Article

Not peer-reviewed version

Deep Research: A Systematic Survey

Zhengliang Shi , Yiqun Chen , Haitao Li , Weiwei Sun , Shiyu Ni , Yougang Lyu , Run-Ze Fan , Bowen Jin , Yixuan Weng , Minjun Zhu , Qiujie Xie , Xinyu Guo , Qu Yang , Jiayi Wu , Jujia Zhao , Xiaqiang Tang , Xinbei Ma , Cunxiang Wang , Jiaxin Mao , Qingyao Ai , Jen-Tse Huang , Wenxuan Wang , Yue Zhang , Yiming Yang , Zhaopeng Tu *, Zhaochun Ren *

Posted Date: 26 November 2025

doi: 10.20944/preprints202511.2077.v1

Keywords: deep research; large language models; information retrieval

 doi:10.20944/preprints202511.2077.v1

Article

Deep Research: A Systematic Survey

Zhengliang Shi [1], Yiqun Chen [2], Haitao Li [3], Weiwei Sun [4], Shiyu Ni [5], Yougang Lyu [6], Run-Ze Fan [7], Bowen Jin [8], Yixuan Weng [9], Minjun Zhu [9], Qiujie Xie [9], Xinyu Guo [10], Qu Yang [11], Jiayi Wu [11], Jujia Zhao [12], Xiaqiang Tang [11], Xinbei Ma [11], Cunxiang Wang [3], Jiaxin Mao [2], Qingyao Ai [3], Jen-Tse Huang [13], Wenxuan Wang [2], Yue Zhang [9], Yiming Yang [4], Zhaopeng Tu [11,*] and Zhaochun Ren [12,*]

[1] Shandong University
[2] Renmin University of China
[3] Tsinghua University
[4] Carnegie Mellon University
[5] UCAS
[6] University of Amsterdam
[7] University of Massachusetts Amherst
[8] University of Illinois Urbana-Champaign
[9] Westlake University
[10] University of Arizona
[11] Tencent
[12] Leiden University
[13] Johns Hopkins University
* Correspondence: tuzhaopeng@gmail.com (Z.T.); z.ren@liacs.leidenuniv.nl (Z.R.)

Abstract

Large language models (LLMs) have rapidly evolved from text generators into powerful problem solvers. Yet, many open tasks demand critical thinking, multi-source, and verifiable outputs, which are beyond single-shot prompting or standard retrieval-augmented generation. Recently, numerous studies have explored *Deep Research* (DR), which aims to combine the reasoning capabilities of LLMs with external tools, such as search engines, thereby empowering LLMs to act as research agents capable of completing complex, open-ended tasks. This survey presents a comprehensive and systematic overview of deep research systems, including a clear roadmap, foundational components, practical implementation techniques, important challenges, and future directions. Specifically, our main contributions are as follows: (i) we formalize a three-stage roadmap and distinguish deep research from related paradigms; (ii) we introduce four key components: query planning, information acquisition, memory management, and answer generation, each paired with fine-grained sub-taxonomies; (iii) we summarize optimization techniques, including prompting, supervised fine-tuning, and agentic reinforcement learning; and (iv) we consolidate evaluation criteria and open challenges, aiming to guide and facilitate future development. *As the field of deep research continues to evolve rapidly, we are committed to continuously updating this survey to reflect the latest progress in this area.*

Keywords: deep research; large language models; information retrieval

Contents

 doi:10.20944/preprints202511.2077.v1

 doi:10.20944/preprints202511.2077.v1

1. Introduction

Large language models (LLMs), trained on web-scale corpora, have rapidly evolved from fluent text generators into autonomous agents capable of long-horizon reasoning in practical complex applications [1–4]. They have exhibited strong generalization across diverse domains, including mathematical reasoning [5,6], creative writing [7], and practical software engineering [8–10]. Many real-world tasks are inherently open-ended, involving **critical thinking**, **factually grounded information**, and the production of **self-contained** responses. This is far beyond what single-shot prompting or static parametric knowledge can provide [11–13]. To address this gap, the **Deep Research (DR)** paradigm [14–19] has emerged. DR frames LLMs within an end-to-end research workflow that iteratively decomposes complex problems, acquire evidence via tool use, and synthesizes validated insights into coherent long-form answers.

Despite rapid progress, there remains no comprehensive survey that systematically analyzes the key components, technical details, and open challenges of DR. Most existing work [20,21] mainly summarizes developments in related areas such as Retrieval-Augmented Generation (RAG) and web-based agents [22–26]. However, in contrast to RAG [27,28], DR adopts a more flexible, autonomous workflow that eschews handcrafted pipelines and aims to produce coherent, evidence-grounded reports. Therefore, a clear overview of its technical landscape is urgent but remains a challenge. This survey fills this gap by providing a comprehensive synthesis of DR: mapping its core components to representative system implementations, consolidating key techniques and evaluation methodologies, and establishing a foundation for consistent benchmarking and sustained progress in AI-driven research.

In this survey, we propose a three-stage roadmap for DR systems, illustrating their broad applications ranging from agentic information seeking to autonomous scientific discovery. Based on the roadmap, we summarize the key components of the task-solving workflow for the most commonly used DR systems. Specifically, we present four foundational components in DR: (i) *query planning*, which decomposes the initially input query into a series of simpler, sub-queries [29,30]; (ii) *information acquisition*, which invokes external retrieval, web browsing, or various tools on demand [31,32]; (iii) *memory management*, which ensures relevant task-solving context through controlled updating or folding [33]; (iv) *answer generation*, which produces comprehensive outputs with explicit source attribution, *e.g.*, a scientific report. This scope is distinct from standard RAG [27,28] techniques, which typically treat retrieval as a heuristic augmentation step, without a flexible research workflow or a broader action space. We also introduce how to optimize DR systems in effectively coordinating these

 doi:10.20944/preprints202511.2077.v1

components, categorizing existing approaches into three types: (i) *workflow prompting*; (ii) *supervised fine-tuning* (SFT), and (iii) *end-to-end reinforcement learning* (RL).

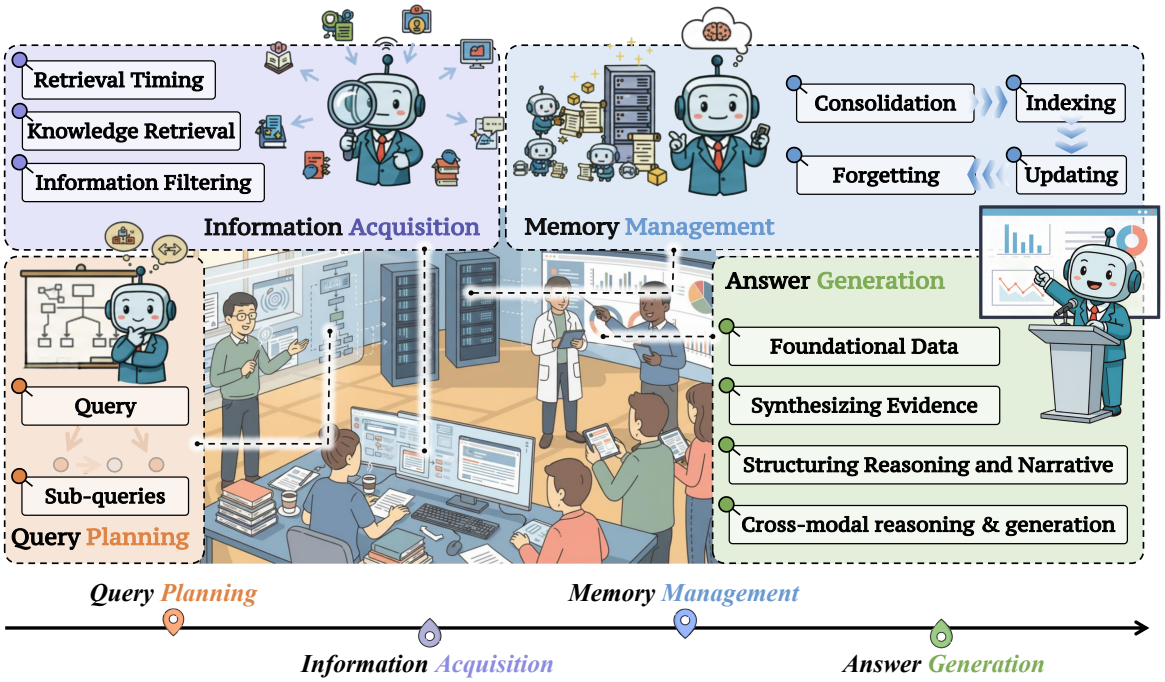

Figure 1. An overview of four key components in a general deep research system, including: Task Planning (Section 3.1). Information Acquisition (Section 3.2). Memory Management (Section 3.3) and Answer Generation (Section 3.4).

The remainder of this paper is organized as follows: Section 2 clearly defines DR and its boundaries; Section 3 introduces four key components in DR; Section 4 introduces technique details about optimizing a DR system; Section 5 summarizes well-known evaluation datasets and resources, and Section 6 discusses challenges for future directions.

To sum up, our survey makes the following contributions: (i) We formalize a three-stage roadmap of DR and clearly distinguish it from related techniques such as standard retrieval-augmented generation; (ii) We introduce four key components of DR systems, together with fine-grained sub-taxonomies for each, to provide a comprehensive view of the research loop; (iii) We summarize detailed optimization approaches for building DR systems, offering practical insights into workflow prompting, supervised fine-tuning, and reinforcement learning; and (iv) We consolidate evaluation criteria and open challenges to enable comparable reporting and to guide future research.

2. Preliminary Concept of Deep Research

2.1. What is Deep Research

DR aims to endow LLMs with an **end-to-end research workflow**, enabling them to function as agents that generate coherent, source-grounded reports with minimal human supervision. Such systems automate the entire research loop, spanning planning, evidence acquisition, analysis, and reporting. In a DR setting, the LLM agent plans queries, acquires and filters evidence from heterogeneous sources (*e.g.*, the web, tools, and local files), maintains and revises a working memory, and synthesizes verifiable answers with explicit attribution. Below, we formally introduce a three-phase roadmap that structures the rapidly evolving, capability-oriented landscape of DR, and we compare it systematically with conventional RAG paradigms.

 doi:10.20944/preprints202511.2077.v1

2.2. Understanding Deep Research from Three Phases

We view DR as a capability trajectory rather than a value hierarchy. The three phases below capture a progressive expansion of what systems can reliably do, from acquiring precise evidence, to synthesizing it into readable analyses, and finally to forming defensible insights.

