously inaccessible data. Consequently, DR agents have the capacity to deliver more precise, adaptive, and context-aware interactions, thereby effectively fulfilling diverse and complex user requirements.

Following the integration of proprietary APIs and databases, the rate-limiting factor in the workflow shifts from data acquisition to webpage interaction efficiency. Conventional human-centred browsers create a further bottleneck for agents. Because they optimise for visual rendering rather than programmatic control, they suffer from sluggish page loads, fragile element locators that shift with every layout change, and aggressive anti-bot defences that often break automated sessions. These shortcomings translate into high latency, unstable scraping and limited parallelism whenever DR agents try to harvest data at scale. To address this bottleneck, researchers have begun to design **AI-native browsers** such as Browserbase [2], Browser Use [65], Dia, Fellou [93], and the Comet [95] from Perplexity. expose a stable, structured DOM view that agents can traverse programmatically [2, 65, 95]. [2, 93] supply explicit API hooks for clicking elements and filling forms, which removes the need for brittle coordinate-based actions. [2] further executes pages asynchronously in a headless container, reducing load-time variance and avoiding the overhead of a visible interface. [2] embeds a vision–language model that tracks dynamic page changes and automatically resolves log-in gates and anti-bot challenges. [65, 95] coordinates dozens of tabs in parallel, allowing DR agents to interact with private dashboards, single-page applications, and interactive visualisations at scale. In combination, these capabilities eliminate the delays and fragility that arise when conventional, human-centred browsers sit between the agent and newly unlocked proprietary data sources.

Fact Checking. To further boost factual accuracy, the latest methods add a structured verification loop and self-reflection abilities on top of multi-step retrieval. Concretely, once an agent has drafted a preliminary answer, it does not rush to deliver a verdict. Instead, it proactively launches cross-checks: it looks for independent sources that confirm the same fact and searches for evidence of contradictions. Grok DeepSearch, for example, follows this strategy—it rates the credibility of every source, inspects consistency through as many as seven layers of depth, and verifies each key claim across multiple origins [107]. This multi-source cross-validation sharply reduces single-source errors and raises answer reliability. At the same time, agents have begun to reflect on their own reasoning. During inference, they inspect and test intermediate results, much like a human researcher's reflective thinking. Zhipu's Rumination model [119], for instance, pauses after concluding, keeps searching to check whether that conclusion holds, and only then finalizes the answer. Such introspection is typically encouraged by adding correctness-oriented rewards in reinforcement learning. If the model detects conflict or uncertainty, it replans its retrieval strategy and, when necessary, backtracks to revise earlier inferences [69]. Through this blend of structured verification and self-reflection, research agents now attain an unprecedented level of rigor in fact-checking: they not only supply an answer but also explain why it is trustworthy, dramatically lowering factual errors and hallucinations. In short, modern agents can lay out a search plan, adapt queries as intermediate evidence comes in, and—where needed—rewind prior steps to recover missing information [69].

Asynchronous Parallel Execution. To address the limitation that most existing DR agents rely exclusively on linear task planning, i.e. the sequential execution of subtasks, we introduce two possible methodologies. These methods overcome the inherent efficiency and robustness constraints of purely linear strategies and enable both the exploitation of parallelism and the implementation of dynamic adjustments during task execution. Firstly, an asynchronous, parallel architecture leveraging advanced task-modeling structures, such as directed acyclic graphs (DAGs), presents a promising future direction which could enable parallel execution and dynamic prioritisation of subtasks, effectively managing complex interdependencies among tasks and facilitating potentially sophisticated planning capabilities such as replanning. Secondly, a learned scheduling agent, trained via reinforcement learning to allocate subtasks and adjust execution order based on runtime performance signals (e.g. execution latency), could be proposed. By treating scheduling decisions as actions in an RL environment, the agent progressively discovers policies that balance parallelism, resource utilisation, and task criticality, yielding more robust and efficient end-to-end research workflows.