Phase I: Agentic Search. Phase I systems specialize in finding the correct sources and extracting answers with minimal synthesis. They typically reformulate the user query (via rewriting or decomposition) to improve recall, retrieve and re-rank candidate documents, apply lightweight filtering or compression, and produce concise answers supported by explicit citations. The emphasis is on faithfulness to retrieved content and predictable runtime. Representative applications include open-domain question answering [168,220], multi-hop question answering [171,175,177], and other information-seeking tasks [178,221–224] where truth is localized to a small set of sources. Evaluation prioritizes retrieval recall@k and answer exact matching, complemented by citation correctness and end-to-end latency, reflecting the phase's focus on accuracy-per-token and operational efficiency.

Phase II: Integrated Research. Phase II systems move beyond isolated facts to produce coherent, structured reports that integrate heterogeneous evidence while managing conflicts and uncertainty. The research loop becomes explicitly iterative: systems plan sub-questions, retrieve and extract key evidence from various raw content (*e.g.*, HTML [92], tables [225,226], and charts [227,227]), and ultimately synthesize comprehensive, narrative reports. The most commonly-used applications include market and competitive analysis [228,229], policy briefs [230], itinerary design under constraints [231], and other long-horizon question answering [16,181,184,186]. Accordingly, evaluation shifts from superficial short-form lexical matching to long-form quality, including: fine-grained factuality [232,233], verified citations [234,235], structural coherence [236], key points coverage [237]. Phase II thus trades a modest increase in compute and complexity for substantial gains in clarity, coverage, and decision support.

Phase III: Full-stack AI Scientist. Phase III aims at advancing scientific understanding and creation beyond mere information aggregation, representing a broader and more ambitious stage of DR In this phase, DR agents are expected not only to aggregate evidence but also to generate hypotheses [238], conduct experimental validation or ablation studies [239], critique existing claims [219], and propose novel perspectives [212]. Common applications include paper reviewing [219,240,241], scientific discovery [242–244], and experiment automation [245,246]. Evaluation at this stage emphasizes the novelty and insightfulness of the findings, the argumentative coherence, the reproducibility of claims (including the ability to re-derive results from cited sources or code), and calibrated uncertainty disclosure.

 doi:10.20944/preprints202511.2077.v1

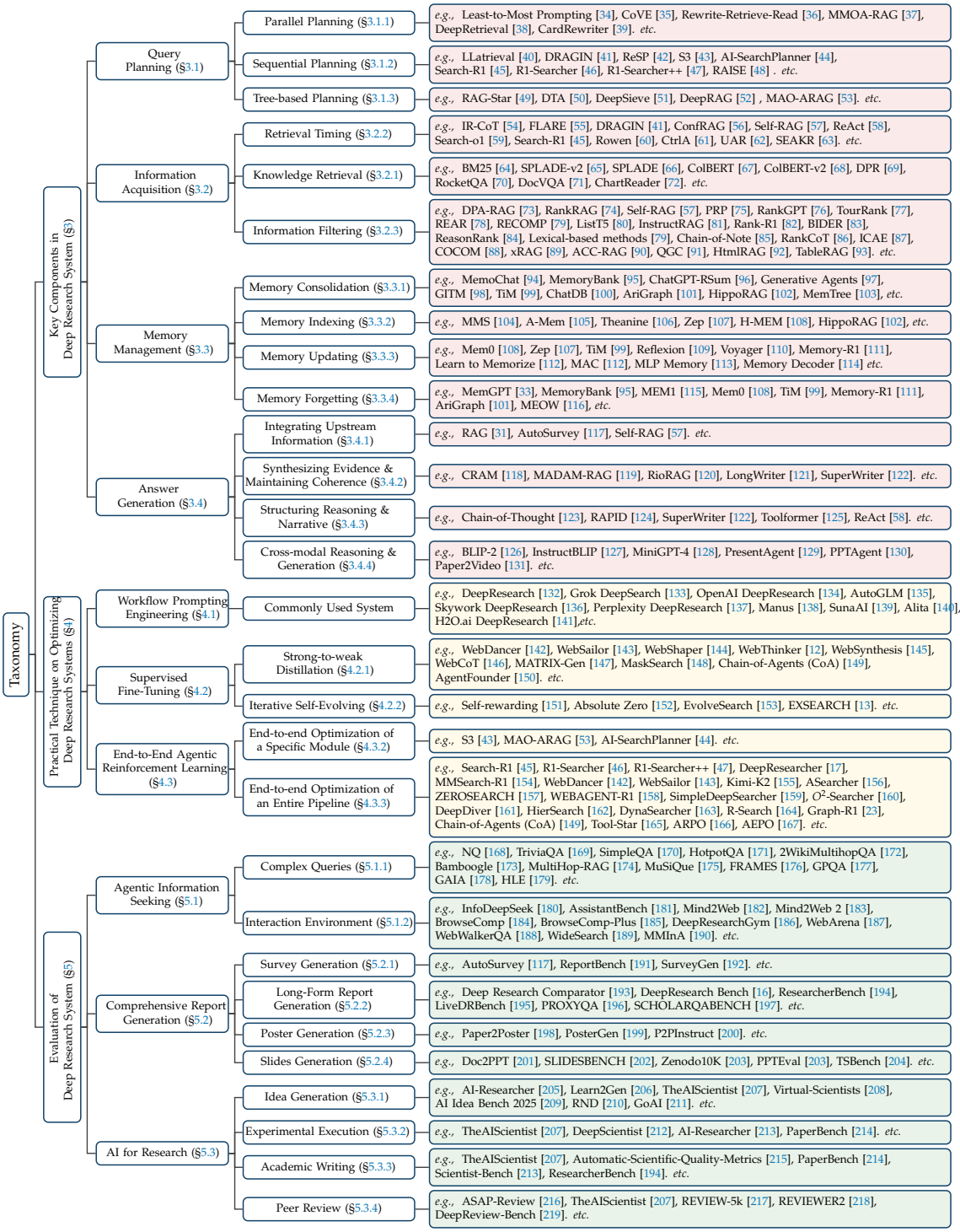

Figure 2. Taxonomy of the main content of this survey.

 doi:10.20944/preprints202511.2077.v1

Table 1. Comparison between conventional RAG (leftmost column) and the three envisioned stages of Deep Research (right columns). The capabilities evolve from static retrieval and generation to adaptive, autonomous, and scientifically creative workflows.

Capability (Key Feature)	Standard RAG	Agentic Search	Integrated Research	Full-stack AI Scientist
Search Engine Access	✓	✓	✓	✓
Use of Various Tools (e.g., Web APIs)	✗	✓	✓	✓
Code Execution for Experiment	✗	✗	✗	✓
Reflection for Action Correction	✗	✓	✓	✓
Task-solving Memory Management	✗	✓	✓	✓
Innovation and Hypothesis Proposal	✗	✗	✗	✓
Long-form Answer Generation & Validation	✓	✗	✓	✓
Action Space	Narrow	Broad	Broad	Broad
Reasoning Horizon	Single	Long-horizon	Long-horizon	Long-horizon
Workflow Organization	Fixed	Flexible	Flexible	Flexible
Output Form and Application	Short Span	Short Span	Report	Academic Paper

2.3. Comparing Deep Research with RAG

Many real-world tasks are inherently open-ended, involving **critical thinking**, **factually grounded information**, and **self-contained** responses. These present several fundamental limitations of existing approaches. Below, we summarize three key challenges that cannot be solved by conventional RAG or scaling LLM parameters alone:

- *Flexible Interaction with the Digital World.* Conventional RAG systems operate in a static retrieval loop, relying solely on pre-indexed corpora [247,248]. However, real-world tasks often require active interaction with dynamic environments such as search engines, web APIs, or even Code executors [239,245,249]. DR systems extend this paradigm by enabling LLMs to perform multi-step, tool-augmented interactions, allowing agents to access up-to-date information, execute operations, and verify hypotheses within a digital ecosystem.
- *Long-horizon Planning with Autonomous Workflows.* Complex research-like problems often require agents to coordinate multiple subtasks [184], manage task-solving context [250], and iteratively refine intermediate outcomes [109]. DR addresses this limitation through closed-loop control and multi-turn reasoning, allowing agents to autonomously plan, revise, and optimize their workflows toward long-horizon objectives.
- *Reliable Language Interfaces for Open-ended Tasks.* LLMs are prone to hallucination and inconsistency [251–255], particularly in open-ended settings. DR systems introduce verifiable mechanisms that align natural language outputs with grounded evidence, establishing a more reliable interface between human users and autonomous research agents.

3. Key Components in Deep Research System

A DR system can be viewed as a closed-loop workflow that takes a complex research question as input and produces a structured answer, typically in the form of long-form text with citations or synthesized reports. As illustrated in Figure 1, the DR system iteratively cycles through a set of interconnected components: (i) *query planning*, which decomposes the original question into sub-queries and tool calls that guide the workflow; (ii) *knowledge acquisition*, which retrieves and filters relevant information from external corpora, tools, or APIs; (iii) *memory management*, which stores, updates, and prunes intermediate findings to maintain context over long horizons; and (iv) *answer generation*, which synthesizes the accumulated evidence into a coherent, verifiable response with citations and checks for consistency. In this work, we provide detailed definitions and functionality for each component, along with representative works.

 doi:10.20944/preprints202511.2077.v1

3.1. Query Planning

Query Planning refers to the process of transforming a complex and logically intricate question into a structured sequence of executable sub-queries (*aka.*, sub-tasks), each of which can be addressed incrementally. This decomposition allows stepwise reasoning and knowledge acquisition, thereby enhancing the reliability and accuracy of the final output generated by deep research system .

Figure 3 shows three widely-used strategies for query planning: (i) *parallel planning*, which decomposes the input into independent sub-queries that may be resolved in parallel [35,37]; (ii) *sequential planning*, which arranges sub-queries into a linear order where each step depends on intermediate outcomes [45,256]; and (iii) *tree-based planning*, which explores branching decision spaces and selects among candidate paths through pruning, backtracking, or heuristic-guided search [257].

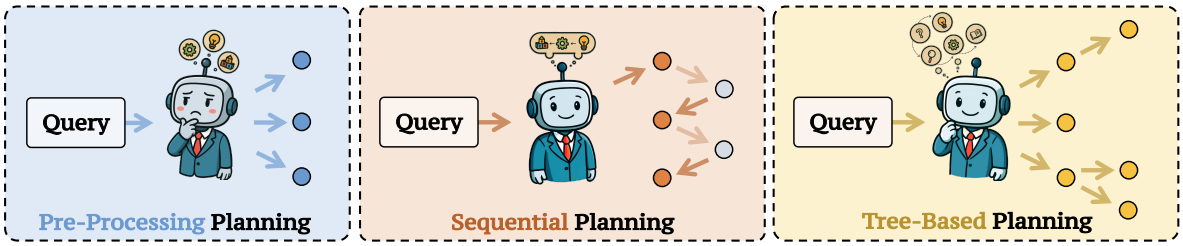

Figure 3. Three commonly-used types of query planning: (i) parallel planning; (ii) sequential planning; and (iii) tree-based planning.

3.1.1. Parallel Planning

Definition. As illustrated in Figure 3(a), parallel planning operates by rewriting or decomposing the original query into multiple sub-questions in a single pass, typically without iterative interaction with downstream components. The primary advantage of this strategy lies in its efficiency: simultaneous generation enables parallel processing of sub-queries.

Representative Work. Early research typically instantiates parallel planning modules through heuristic approaches, most notably via prompt engineering [34,35] or training on manually annotated datasets. For example, Least-to-Most Prompting [34] guides GPT-3 [258] to decompose a complex task into an ordered sequence of simpler, self-contained sub-queries in a few-shot setting. Similarly, CoVE [35] prompts LLMs to first generate multiple independent sub-questions and then ground each one with well-established evidence in parallel, a strategy widely adopted in knowledge-intensive applications.

Despite these advancements, query planning based on general heuristics or task-agnostic supervision often suffers from misalignment with end-to-end objectives in downstream applications, particularly in complex QA scenarios [29,171,172,175]. To mitigate this issue, recent work has turned to end-to-end planning optimization via RL. For example, the Rewrite-Retrieve-Read framework [36] trains a query planner to maximize final answer accuracy using the Proximal Policy Optimization algorithm [259]. Crucially, the planner is reinforced only when documents retrieved by its sub-queries enable an LLM to generate a correct answer, which replaces reliance on heuristic decomposition rules. Building on this approach, subsequent efforts such as DeepRetrieval [38] and CardRewriter [39] have extended reward modeling for query planners to incorporate diverse downstream metrics (*e.g.*, evidence recall, retrieval NDCG@k). More recently, studies have also explored jointly optimizing query planning with other components in modular dense retrieval pipelines through multi-agent RL methods [37].

Advantages & Disadvantages. Despite their efficiency, parallel planning has two primary limitations. First, they typically operate in a *one-shot* fashion, interacting with other modules (*e.g.*, retriever, reasoner, aggregator) non-iteratively. As a result, they lack mechanisms to incorporate intermediate evidence, correct earlier decisions, or adaptively allocate computational resources. Second, they often *ignore data and logical dependencies* across sub-queries. Parallel execution assumes conditional independence, yet many real-world queries involve sequential reasoning in which later subtasks depend on the resolution

 doi:10.20944/preprints202511.2077.v1

of earlier ones. This can result in ill-posed or unanswerable sub-queries due to missing contextual information.

3.1.2. Sequential Planning

Definition. As illustrated in Figure 3(b), the sequential planning decomposes the original query through multiple iterative steps, where each round of decomposition builds upon the outputs of previous rounds. At each stage, the sequential planning may invoke different modules or external tools to process intermediate results, enabling a dynamic, feedback-driven reasoning process. This multi-turn interaction allows the sequential planning to perform logically dependent query decompositions that are often intractable for pre-processing planning, which typically assumes conditional independence among sub-queries. By incorporating intermediate evidence and adapting the query trajectory accordingly, sequential planning is particularly well-suited for complex tasks that require stepwise inference, disambiguation, or progressive information gathering.

Representative Work. The sequential planning is often used to provide a series of sub-queries for the external knowledge needed in a step-by-step manner, which has been widely used in iterative QA systems [40–42]. For example, LLatrieval [40] introduces an iterative query planner that, whenever the current documents fail verification, leverages the LLM to pinpoint missing knowledge and generate a new query, either a question or a pseudo-passage, to retrieve supplementary evidence, repeating the cycle until the accumulated context fully supports a verifiable answer. DRAGIN [41] introduces a query planner that can utilize the self-attention scores to select the most context-relevant tokens from the entire generation history and reformulate them into a concise and focused query. This dynamic, attention-driven approach produces more accurate queries compared to the static *last sentence* or *last n tokens* strategies in previous methods, resulting in higher-quality retrieved knowledge and improved downstream generation. In ReSP [42], the query planner dynamically guides each retrieval iteration by formulating novel sub-questions explicitly targeted at identified information gaps whenever the currently accumulated evidence is deemed insufficient. By conditioning this reformulation process on both global and local memory states and by disallowing previously issued sub-questions, the approach mitigates the risks of over-planning and redundant retrieval. This design ensures that each newly generated query substantially contributes to advancing the multi-hop reasoning trajectory toward the final answer. RAISE [48] sequentially decomposes a scientific question into sub-problems, generates logic-aware queries for each, and retrieves step-specific knowledge to drive planning and reasoning. Additionally, S3 [43] and AI-SearchPlanner [44] both adopt sequential decision-making to control when and how to propose retrieval queries during multi-turn search. At each turn, the sequential planner evaluates the evolving evidence state and decides whether to retrieve additional context or to stop. Besides, more recent studies, including Search-R1 [45], R1-Searcher [46,47] integrate a sequential planning strategy into an end-to-end, multi-turn search framework, thereby leveraging LLMs' internal reasoning for query planning.

Advantages & Disadvantages. Sequential planning enables dynamic, context-aware reasoning and fine-grained query reformulation, thereby facilitating more accurate acquisition of external knowledge. However, excessive reasoning turns or overly long reasoning chains can incur substantial computational costs and latency. In addition, an increased number of turns may introduce cumulative noise and error propagation, potentially causing instability during reinforcement learning training.

3.1.3. Tree-Based Planning

Definition. As illustrated in Figure 3(c), the tree-based planning integrates features of both parallel and sequential planning by recursively treating each sub-query as a node within a structured search space, typically represented as a tree or a directed acyclic graph (DAG) [260]. This structure enables the use of advanced search algorithms, such as Monte Carlo Tree Search (MCTS) [261], to explore and refine potential reasoning paths. Compared to linear or flat decompositions, this approach supports more

 doi:10.20944/preprints202511.2077.v1

flexible and fine-grained decomposition of the original query, facilitating comprehensive knowledge acquisition.

Representative Work. A representative example is RAG-Star [49], which leverages MCTS in conjunction with the Upper Confidence Bound for Trees (UCT) [262] to guide a query planner in the iterative decomposition of complex questions. At each iteration, the planning model selects the most promising node using the UCT criterion, expands it by generating a sub-query and corresponding answer using a language model, evaluates the quality of the expansion via a retrieval-based reward model, and back-propagates the resulting score. This iterative process grows a reasoning tree of sub-queries until a satisfactory final answer is obtained. Other examples include DTA [50] and DeepSieve [51], which use a tree-based planner to restructure sequential reasoning traces into a DAG. This design enables the planning to aggregate intermediate answers along multiple branches and improves the model's ability to capture both hierarchical and non-linear dependencies across sub-tasks. DeepRAG [52] introduces tree-based planning via binary-tree exploration to iteratively decompose queries and decide parametric vs. retrieved reasoning, yielding large accuracy gains with fewer retrievals. More recently, MAO-ARAG [53] trains a planning agent that can dynamically orchestrate multiple, diverse query reformulation modules through a DAG structure. This adaptive workflow enables comprehensive query decomposition to enhance performance.

Advantages & Disadvantages. Tree-based planning integrates the strengths of parallel and sequential planning. It facilitates the decomposition of interdependent sub-queries and supports local parallel execution, striking an effective balance between efficiency and effectiveness. Nevertheless, training a robust Tree-based Planning module is challenging, requiring precise dependency modeling, careful trade-offs between speed and quality, addressing data scarcity, and tackling credit assignment issues in reinforcement learning.

Takeaway

This section on query planning provides a detailed overview of strategies for enhancing DR systems by decomposing complex queries into simpler, manageable subtasks. Each type of planning strategy offers unique benefits and faces specific challenges.

- *Pre-processing planning* is efficient in executing sub-queries simultaneously, though they may overlook dependencies between them.
- *Sequential planning* excels in managing dependencies through iterative processes but can incur higher computational costs.
- *Tree-based planning* strikes a balance by combining the strengths of both sequential and pre-processing approaches, allowing for adaptive and flexible query decomposition.

3.2. Information Acquisition

DR systems often acquire external information to augment LLMs' internal knowledge. However, due to the cost of retrieval and the uncertainty of document quality, it is necessary to determine when retrieval is needed [263–265]. Moreover, how to perform retrieval and manage retrieved information is key to the DR system's interaction with external knowledge. In the following, we discuss retrieval tools, retrieval timing, and information filtering in turn.