Tool-Integrated Reasoning. A fundamental challenge in developing effective DR agents lies in the implementation of Tool-Integrated Reasoning (TIR), a paradigm that extends beyond simple tool usage to encompass complex, multi-step reasoning with dynamic tool integration. TIR requires agents to not only invoke appropriate tools in logical sequence but also to adaptively adjust their reasoning pathways based on intermediate results. Traditional supervised fine-tuning approaches have demonstrated limited generalization capabilities in tool-based reasoning tasks, often leading to over-reasoning or inappropriate tool selection. Recent research by [74] has shown that reinforcement learning frameworks with carefully designed reward structures can significantly enhance models' tool reasoning abilities. By incorporating fine-grained rewards that evaluate not only final answer correctness but also tool selection appropriateness, parameter specification accuracy, and reasoning efficiency, TIR-optimized agents have demonstrated performance improvements of 15-17% across multiple benchmarks. Furthermore, these agents exhibit superior generalization to unseen tools and tasks, more rational invocation patterns, and better balance between tool utilization and self-knowledge. Implementing TIR effectively within DR agents represents a critical step toward achieving truly autonomous research assistants capable of navigating complex information landscapes with minimal human intervention.

Benchmark Misalignment. Most public DR evaluations remain anchored in traditional QA suites whose items are harvested chiefly from static corpora such as Wikipedia. Since a considerable amount of this content is now embedded in backbone model parameters, current competitive agents can often answer directly from memory, bypassing any research procedure and thus inflating their performance. To probe genuine capabilities of retrieval, reasoning and tool usage, the field of DR urgently needs open-web, time-sensitive benchmarks. From this perspective, BrowseComp [70] constitutes a meaningful step forward by filtering out questions solvable with parametric knowledge and forcing agents to locate hard-to-find information online. Besides, a complementary direction is a continually refreshed leaderboard that updates problems from the latest web environment and events, deterring benchmark hacking through parametric memorisation.

Beyond parametric knowledge hacking of QA benchmark, the metrics of the most existing DR research still collapse open-ended research workflows into narrowly scoped QA prompts or rudimentary GUI-control tasks, overlooking the paradigm's defining outcome, a structured, multi-modal research report that weaves together textual narrative, tables, figures, and citations. Since the metrics of these benchmarks centre almost exclusively on information retrieval and extraction and tool invocation, they under-assess higher-level competencies such as evidence aggregation across heterogeneous sources, cross-modal synthesis, and discourse-level organization. Thus, a key research priority is the development of comprehensive benchmarks that evaluate DR agents' capacity for end-to-end report generation, encompassing long-form narrative, integrated tables and figures, and multimodal coherence, thereby assessing factual accuracy, discourse structure, and cross-modal alignment within a single task.

Parametric Optimization of Multi-Agent Architectures. End-to-end RL has been demonstrated by OpenAI [69, 71] to significantly enhance the reasoning capabilities of backbone models for DR tasks, a result successfully replicated by several open-source initiatives. However, current implementations predominantly utilize single-agent architectures, requiring the backbone model to simultaneously manage planning, tool invocation, and report generation. This multitasking places excessive computational and cognitive demands on backbone models, thereby reducing their efficiency and robustness. Distributing workloads across multiple specialized agents has shown promising improvements in system performance [96], yet achieving effective end-to-end training and efficient coordination among multiple agents remains a critical open challenge.

To optimize multi-agent architectures for DR tasks, we propose two promising future directions: (i) adopting **hierarchical reinforcement learning** (HRL), which introduces layered internal reward mechanisms that facilitate efficient feedback propagation and foster cooperative learning among agents; or implementing a post-training optimization pipeline consisting of multiple refinement stages specifically tailored for DR tasks, which could iteratively enhance inter-agent interactions and thus improve overall system stability and adaptability; and (ii) employing an **RL-based dedicated scheduling agent designed to dynamically allocate subtasks and adjust execution order based on real-time performance metrics**. By modeling scheduling decisions as actions within an RL framework, this method progressively learns adaptive policies that optimally balance parallel execution, resource utilization, and task prioritization, enhancing both the robustness and efficiency of end-to-end research workflows.

Self-Evolving Language Model Agents. Although initial attempts at self-evolution methods for DR agents have emerged, exemplified by simulated collaborative platforms such as AgentRxiv [78] that facilitate online sharing and reuse of structured research experiences, the paradigm remains underdeveloped and narrowly focused on only the case-based reasoning paradigm. Similarly, CycleResearcher [103] enables the entire research process simulation (research-evaluation-refine) through iterative preference learning with a robust verifier [121], representing a significant step toward fully automated scientific inquiry and sharing the similar self-evolution concept with AlphaEvolve [68].