3.2.1. Retrieval Tools

Definition. In the context of DR, *retrieval tools* [266–268] are used to identify relevant information from large-scale corpora in response to a query, typically containing indexing and search techniques. Within typical DR workflows, retrieval serves as a core mechanism for bridging knowledge gaps by surfacing candidate evidence that can then be checked for accuracy, filtered for relevance, or combined into a coherent answer. Below, we systematically review widely adopted retrieval techniques, organized by modality: (i) *text-only retrieval*, and (ii) *multimodal retrieval*.

 doi:10.20944/preprints202511.2077.v1

Text Retrieval. Conceptually, modern text retrieval can be organized into three families: (i) lexical retrieval, (ii) semantic retrieval, and (iii) commercial web search. Lexical and semantic retrieval are typically implemented on local resources, while commercial web search is typically accessed only via paid APIs. Specifically, *lexical retrieval* refers to methods that match documents based on exact term overlaps and statistical term weighting, including traditional approaches like TF-IDF and BM25 [64], as well as neural sparse models that learn to expand queries and documents with relevant terms while maintaining interpretable inverted-index structures [65,66,66–70,269].

Different from the lexical retrieval, *semantic retrieval* refers to dense neural methods that encode queries and documents into continuous vector spaces to capture semantic similarity beyond exact term matching [270–273], which has been widely adopted in recent works [13,45].

More recently, *commercial web search* (like Google or Bing) has also been widely used in DR systems and web agents [274–277]. It diverges from lexical and semantic retrieval models by providing access to real-time information, leveraging massive-scale web crawling and indexing, incorporating sophisticated ranking algorithms that consider authority and freshness signals, and offering built-in fact verification through cross-source validation. Previous work, such as WebGPT [32] and SearchGPT [278], demonstrates that commercial search APIs enable research agents to access current events and dynamic content that would be missing from static corpora.

Recent studies [12,59,279,280] exemplify a shift towards more autonomous and capable research agents. These models feature deep web exploration capabilities, allowing them to interactively navigate beyond static search results to gather information. Overall, the evolution from lexical and semantic retrieval to commercial web search marks a shift from static, closed-corpus search toward dynamic, real-world information access, enabling DR systems to retrieve not only relevant but also timely and verifiable knowledge.

Multimodal Retrieval. Multimodal retrieval aims to mine multimodal information, including text, layout, and visuals (figures, tables, charts), and to preserve grounded pointers (spans, cells, coordinates) for verifiable citation, while maximizing recall under tight latency to support iterative DR. Multimodal information retrieval can be organized into three classes based on the primary type of information modality being indexed and retrieved: (i)*text-aware retrieval with layout*, which indexes titles, captions, callouts, and surrounding prose and leverages document understanding models (LayoutLM [281], Donut [282], DocVQA [71]) plus layout/metadata filters; (ii) *visual retrieval via text–image similarity*, which encodes figures and chart thumbnails with CLIP [283], SigLIP [284], or BLIP [285] and performs ANN search for text-to-image matching or composed image retrieval [286]; and (iii) *structure-aware retrieval over parsed tables and charts*, which indexes axes, legends, data marks, and table schemas to support grounded lookup of numeric facts and relations (*e.g.,* ChartReader [72] or Chartformer [287]). These three approaches are typically combined: queries are searched across all indices simultaneously, with results fused using reciprocal-rank fusion [288] or cross-modal reranking to preserve grounded pointers for citations. Recent chart-focused VLMs [289–292] further enhance the quality of visual-textual features.

Comparing Text Retrieval and Multimodal Retrieval. Compared to text-only retrieval, multimodal retrieval provides several key advantages. First, it captures visually encoded information and numeric trends that text-based methods often overlook, and facilitates cross-modal verification through hybrid fusion [288]. Second, it enables grounded citations using techniques such as layout parsing (*e.g.,* LayoutLM [281], Donut [282]) and chart understanding (*e.g.,* ChartReader [72] or Chartformer [287]). However, multimodal retrieval also presents several challenges, including increased computational costs for visual processing [283,284], sensitivity to OCR errors and variations in chart formats [293,294], and the complexity of aligning information across different modalities.

 doi:10.20944/preprints202511.2077.v1

Takeaway

Knowledge retrieval for DR has evolved from traditional lexical and dense-text search to the use of real-time commercial web search engines for up-to-date information. However, text-only methods fail to capture information embedded in visual elements like charts, tables, and layouts. Multimodal retrieval addresses this gap by modeling visual and structural data. Its primary contribution is enabling grounded, verifiable citations by linking retrieved evidence back to specific data points (*e.g.*, table cells or chart coordinates), though this introduces higher computational costs and challenges in cross-modal alignment and format processing.

3.2.2. Retrieval Timing

Definition. Retrieval timing refers to determining when a model should trigger retrieval tools during information seeking, which is also known as adaptive retrieval [60,63,295]. Because the quality of retrieved documents is not guaranteed, blindly performing retrieval at every step is often suboptimal [13,296,297]. Retrieval introduces additional computational overhead, and low-quality or irrelevant documents may even mislead the model or degrade its reasoning performance [256]. Consequently, adaptive retrieval aims to invoke retrieval only when the model lacks sufficient knowledge, which requires the model to recognize its own knowledge boundaries [298–301], *i.e.*, knowing what it knows and what it does not.

Prior work on adaptive retrieval follows two main directions: (i) estimating and enhancing a model's ability to *recognize its own knowledge boundaries* for a given query, and (ii) optimizing the *retrieval-trigger mode*l in multi-step settings to maximize downstream task performance.

Confidence Estimation as a Proxy for Boundary Perception. There are extensive works that investigate LLMs' perception of their knowledge boundaries. The degree to which a model perceives its boundaries is typically measured by the alignment between its confidence and factual correctness. Since factual correctness is typically evaluated by comparing the model's generated answer with the ground-truth answer, existing studies focus on how to measure the model's confidence, which can be broadly divided into four categories.

- *Probabilistic Confidence*. This line of work treats a model's token-level generation probabilities as its confidence in the answer [302–308]. Prior to the emergence of LLMs, a line of work had already shown that neural networks tend to be poorly calibrated, often producing overconfident predictions even when incorrect [302–304]. More recently, some research[305,306] reported that LLMs can be well calibrated on structured tasks such as multi-choice question answering or appropriate prompts, but for open-ended generation tasks, predicted probabilities still diverge from actual correctness. To address this gap, Duan et al. [308] proposed SAR, which computes confidence by focusing on important tokens, while Kuhn et al. [307] introduced semantic uncertainty, which estimates confidence from the consistency of outputs across multiple generations.
- *Consistency-based Confidence*. Since probabilistic confidence often fails to capture a model's semantic certainty and is inapplicable to black-box models without accessible generation probabilities, recent works represent confidence via semantic consistency across multiple responses [60,307,309–311]. The key idea is that a confident model should generate highly consistent answers across runs. Fomicheva et al. [309] first measured consistency through lexical similarity, while later studies used NLI (*i.e.*, natural language inference) models or LLMs to assess semantic consistency [307,310]. To address the issue of consistent but incorrect answers, Zhang et al. [311] measure consistency across different models, as incorrect answers tend to vary between models, whereas correct ones align. Ding et al. [60] further extended this idea to multilingual settings.
- *Confidence Estimation Based on Internal States*. LLMs' internal states have been shown to capture the factuality of their generated content [312–317]. Azaria and Mitchell [312] first discovered that internal states can signal models' judgment of textual factuality. Subsequent studies [313,314] found that internal states after response generation reflect the factuality of self-produced answers.

 doi:10.20944/preprints202511.2077.v1

More recently, Wang et al. [315] and Ni et al. [316] demonstrated that factuality-related signals already exist in the pre-generation states, enabling the prediction of whether the output will be correct.

- *Verbalized Confidence*. Several studies explore enabling LLMs to express confidence in natural language, akin to humans, viewing such verbalization as a sign of intelligence [56,318–323]. Yin et al. [319] and Ni et al. [56] examined whether LLMs can identify unanswerable questions, finding partial ability but persistent overconfidence. Other works [320,321] investigated fine-grained confidence expression. Xiong et al. [321] offered the first comprehensive study for black-box models, while Tian et al. [320] proposed generating multiple answers per pass for more accurate estimation. Beyond prompting, some methods explicitly train models to verbalize confidence [318,322,323], with Lin et al. [318] introducing this idea and using correctness-based supervision.

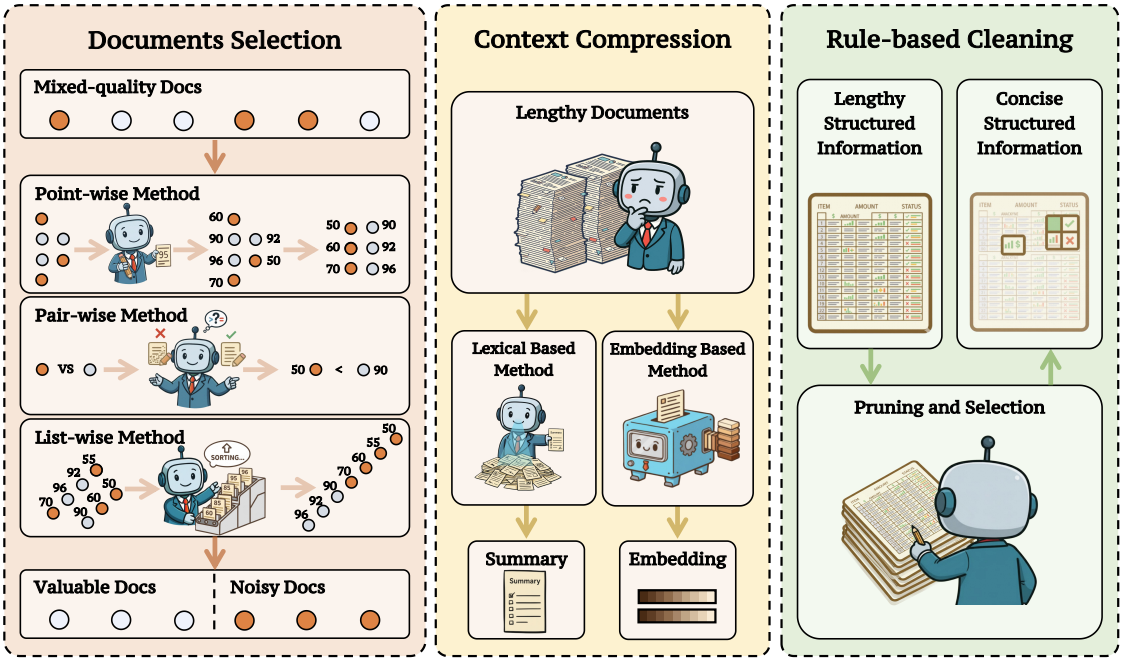

Figure 4. Existing information filtering approaches can be broadly categorized into the following types: (i) *Document Selection*; (ii) *Context Compression*; and (iii) *Rule-based Cleaning*.

Representative Adaptive Retrieval Approaches. Deep research systems typically involve iterative interactions between model inference and external document retrieval, differing mainly in how they determine when to retrieve. Early works such as IR-CoT [54] enforce retrieval after every reasoning step, ensuring continual grounding in external knowledge but at the cost of efficiency. Building on insights from studies of models' perceptions of their own knowledge boundaries, recent approaches treat retrieval as a model-issued action, enabling the model to perform it dynamically only when needed. Similar to techniques in confidence estimation, these methods assess whether the model can answer a question correctly given the current context and perform retrieval when knowledge is deemed insufficient. They can be broadly categorized into four paradigms.

- *Probabilistic Strategy*. It triggers retrieval based on token-generation probabilities: when the model produces a token with low confidence, retrieval is initiated [41,55].
- *Consistency-based Strategy*. Recognizing that both token-level probabilities and single-model self-consistency may fail to capture true semantic uncertainty, Rowen [60] evaluates consistency across responses generated by multiple models and languages, triggering retrieval when cross-model or cross-lingual agreement is low.

 doi:10.20944/preprints202511.2077.v1

- *Internal States Probing*. CtrlA [61], UAR [62], and SEAKR [63] further propose that compared to generated responses, a model's internal states provide a more faithful reflection of its confidence, using them to guide adaptive retrieval decisions.
- *Verbalized Strategy*. It enables the model to directly express its confidence via natural language. These methods typically generate special tokens directly in the response to indicate the need for retrieval. ReAct [58] directly prompts the model to generate corresponding action text when retrieval is needed. Self-RAG [57] trains the model to explicitly express uncertainty through the special token (*i.e.,* `<retrieve>`), signaling the need for retrieval. With LLMs' growing reasoning capacity, recent research has shifted toward determining retrieval timing through reasoning and reflection. Search-o1 [59] introduces a Reason-in-Documents module, which prompts the model to selectively invoke search during reasoning. Search-R1 [45] further frames retrieval as part of the environment and employs reinforcement learning to jointly optimize both when and what to retrieve.

Collectively, these methods trace an evolution from fixed or per-step retrieval (*e.g.,* IR-CoT [54]) to dynamically triggered retrieval (*e.g.,* ReAct [58], Self-RAG [57], Search-o1 [59]), and finally to RL–based systems that explicitly train retrieval policies (*e.g.,* Search-R1 [45]).

3.2.3. Information Filtering

Definition. Information filtering refers to the process of selecting, refining, or transforming retrieved documents so that only the most relevant and reliable evidence is passed to subsequent steps. Since retrieval tools are not perfect, the retrieved information often contains considerable noise [324–326]. This includes the content that is entirely irrelevant to the query or plausible-looking statements that nevertheless provide incorrect or misleading context. As shown in prior work [324,327], LLMs are highly sensitive to such noise; without additional filtering or optimization, they can be easily misled into generating incorrect or hallucinated responses. Figure 4 summarizes three information filtering approaches: (i) *Document Selection*, (ii) *Context Compression*, and (iii) *Rule-based Cleaning*.

Document Selection. Document selection aims to rank a set of candidate documents based on their relevance and usefulness to the query, selecting the top-k helpful documents for question answering [74,79,81]. This selection operation reduces the impact of noisy documents on LLMs, improving the question-answering accuracy in downstream tasks. Below, we review three document selection strategies: *point-wise* selection, *pair-wise* selection, and *list-wise* selection.

- *Point-wise Selection*. Given an initially retrieved document list, **point-wise** methods independently score each candidate document. The most common approach involves fine-tuning an embedding model (*e.g.,* BGE [328]) that encodes the query and each document separately, after which their relevance is estimated via inner-product similarity [79,329]. Another widely adopted strategy employs a cross encoder, which takes the concatenation of the query and a document as input and directly predicts a binary relevance score [73,78]. More recently, several studies have leveraged LLMs' natural language understanding capabilities for relevance assessment. These methods train LLMs to output special tokens, such as `<ISREL>` [57] or the identifier `True` [74], to indicate whether an input document is relevant to the query.
- *Pair-wise Selection*. Unlike the point-wise approach, which assigns an absolute relevance score, the pair-wise method compares the relevance of two input candidate information snippets (typically two documents) and predicts which one is more relevant to the query. Pair-wise selection is less common than point-wise selection. A representative work is PRP [75], which adopts a pairwise-ranking-prompting approach. In PRP, the LLM receives a query and two candidate documents to decide which is more relevant, and the final ranking list is then obtained using a heapsort algorithm. To mitigate positional bias, PRP performs the comparison twice, swapping the document order each time, and aggregates the results to yield a more stable judgment.
- *List-wise methods*.Given a document list, a list-wise selection strategy directly selects the final set of relevant documents from the candidate list. A representative work is RankGPT [76], which feeds

 doi:10.20944/preprints202511.2077.v1

the entire candidate sequence into an LLM and leverages prompt engineering to produce a global ranking. In addition to RankGPT, other work, such as TourRank [77], uses a tournament-inspired strategy to generate a robust ranking list [77,80]. ListT5 [80] proposes a list re-ranking method based on the Fusion-in-Decoder (FiD) [330] architecture, which independently encodes multiple documents in parallel and orders them by relevance, mitigating positional sensitivity while preserving efficiency. For large document sets, it builds m-ary tournament trees to group, rank, and merge results in parallel. Recently, more and more work has employed the reasoning model for list-wise document selection, advancing document selection by explicitly modeling a chain of thought. For example, InstructRAG [81] trains an LLM to generate detailed rationales via instruction tuning [331], directly judging the usefulness of each document in the raw retrieved document list. Rank-R1 [82] employs the reinforcement learning algorithm GRPO [332] to train the LLM, enabling it to learn how to select the documents most relevant to a query from a list of candidates. ReasonRank [84] empowers a list-wise selection model through a proposed multi-view ranking-based GRPO [332], training an LLM on automatically synthesized multi-domain training data.

Content Compression. Content Compression aims to remove redundant or irrelevant information from retrieved knowledge, thereby increasing the density of useful content within the model's context. Existing approaches primarily fall into two categories: *lexical-based* and *embedding-based* methods.

- **Lexical-based methods** condense retrieved text into concise natural language, aiming to only include the key point related to the given query [79,333]. Representative works such as RECOMP [79] fine-tune a smaller, open-source LLM to summarize the input retrieved documents, where the ground truth is synthesized by prompting powerful commercial LLMs like GPT-4 [334]. Chain-of-Note [85] introduces a reading-notes mechanism that compels the model to assess the relevance of retrieved documents to the query and extract the most critical information before generating an answer, with training data annotated by GPT-4 and further validated through human evaluation. Other work, like BIDER [83], eliminates reliance on external model distillation by synthesizing Key Supporting Evidence (KSE) for each document, using it for compressor SFT, and further optimizing with PPO based on gains in answer correctness. Zhu et al. [335] argue that previous compressors optimized with log-likelihood objectives failed to precisely define the scope of useful information, resulting in residual noise. They proposed a noise-filtering approach grounded in the information bottleneck principle, aiming to maximize the mutual information between the compressed content and the target output while minimizing it between the compressed content and retrieved passages. RankCoT [86] implicitly learns document reranking during information refinement. It first employs self-reflection to generate summary candidates for each document. In subsequent DPO [259] training, the compression model is encouraged to assign higher probabilities to correct summaries when all documents are fed in, thereby inducing implicit reranking in the final summarization.
- **Embedding-based methods** compress context into dense embedding sequences [90,336,337]. Because embedding sequences can store information flexibly, embedding-based methods can be more efficient and effective than lexical-based methods. ICAE [87] uses an encoder to compress context into fixed-length embedding sequences and designs training tasks to align the embedding space with the answer generation model. COCOM [88] jointly fine-tunes the encoder and answer generation model, enhancing the latter's ability to capture the semantics of embeddings. xRAG [89] focuses on achieving extreme compression rates. It introduces a lightweight bridging module, initialized with a two-layer MLP and trained through paraphrase pretraining and context-aware instruction tuning. This module projects the document embedding vectors originally used for initial retrieval into a single token in the answer generation model's representation space, achieving contextual compression with only a single additional token. ACC-RAG [90] adapts compression rates for different documents by employing a hierarchical compressor to produce multi-granularity embedding sequences and dynamically selecting compression rates based on

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 November 2025 doi:10.20944/preprints202511.2077.v1

query complexity. Similarly, QGC [91] adjusts compression rates based on query characteristics, dynamically selecting different rates for different documents based on their relevance to the query.

Rule-based Cleaning. Rule-based methods are effective for cleaning externally sourced information with specific structures. For example, HtmlRAG [92] applies rule-based compression to remove structurally present but semantically empty elements, such as CSS styling and JavaScript code, from retrieved web pages. This is combined with a two-stage block-tree pruning strategy that first uses embeddings for coarse pruning, followed by a generative model for fine-grained pruning. Separately, TableRAG [93] accurately extracts core table information through schema retrieval, which identifies key column names and data types, and cell retrieval, which locates high-frequency cell value pairs. This method addresses the challenges of context length limitations and information loss in large table understanding.

Advantages & Disadvantages. Filtering the retrieved knowledge is a simple yet effective strategy to enhance the performance of DR systems, as widely demonstrated in previous work [73,79,92]. However, incorporating an additional filtering module typically incurs additional computational costs and increased latency [325]. Moreover, overly filtering may remove useful or even correct information, thereby degrading model performance. Therefore, balancing filtering precision and information retention is crucial for building efficient and reliable DR systems.