To fully realize the potential of self-evolution in DR agents, future research should expand the self-evolution method along two complementary directions. (i) **Comprehensive case-based reasoning framework**. Case-based reasoning approaches [1] leverage hierarchical experience traces, including planning trajectories and structured tool invocation logs, and employ advanced retrieval and selection mechanisms to enable fine-grained, context-specific adaptation. (ii) **Autonomous workflow evolution** promises enhanced efficiency and flexibility. By representing agent workflows as mutable structures such as trees or graphs, researchers can apply evolutionary algorithms or adaptive graph optimization to explore, modify and refine execution plans dynamically. Pursuing both directions in tandem will strengthen the robustness of frameworks and reduce the reliance on data and computation resources.

7 Conclusion

LLM-driven Deep Research Agents represent an emerging paradigm for automated research support, integrating advanced techniques such as iterative information retrieval, long-form content generation, autonomous planning, and sophisticated tool utilization. In this survey, we systematically reviewed recent advancements in DR agents, categorizing existing methodologies into prompt-based, fine-tuning-based, and reinforcement learning-based approaches from the perspectives of information retrieval and report generation. Non-parametric methods utilize LLMs and carefully designed prompts to achieve efficient and cost-effective deployment, making them suitable for rapid prototyping. In contrast, fine-tuning and reinforcement learning approaches explicitly optimize model parameters, significantly enhancing the agents' reasoning and decision-making capabilities. We also examined prominent DR agent systems developed by industry leaders and discussed their technical implementations, strengths, and limitations.

Limitation

Despite notable progress, key challenges remain, including limited generalization across diverse tasks, inflexible task workflows, difficulty in integrating granular external tools, and substantial computational complexity associated with advanced planning and optimization. Future research directions thus emphasize broader and more flexible tool integration through modular capability providers (e.g., Operator-based architectures), development of asynchronous and parallel planning frameworks (e.g., Directed Acyclic Graph-based approaches), and sophisticated end-to-end optimization methods for multi-agent architectures, such as hierarchical reinforcement learning or multi-stage fine-tuning pipelines. With continued advancements in LLM technologies, DR agents have significant potential to transform complex research workflows, enhance human productivity, and drive innovation across academic and industrial domains.

References

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[2] McQuilkin Adam, Kamath Anirudh, McGuire Sean, and Chance Sophie. Browserbase: A web browser for your ai, 2024.

[3] Jina AI. Jina ai. `https://jina.ai/deepsearch/`, 2025. Accessed: 2025-04-28.

[4] Kortix AI. Suna: Open source generalist ai agent. `https://github.com/kortix-ai/suna`, 2025. Accessed: 2025-05-28.

[5] Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, Manzil Zaheer, Felix Yu, and Sanjiv Kumar. Rest meets react: Self-improvement for multi-step reasoning llm agent, 2023.

[6] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. Trec ikat 2023: The interactive knowledge assistance track overview. *arXiv preprint arXiv:2401.01330*, 2024.

[7] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

[8] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models. *arXiv preprint arXiv:2401.18058*, 2024.

[9] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[11] CAMEL-AI.org. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. `https://github.com/camel-ai/owl`, 2025. Accessed: 2025-03-07.

[12] Hyungjoo Chae, Yongho Song, Kai Tzu-iunn Ong, Taeyoon Kwon, Minjin Kim, Youngjae Yu, Dongha Lee, Dongyeop Kang, and Jinyoung Yeo. Dialogue chain-of-thought distillation for commonsense-aware conversational agents. *arXiv preprint arXiv:2310.09343*, 2023.

[13] Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. Mle-bench: Evaluating machine learning agents on machine learning engineering, 2025.

[14] Jingxuan Chen, Derek Yuen, Bin Xie, Yuhao Yang, Gongwei Chen, Zhihao Wu, Li Yixing, Xurui Zhou, Weiwen Liu, Shuai Wang, Kaiwen Zhou, Rui Shao, Liqiang Nie, Yasheng Wang, Jianye Hao, Jun Wang, and Kun Shao. Spa-bench: A comprehensive benchmark for smartphone agent evaluation. *arXiv preprint arXiv:2410.15164*, 2025.

[15] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Fan Yang, Zenan Zhou, Weipeng Chen, Haofen Wang, Jeff Z Pan, et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025.