Takeaway

Knowledge filtering can further process the metadata retrieved by DR systems, providing them with more concise and useful external knowledge while reducing noise interference and attention dilution caused by long context lengths. Filtering methods can be categorized into post-ranking selection, context compression, and rule-based cleaning. However, these knowledge filtering techniques often introduce additional time and computational costs. Therefore, different DR systems should choose the most suitable filtering method based on task characteristics to balance performance and resource consumption.

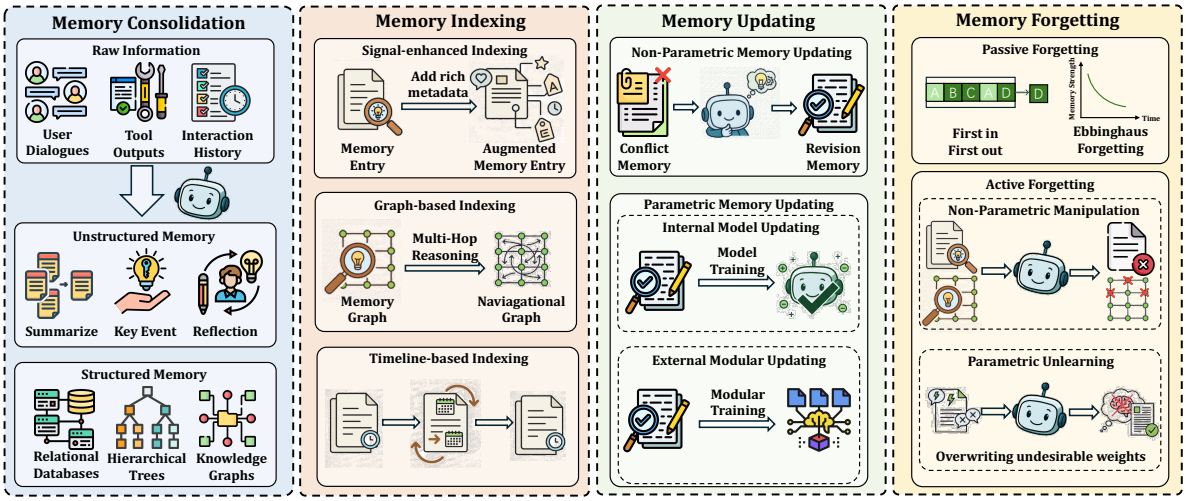

Figure 5. Memory management contains four key stages: (1) Memory Consolidation, (2) Memory Indexing, (3) Memory Updating, and (4) Memory Forgetting.

3.3. Memory Management

Definition. Memory management is a foundational component of advanced DR architectures, which governs the dynamic lifecycle of context used by DR agents in complex, long-horizon tasks [338–340], aiming to maintain coherent and relevant task-solving context [341–343].

 doi:10.20944/preprints202511.2077.v1

Core Operation. As illustrated in Figure 5, memory management typically involves four core operations: consolidation, indexing, updating, and forgetting. Consolidation converts short-term experiences into durable representations that form the basis for later indexing. Indexing organizes these representations into retrieval structures that support efficient recall during problem solving. Updating refines or corrects stored knowledge, whereas forgetting selectively removes outdated or irrelevant content to reduce interference. In the following sections, we discuss consolidation, indexing, updating, and forgetting in detail.

3.3.1. Memory Consolidation

Definition. Memory consolidation is the process of transforming transient, short-term information, such as user dialogues or tool execution outputs, into stable, long-term representations [338,339,344]. Drawing an analogy to cognitive neuroscience, this process encodes and abstracts raw inputs to create durable memory engrams, laying the groundwork for efficient long-term storage and retrieval [338].

Memory consolidation involves transforming interaction histories into durable formats, including but not limited to model parameters [345], structured graphs [346], or knowledge bases [94,339]. Distinct from memory indexing, which creates navigable access pathways over existing memories, consolidation is fundamentally concerned with the initial transformation and structural organization of raw experience. Two primary paradigms for this process have emerged: (i) *unstructured memory consolidation* and (ii) *structured memory consolidation*.

Unstructured Memory Consolidation. This paradigm distills lengthy interaction histories or raw texts into high-level, concise summaries or key event logs. For example, MemoryBank [95] processes and distills conversations into a high-level summary of daily events, which helps in constructing a long-term user profile. Similarly, MemoChat [94] summarizes conversation segments by abstracting the main topics discussed, while ChatGPT-RSum [96] adopts a recursive summarization strategy to manage extended conversations. Other approaches focus on abstracting experiences; Generative Agents [97] utilize a reflection mechanism triggered by sufficient event accumulation to generate more abstract thoughts as new, consolidated memories. To create generalizable plans, GITM [98] summarizes key actions from multiple successful plans into a common reference memory.

Structured Memory Consolidation. This paradigm transforms unstructured information into highly organized formats such as databases, graphs, or trees. This structural encoding is the primary act of consolidation, designed to capture complex inter-entity relationships and create an organized memory corpus. For instance, TiM [99] extracts entity relationships from raw information and stores them as tuples in a structured database. ChatDB [100] leverages a database as a form of symbolic memory, transforming raw inputs into a queryable, relational format. AriGraph [101] implements a memory graph where knowledge is represented as vertices and their interconnections as edges. Similarly, HippoRAG [102] constructs knowledge graphs over entities, phrases, and summaries to form an interconnected web of fragmented knowledge units. MemTree [103] builds and updates a tree structure by traversing from the root and deciding whether to deepen the tree with new information or create new leaf nodes based on semantic similarity. This hierarchical organization is the core of its consolidation strategy, enabling structured storage of memories.

3.3.2. Memory Indexing

Definition. Memory indexing involves constructing a navigational map over a DR agent's consolidated memories, analogous to a library's catalog or a book's index for efficient information retrieval [347]. Unlike memory consolidation, which focuses on the initial transformation of raw data into a durable format, indexing operates on already consolidated memories to create efficient, semantically rich retrieval pathways. This process builds auxiliary access structures that enhance retrieval not only in efficiency but also in relevance.

 doi:10.20944/preprints202511.2077.v1

Effective indexing goes beyond simple keyword matching by encoding temporal [348] and relational [102] dependencies among memories. This is typically achieved by generating auxiliary codes, such as vector embeddings, summaries, or entity tags, which serve as retrieval entry points into the memory store. Given the vast, high-dimensional spaces these codes inhabit, specialized search techniques are required, such as Locality-Sensitive Hashing (LSH) [349], Hierarchical Navigable Small World (HNSW) graphs [350], or libraries like FAISS [351] for high-speed similarity search. These access mechanisms are commonly organized through three established paradigms:

- **Signal-enhanced Indexing.** This paradigm augments consolidated memory entries with auxiliary metadata, including emotional context, topics, and keywords, which function as granular pivots for context-aware retrieval [104,352]. For instance, LongMemEval [353] enhances memory keys by integrating temporal and semantic signals to improve retrieval precision. Similarly, the Multiple Memory System (MMS) [104] decomposes experiences into discrete components, such as cognitive perspectives and semantic facts, thereby facilitating multifaceted retrieval strategies.
- **Graph-based Indexing.** This paradigm leverages a graph structure, where memories are nodes and their relationships are edges, as a sophisticated index. By representing memory networks in this way, agents can perform complex multi-hop reasoning by traversing chains of connections to locate information that is not explicitly linked to the initial query [108,354]. For instance, HippoRAG [102] uses lightweight knowledge graphs to explicitly model inter-memory relations, enabling structured, interpretable access. A-Mem [105] adopts a dynamic strategy where the agent autonomously links related memory notes, progressively growing a flexible access network.
- **Timeline-based Indexing.** This paradigm creates a temporal index by organizing memory entries along chronological or causal sequences. Such structuring provides a historical access pathway, which is essential for understanding progression, maintaining conversational coherence, and supporting lifelong learning [355]. For example, the Theanine system [106] arranges memories along evolving timelines to facilitate retrieval based on both relevance and temporal dynamics. Zep [107] introduces a bi-temporal model for its knowledge graph, indexing each fact with t_{valid} and $t_{invalid}$ timestamps, which allows the agent to navigate the memory based on temporal validity.

3.3.3. Memory Updating

Definition. Memory updating is a core capability of DR agents, involving the reactivation and modification of existing knowledge in response to new information or environmental feedback [356–358]. This process is essential for maintaining the consistency, accuracy, and relevance of the agent's internal world model, thereby enabling continual learning and adaptive behavior in dynamic environments [110,359].

Memory updating governs how an agent corrects factual inaccuracies, incorporates new information, and gradually improves its knowledge base [109,360,361]. Although related to memory forgetting, which focuses on removing outdated or incorrect content, memory updating centers on modifying and refining existing knowledge to increase its fidelity. In the following, we introduce two updating strategies, depending on whether the memory is external (non-parametric) or internal (parametric) to the model [359].

Non-Parametric Memory Updating. Non-parametric memory, stored in external formats such as vector databases or structured files, is updated via explicit, discrete operations on the data itself. This approach offers flexibility and transparency. Key operations include:

- *Integration and Conflict Updating.* This operation focuses on incorporating new information and refining existing entries to maintain logical consistency. For example, the Mem0 framework employs an LLM to manage its knowledge base through explicit operations, such as adding new facts (`ADD`) or modifying existing entries with new details (`UPDATE`) to resolve inconsistencies [108]. To handle temporal conflicts, Zep updates its knowledge graph by modifying an existing fact's effective time range, setting an invalidation timestamp ($t_{invalid}$) to reflect that a newer fact has

 doi:10.20944/preprints202511.2077.v1

superseded it [107]. Similarly, the TiM framework curates its memory by using `MERGE` operations to combine related facts into a more coherent representation [99]

- *Self-Reflection Updating.* Inspired by human memory reconsolidation, this paradigm enables agents to iteratively refine their knowledge by reflecting on past experiences [109,362]. Early systems like Reflexion [109] and Voyager [110] implement this through verbal self-correction and updates to a skill library. More dynamically, A-Mem [105] triggers a Memory Evolution process that re-evaluates and autonomously refines previously linked memories based on new contextual information.