[16] Yiqun Chen, Lingyong Yan, Weiwei Sun, Xinyu Ma, Yi Zhang, Shuaiqiang Wang, Dawei Yin, Yiming Yang, and Jiaxin Mao. Improving retrieval-augmented generation through multi-agent reinforcement learning. *arXiv preprint arXiv:2501.15228*, 2025.

[17] Zhixun Chen, Ming Li, Yuxuan Huang, Yali Du, Meng Fang, and Tianyi Zhou. Atlas: Agent tuning via learning critical steps. *arXiv preprint arXiv:2503.02197*, 2025.

[18] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2025.

[19] DanielWalnut. Deerflow. `https://github.com/bytedance/deer-flow`, 2025. Accessed: 2025-05-28.

[20] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011*, 2021.

[21] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*, 2019.

[22] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501, 2024.

[23] Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. Retrieval-generation synergy augmented large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11661–11665. IEEE, 2024.

[24] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

[25] Tong Fu, Liquan Chen, Zhangjie Fu, Kunliang Yu, and Yu Wang. Ccnet: Cnn model with channel attention and convolutional pooling mechanism for spatial image steganalysis. *Journal of Visual Communication and Image Representation*, 88:103633, 2022.

[26] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2, 2023.

[27] Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D Manning. Synthetic data generation & multi-step rl for reasoning & tool use. *arXiv preprint arXiv:2504.04736*, 2025.

[28] Peiyuan Gong, Jiamian Li, and Jiaxin Mao. Cosearchagent: a lightweight collaborative search agent with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2729–2733, 2024.

[29] Google. Announcing the Agent2Agent Protocol (A2A), 2025. Accessed: 2025-04-22.

[30] Google Team. Introducing gemini deep research. `https://gemini.google/overview/deep-research/`, 2025. Accessed: 2025-04-06.

[31] Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, Hamza Cherkaoui, Youssef Attia El-Hili, Kun Shao, Jianye Hao, Jun Yao, Balazs Kegl, Haitham Bou-Ammar, and Jun Wang. Large language models orchestrating structured reasoning achieve kaggle grandmaster level, 2024.

[32] Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. Deeprag: Thinking to retrieval step by step for large language models. *arXiv preprint arXiv:2502.01142*, 2025.

[33] Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning, 2024.

[34] Siyuan Guo, Huiwu Liu, Xiaolong Chen, Yuming Xie, Liang Zhang, Tao Han, Hechang Chen, Yi Chang, and Jun Wang. Optimizing case-based reasoning system for functional test script generation with large language models. *arXiv preprint arXiv:2503.20576*, 2025.

[35] H2O.ai. H2o.ai, 2025. Accessed: 2025-04-28.

[36] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

[37] Sheryl Hsu, Omar Khattab, Chelsea Finn, and Archit Sharma. Grounding by trying: Llms with reinforcement learning-enhanced retrieval. *arXiv preprint arXiv:2410.23214*, 2024.

[38] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

[39] Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Julia Hockenmaier, and Tong Zhang. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *arXiv preprint arXiv:2503.12759*, 2025.

[40] Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2024.

[41] Tatsuro Inaba, Hirokazu Kiyomaru, Fei Cheng, and Sadao Kurohashi. Multitool-cot: Gpt-3 can use multiple external tools with chain of thought prompting. *arXiv preprint arXiv:2305.16896*, 2023.

[42] Shayekh Bin Islam, Md Asib Rahman, KSM Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez. Open-rag: Enhanced retrieval-augmented reasoning with open-source large language models. *arXiv preprint arXiv:2410.01782*, 2024.

[43] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

[44] Pengcheng Jiang. Deepretrieval: Powerful query generation for information retrieval with reinforcement learning. *arXiv preprint arXiv:2503.00223*, 2025.

[45] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, 2023.

[46] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2024.

[47] Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[48] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[49] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, 2023.

[50] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[51] Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. Start: Self-taught reasoner with tools. *arXiv preprint arXiv:2503.04625*, 2025.

[52] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025.

[53] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. `https://www.notion.so/WebThinker-Empowering-Large-Reasoning-Models-with-Deep-Research-Capability-d13158a27d924a4b9df7f9ab94066b64`, 2025. Notion Blog.