Parametric Memory Updating. Parametric memory, encoded directly in a model's weights, is updated by modifying internal representations. This is typically more complex and computationally intensive. Three main approaches have emerged:

- *Global Updating.* This approach integrates new knowledge by continuing model training on additional datasets [363]. While effective for large-scale adaptation, it is computationally expensive and prone to catastrophic forgetting [356]. To address this, instead of simply injecting factual knowledge, Memory-R1 trains a dedicated Memory Manager agent to learn an optimal policy for modification operations such as `ADD` and `UPDATE`, moving beyond heuristic rules [111]. Additionally, a recent framework refines this process by employing methods such as Direct Preference Optimization to fine-tune the model's memory utilization strategy [112].
- *Localized Updating.* This technique modifies specific facts in the model's parameters without requiring full retraining [360,361]. It is especially suited for online settings where rapid adaptation is needed, such as updating a user's preference [357]. Methods typically follow a *locate-and-edit* strategy or use meta-learning to predict weight adjustments while preserving unrelated knowledge [357,361].
- *Modular Updating.* This emerging paradigm avoids the risks of continual weight modification by distilling knowledge into a dedicated, plug-and-play parametric module. Frameworks such as MLP Memory [113] and Memory Decoder [114] train a lightweight external module to imitate the output distribution of a non-parametric kNN retriever. This process effectively compiles a large corpus of external knowledge into the compact weights of the module. The resulting module can then be attached to any compatible LLM to provide specialized knowledge without modifying the base model's parameters, thereby avoiding catastrophic forgetting and reducing the latency of real-time retrieval [113,114].

3.3.4. Memory Forgetting

Definition. Forgetting constitutes a fundamental mechanism in advanced agent architectures, enabling the selective removal or suppression of outdated, irrelevant, or potentially erroneous memory content. Rather than a system defect, forgetting is a functional process critical for filtering noise, reclaiming finite storage resources, and mitigating interference between conflicting information. In contrast to memory updating, which modifies existing knowledge to improve its accuracy, forgetting is a subtractive process that streamlines the memory store by eliminating specific content. This process can be broadly categorized into passive and active mechanisms.

Passive Forgetting. This simulates the natural decay of human memory, in which infrequently accessed or temporally irrelevant memories gradually lose prominence. This mechanism is particularly critical for managing the agent's immediate working memory or context window. Implementations are typically governed by automated, time-based rules rather than explicit content analysis. For instance, MemGPT [33] employs a First-In-First-Out (FIFO) queue for recent interactions, automatically moving the oldest messages from the main context into long-term storage. MemoryBank [95] draws inspiration from the Ebbinghaus forgetting curve, in which memory traces decay over time unless reinforced, allowing the agent to naturally prioritize recent content. A more aggressive approach, MEM1 [115], employs a *use-and-discard* policy: after each interaction, the agent synthesizes essential information

 doi:10.20944/preprints202511.2077.v1

into a compact state and immediately discards all prior contextual data to maintain constant memory consumption.

Active Forgetting. Active forgetting involves the intentional and targeted removal or invalidation of specific memory content. This process is a deliberate action, often triggered by the detection of contradictions or the need to correct inaccurate information, and its implementation varies depending on the memory type.

- *Non-Parametric Memory.* Active forgetting in external memory stores involves direct data manipulation. For example, Mem0 [108] implements an explicit `DELETE` command to remove outdated or contradictory facts. Similarly, TiM [99] introduces a dedicated `FORGET` operation to actively purge irrelevant or incorrect thoughts from its memory cache. Reinforcement learning can also be used to train a specialized Memory Manager agent to autonomously decide when to execute a `DELETE` command, as seen in the Memory-R1 framework [111]. AriGraph [101] maintains a structured memory graph by removing outdated vertices and edges. Some systems employ non-destructive forgetting; the Zep architecture [107], for example, uses *edge invalidation* to assign an invalid timestamp to an outdated entry, effectively retiring it without permanent deletion.
- *Parametric Memory.* In this context, active forgetting is typically achieved through machine unlearning techniques that modify a model's internal parameters to erase specific knowledge without full retraining. Approaches include locating and deactivating specific neurons or adjusting training objectives to promote the removal of targeted information. For example, MEOW [116] facilitates efficient forgetting by fine-tuning an LLM on generated contradictory facts, effectively overwriting undesirable memories stored in its weights.

Takeaway

Memory management is a cornerstone of the DR paradigm, enabling agents to transcend single-turn interactions and conduct complex, long-horizon investigations by governing the information lifecycle. Through the interdependent operations of consolidation, indexing, updating, and forgetting, the DR system maintains the context and coherence essential for an iterative research loop. Consequently, a sophisticated memory framework is what fundamentally distinguishes a DR agent from a simple RAG system, equipping it with the consistency, adaptability, and self-evolution necessary to autonomously synthesize comprehensive, trustworthy, and verifiable reports from a vast and dynamic information landscape.

3.4. Answer Generation

Definition. Answer generation typically represents the culminating stage of a DR system. It synthesizes information from upstream components, such as query planning (Section 3.1), information acquisition (Section 3.3.1), and memory systems (Section 3.3.1), and generates a coherent, comprehensive, and well-supported response that accurately reflects the user's original intent.

Unlike traditional text generation, the answer generation within an advanced DR workflow addresses complex challenges such as reconciling conflicting evidence, maintaining long-range coherence, and structuring outputs with transparent reasoning and proper citations. It has evolved from template-based generation [364] to sophisticated synthesis shown in Figure 6, which reflects the growing demand for trustworthy, explainable, and multimodal research outputs [31,365]. To deconstruct this process, we will explore it across four progressive stages: beginning with the integration of diverse information sources, moving to the synthesis of evidence and maintenance of coherence, then structuring the reasoning and narrative, and finally, advancing to the frontier of cross-modal generation.

 doi:10.20944/preprints202511.2077.v1

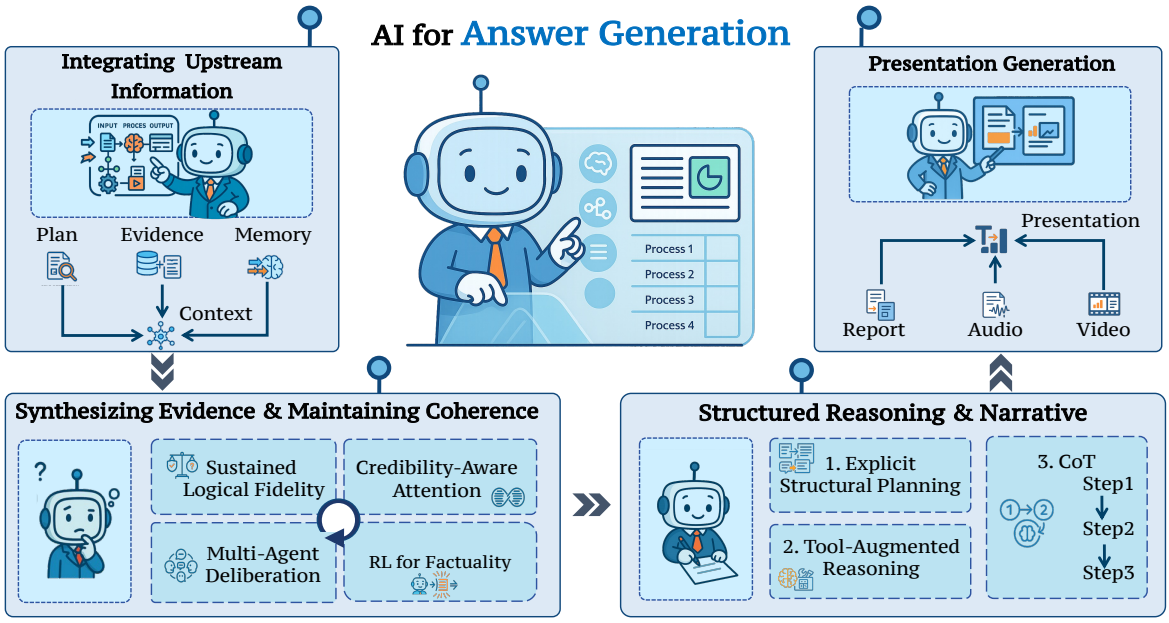

Figure 6. Illustrating the schematic of the answer generation process in DR. The workflow begins by integrating upstream information, moves to synthesizing evidence and ensuring coherence, then constructs a structured narrative via reasoning, and culminates in a multimodal presentation output.

3.4.1. Integrating Upstream Information

Definition. The main principle of trustworthy answer generation is to ensure that every statement is grounded in verifiable external evidence. Thus, the first stage of answer generation is integrating information from its upstream components, including: the sub-queries from the query planning, the ranked and potentially conflicting evidence, and the evolving contextual state stored in memory.

Recent developments in this area demonstrate sophisticated strategies for integrating upstream information, moving from simple evidence-feeding to dynamic, state-aware synthesis. The most common approach involves integrating ranked evidence from the retrieval module. Frameworks like Self-RAG [57], for example, employ a more dynamic integration by adaptively retrieving passages on demand. It then generates reflection tokens to assess the relevance of the retrieved information and its own generation, effectively integrating an internal self-correction mechanism to steer the synthesis. Moving beyond static evidence, more advanced systems integrate the query plan with an evolving memory state to ensure long-range coherence, a paradigm known as Stateful Query Planning. For instance, graph-centric frameworks like Plan-on-Graph (PoG) [366] explicitly integrate the plan with a dynamic memory (storing sub-goal status, explored paths, and retrieved entities). This memory is then actively used during a *reflection* step to guide and self-correct subsequent planning, tightly coupling the reasoning state with the generation process. Similarly, search-based frameworks like MCTS-OPS [367] formalize this by treating the MCTS tree itself as the state of the evolving query plan. Here, the system integrates its experiential memory (node values from past rollouts) to guide the `SELECTION` and `EXPANSION` of the next planning step, ensuring the final answer synthesizes the full context of the problem-solving process.