[54] Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, and Sirui Hong. Openmanus: An open-source framework for building general ai agents. `https://github.com/mannaandpoem/OpenManus`, 2025. Accessed: 2025-04-06.

[55] Jingyang Lin, Andy Wong, Tian Xia, Shenghua He, Hui Wei, Mei Han, and Jiebo Luo. Facilitating long context understanding via supervised chain-of-thought reasoning. *arXiv preprint arXiv:2502.13127*, 2025.

[56] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

[57] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[58] Ali Maatouk, Fadhel Ayed, Nicola Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. Teleqna: A benchmark dataset to assess large language models telecommunications knowledge, 2023.

[59] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023.

[60] Manus AI. Leave it to manus. `https://manus.im/`, 2025. Accessed: 2025-04-06.

[61] Martin. Agenticseek: Private, local manus alternative. `https://github.com/Fosowl/agenticSeek`, 2025. Accessed: 2025-05-28.

[62] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

[63] Microsoft. Introducing researcher and analyst in microsoft 365 copilot. `https://www.microsoft.com/en-us/microsoft-365/blog/2025/03/25/introducing-researcher-and-analyst-in-microsoft-365-copilot/`, March 2025. Accessed: 2025-04-28.

[64] Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. Octopack: Instruction tuning code large language models. *arXiv preprint arXiv:2308.07124*, 2024.

[65] Magnus Müller and Gregor Žunič. Browser use: Enable ai to control your browser, 2024.

[66] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[67] Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*, 2025.

[68] Alexander Novikov, Ngân Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *Google DeepMind*, 2025.

[69] OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, 2025. Accessed: 2025-04-06.

[70] OpenAI Team. Browsecomp: a benchmark for browsing agents. https://openai.com/index/browsecomp/, 2025. Accessed: 2025-04-29.

[71] Jie Ouyang, Ruiran Yan, Yucong Luo, Mingyue Cheng, Qi Liu, Zirui Liu, Shuo Yu, and Daoyu Wang. Training powerful llm agents with end-to-end reinforcement learning. https://github.com/0russwest0/Agent-R1, 2025. Accessed: 2025-04-06.

[72] Perplexity Team. Introducing perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research, 2025. Accessed: 2025-04-06.

[73] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[74] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs, 2025.

[75] Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

[76] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8):198343, 2025.

[77] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023.

[78] Samuel Schmidgall and Michael Moor. Agentrxiv: Towards collaborative autonomous research. *arXiv preprint arXiv:2503.18102*, 2025.

[79] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.

[80] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[81] Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. Assisting in writing wikipedia-like articles from scratch with large language models. *arXiv preprint arXiv:2402.14207*, 2024.

[82] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[83] Zhili Shen, Chenxin Diao, Pavlos Vougiouklis, Pascual Merita, Shriram Piramanayagam, Damien Graux, Dandan Tu, Zeren Jiang, Ruofei Lai, Yang Ren, et al. Gear: Graph-enhanced agent for retrieval-augmented generation. *arXiv preprint arXiv:2412.18431*, 2024.

[84] Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lifeng Shang, Fisher Yu, et al. Pangu deepdiver: Adaptive search intensity scaling via open-web reinforcement learning. *arXiv preprint arXiv:2505.24332*, 2025.

[85] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.

[86] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

[87] Zirui Song, Yaohang Li, Meng Fang, Zhenhao Chen, Zecheng Shi, Yuan Huang, and Ling Chen. Mmac-copilot: Multi-modal agent collaboration operating system copilot. *arXiv preprint arXiv:2404.18074*, 2024.

[88] Jared Spataro. Introducing Researcher and Analyst in Microsoft 365 Copilot, March 2025.

[89] Md Arafat Sultan, Jatin Ganhotra, and Ramón Fernandez Astudillo. Structured chain-of-thought prompting for few-shot generation of content-grounded qa conversations. *arXiv preprint arXiv:2402.11770*, 2024.

[90] Shuang Sun*, Huatong Song*, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen Wayne Xin Zhao. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. 2025.

[91] Jiabin Tang, Tianyu Fan, and Chao Huang. Autoagent: A fully-automated and zero-code framework for llm agents, 2025.

[92] Xiangru Tang, Yuliang Liu, Zefan Cai, Yanjun Shao, Junjie Lu, Yichi Zhang, Zexuan Deng, Helan Hu, Kaikai An, Ruijun Huang, Shuzheng Si, Sheng Chen, Haozhe Zhao, Liang Chen, Yan Wang, Tianyu Liu, Zhiwei Jiang, Baobao Chang, Yin Fang, Yujia Qin, Wangchunshu Zhou, Yilun Zhao, Arman Cohan, and Mark Gerstein. Ml-bench: Evaluating large language models and agents for machine learning tasks on repository-level code. *arXiv preprint arXiv:2311.09835*, 2024.

[93] Fellou AI Team. Fellou: The world's first agentic browser, 2025.

[94] Genspark Team. Genspark super agent with enhancements in mixture of agents, 2025.

[95] Perplexity Team. Comet: A browser for agentic search by perplexity, 2024.

[96] Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *arXiv preprint arXiv:2406.01014*, 2024.

[97] Keheng Wang, Feiyu Duan, Peiguang Li, Sirui Wang, and Xunliang Cai. Llms know what they need: Leveraging a missing information guided framework to empower retrieval-augmented generation. *arXiv preprint arXiv:2404.14043*, 2024.

[98] Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-retrieval augmented generation. *arXiv preprint arXiv:2501.14342*, 2025.

[99] Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv preprint arXiv:2402.02244*, 2024.

[100] Yaoxiang Wang, Zhiyong Wu, Junfeng Yao, and Jinsong Su. Tdag: A multi-agent framework based on dynamic task decomposition and agent generation. *Neural Networks*, page 107200, 2025.

[101] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

[102] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[103] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*, 2024.

[104] Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. Re-bench: Evaluating frontier ai r&d capabilities of language model agents against human experts. *arXiv preprint arXiv:2411.15114*, 2024.

[105] Junde Wu, Jiayuan Zhu, and Yuyuan Liu. Agentic reasoning: Reasoning llms with tools for the deep research. *arXiv preprint arXiv:2502.04644*, 2025.

[106] Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis N. Ioannidis, Karthik Subbian, Jure Leskovec, and James Zou. Avatar: Optimizing llm agents for tool usage via contrastive reasoning. *arXiv preprint arXiv:2406.11200*, 2024.

[107] xAI Team. Introducing grok deepsearch. `https://x.ai/news/grok-3`, 2025. Accessed: 2025-04-06.

[108] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

[109] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[110] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023.

[111] Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

[112] Haofei Yu, Zhaochen Hong, Zirui Cheng, Kunlun Zhu, Keyang Xuan, Jinwei Yao, Tao Feng, and Jiaxuan You. Researchtown: Simulator of human research community. *arXiv preprint arXiv:2412.17767*, 2024.

[113] Tian Yu, Shaolei Zhang, and Yang Feng. Auto-rag: Autonomous retrieval-augmented generation for large language models. *arXiv preprint arXiv:2411.19443*, 2024.

[114] Saber Zerhoudi and Michael Granitzer. Personarag: Enhancing retrieval-augmented generation systems with user-centric agents. *arXiv preprint arXiv:2407.09394*, 2024.

[115] Liang Zhang, Katherine Jijo, Spurthi Setty, Eden Chung, Fatima Javid, Natan Vidra, and Tommy Clifford. Enhancing large language model performance to answer questions and extract information more accurately. *arXiv preprint arXiv:2402.01722*, 2024.

[116] Zhebin Zhang, Xinyu Zhang, Yuanhang Ren, Saijiang Shi, Meng Han, Yongkang Wu, Ruofei Lai, and Zhao Cao. Iag: Induction-augmented generation framework for answering reasoning questions. *arXiv preprint arXiv:2311.18397*, 2023.

[117] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.

[118] Yuxiang Zheng, Shichao Sun, Lin Qiu, Dongyu Ru, Cheng Jiayang, Xuefeng Li, Jifan Lin, Binjie Wang, Yun Luo, Renjie Pan, et al. Openresearcher: Unleashing ai for accelerated scientific research. *arXiv preprint arXiv:2408.06941*, 2024.

[119] Zhipu AI. Autoglm rumination. `https://autoglm-research.zhipuai.cn/`, 2025. Accessed: 2025-04-06.

[120] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2024.

[121] Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. Deepreview: Improving llm-based paper review with human-like deep thinking process. *arXiv preprint arXiv:2503.08569*, 2025.