While these architectures provide a robust foundation, the core challenges of synthesizing contradictory evidence and maintaining long-form coherence remain the next frontier.

3.4.2. Synthesizing Evidence and Maintaining Coherence

Producing answers to research-level questions requires resolving informational conflicts and sustaining coherent, information-dense narration across extended outputs.

 doi:10.20944/preprints202511.2077.v1

Resolving Conflicting Evidence. Research queries frequently surface contradictory sources, requiring the model to discriminate among varying levels of reliability. Building on fact-verification paradigms [368], recent systems adopt three major strategies.

- *Credibility-Aware Attention*: Instead of treating all retrieved information equally, this approach intelligently weighs evidence based on its source. The system assigns a higher score to information coming from more credible sources (*e.g.*, a top-tier scientific journal) compared to less reliable ones (*e.g.*, an unverified blog) [118]. This allows the model to prioritize trustworthy information while still considering relevant insights from a wider range of sources [369].
- *Multi-Agent Deliberation*: This strategy simulates an expert committee meeting to debate the evidence. Frameworks like MADAM-RAG [119] employ multiple independent AI agents, each tasked with analyzing the retrieved documents from a different perspective. Each agent forms its own assessment and conclusion. Afterwards, a final *meta-reasoning* step synthesizes these diverse viewpoints to forge a more robust and nuanced final answer, much like a panel of experts reaching a consensus [370].
- *Reinforcement Learning for Factuality*: This method trains the generator through a trial-and-error process that rewards factual accuracy [371]. A representative approach is RioRAG [120], in which an LLM receives a positive reward when it generates statements that are strongly and consistently supported by the provided evidence. Conversely, it is penalized for making unsubstantiated claims or statements that contradict the source material, shaping the model to inherently prefer generating factually grounded and reliable answers.

Long-form Coherence and Information Density. Another key challenge is ensuring **Sustained Informational Accuracy**. Research answers are often lengthy, and maintaining a logical thread while avoiding repetition or verbosity is non-trivial. Let L_{model} denote the maximum coherent length of a model's output, and L_{SFT} represent the average length of examples in its supervised fine-tuning dataset. SFT offers an intuitive approach to enhancing the long-form generation capabilities of large language models. However, LongWriter [121] empirically demonstrates that the maximum coherent length of a model's output often scales with the average length of its fine-tuning samples, which can be formally expressed as $L_{\text{model}} \propto L_{\text{SFT}}$ [121]. To address this, LongWriter focuses on systematic training for extended generation, while others use reflection-driven processes to iteratively improve consistency [122]. Additionally, RioRAG [120] introduces a length-adaptive reward function to promote information density, which penalizes verbosity that fails to add informational value, preventing reward hacking through verbosity. Together, these techniques shift the focus of generation from mere content aggregation toward credible, concise, and coherent synthesis, laying the groundwork for structured reasoning.

3.4.3. Structuring Reasoning and Narrative

The research community's focus is shifting from the mere factual accuracy of DR systems to the crucial need for explainability and logical rigor in their answers. An opaque answer, which prevents users from tracing the underlying reasoning process, has significantly diminished utility in critical domains like scientific research [270,372,373]. Consequently, a significant line of work has emerged to enable models to generate structured reasoning processes rather than just monolithic final answers [34,123,124]. This trend is reflected in the design of most modern DeepResearch systems, which increasingly favor the explicit presentation of this structural information [362,374].

Prompt-based Chain-of-Thought. This foundational approach focuses on eliciting intermediate reasoning steps before producing a final answer. The most prominent technique is Chain-of-Thought (CoT) prompting [123], which can be formally expressed as $\mathcal{R} = \text{LLM}(\text{CoT-Prompt} + \mathcal{Q} + \text{Evidence})$. This method enhances both interpretability and multi-step reasoning performance. Its applicability has been broadened by extensions such as zero-shot CoT [375] and Least-to-Most prompting [34].

 doi:10.20944/preprints202511.2077.v1

Explicit Structural Planning. More advanced systems move beyond simple linear chains to formalize the structure of the entire answer. For instance, RAPID [124] formalizes this process into three stages: (i) *outline generation*; (ii) *outline refinement through evidence discovery*; and (iii) *plan-guided writing*, where the outline forms a directed acyclic graph to support complex, non-linear argumentation. Similarly, SuperWriter [122] extends this idea by decoupling the reasoning and text-production phases and optimizing the entire process via hierarchical Direct Preference Optimization.

Tool-Augmented Reasoning. This line of work enhances reasoning by dynamically interfacing with external resources. Representative work allows models to invoke external computational or retrieval tools dynamically, ensuring both analytic rigor and factual grounding [125,376–379].

3.4.4. Presentation Generation

The frontier of answer generation extends beyond text, encompassing the integration of multimodal and structured information. Research questions increasingly demand answers that combine textual reasoning with visual, tabular, or auditory data, maintaining semantic and presentational coherence. Early breakthroughs such as BLIP-2 [126] and InstructBLIP [127] enable multimodal instruction-following by aligning vision-language embeddings. MiniGPT-4 [128] advances this by leveraging cross-modal attention to seamlessly integrate visual and textual evidence.

Recently, a series of works have demonstrated higher presentation capabilities, signaling an evolution from content generation to presentation generation [131,380,381]. Existing work like MedConQA [382], LIDA [383], ChartGPT [384], and Urania [380] can synthesize data analyses into dynamic, interactive visualizations. Others work, including PresentAgent [129], Qwen2.5-Omni [385], and AnyToAny [386], generates synchronized audio narrations alongside text. More recently, PPTAgent [130] and Paper2Video [131] even extend to editable presentation generation, where full analytical reports are automatically transformed into slide decks with coordinated text, figures, and layout elements. At the leading edge, video-grounded agents [387,388] retrieve or generate relevant visual footage, delivering answers through multimodal storytelling. As summarized in Table 2, while most DR systems still focus on textual synthesis with citations, only a handful, such as OpenAI DeepResearch [134] and H2O.ai DeepResearch [141], currently support comprehensive multimodal output. The emerging consensus suggests that rich, multi-format answer generation will soon become a standard expectation [389], bridging the gap between knowledge synthesis and human-centered presentation.

Takeaway

Answer generation represents the synthesis core of DR systems, integrating upstream information, reconciling conflicting evidence, and structuring coherent, evidence-grounded narratives. Recent advances, from credibility-aware attention and multi-agent deliberation to reinforcement learning for factuality, have enhanced both factual reliability and interpretability. Systems now move beyond content aggregation toward concise, logically structured synthesis supported by transparent reasoning frameworks such as Chain-of-Thought and plan-guided writing. Moreover, the frontier of answer generation extends into multimodal generation, where text, visuals, tables, and audio coalesce into rich, human-centered outputs. These developments mark a paradigm shift from generating text to generating explainable, trustworthy, and presentation-ready knowledge.

4. Practical Techniques for Optimizing Deep Research Systems

So far, we have introduced the core components that constitute a typical DR system. Building on these foundation, we now delve into practical techniques for improving such DR systems in real-world settings. These techniques focus on how to flexibly coordinate and enhance the key components, with the goal of achieving more reliable and effective task completion. Below, we discuss three commonly used paradigms: workflow prompting, supervised fine-tuning, and agentic reinforcement learning. Workflow prompting typically relies on a carefully designed pipeline (*aka.*, prompting engineering)

 doi:10.20944/preprints202511.2077.v1

that guides the agents. The latter two paradigms aim to train a specific DR agent capable of reasoning, retrieving information, and generating high-quality answers.

4.1. Workflow Prompt Engineering

Definition. A simple yet effective way to build a DR system is to construct a complex workflow that enables collaboration among multiple agents. In the most common setting, an orchestration agent coordinates a team of specialized *worker agents*, allowing them to operate in parallel on different aspects of a complex research task. To illustrate the key principles and design considerations behind such a DR workflow, we introduce Anthropic Deep Research [391] as a representative example.

Table 2. Comparing output capabilities of contemporary representative DR systems, where the ■ indicates **supported capability**

System	Content Generation					Structured Output			Advanced		
	Text	Image	Audio	Video	Pres.	Table	JSON	Code	Chart	GUI	Cite
Gemini DeepResearch [132]	■	■				■			■		■
Grok DeepSearch [133]	■										■
OpenAI DeepResearch [134]	■	■				■		■	■	■	■
AutoGLM [135]	■										■
H2O.ai DeepResearch [141]	■	■	■		■	■	■	■	■		■
Skywork DeepResearch [136]	■			■	■						■
Perplexity DeepResearch [137]	■										■
Manus [138]	■				■	■			■		■
OpenManus [390]	■								■		■
OWL (CAMEL-AI) [377]	■				■	■			■		■
SunaAI [139]	■				■	■			■		
Alita [140]	■								■		

4.1.1. Deep Research System of Anthropic

Anthropic proposes a multi-agent Deep Research (DR) framework where a **lead orchestrator** coordinates multiple **worker agents** through structured, auditable interactions. The system transforms an open-ended research query into a complete workflow, from planning and delegation to synthesis and citation, under an explicit research budget controlling agent count, tool usage, and reasoning depth. We highlight several **core points** that enable the system's efficiency and reliability:

- *Query Stratification and Planning.* The orchestrator first analyzes the semantic type and difficulty of the input query (*e.g.*, depth-first vs. breadth-first) to determine research strategy and allocate a corresponding budget of agents, tool calls, and synthesis passes.
- *Delegation and Scaling.* Effort scales with complexity: from 1–2 agents for factual lookups to up to 10 or more for multi-perspective analyses, each assigned with clear quotas and stopping criteria to enable dynamic budget reallocation.
- *Task Decomposition and Prompt Specification.* The main query is decomposed into modular subtasks, each encoded as a structured prompt specifying objectives, output schema, citation policy, and fallback actions to ensure autonomy with accountability.
- *Tool Selection and Evidence Logging.* A central tool registry (*e.g.*, web fetch, PDF parsing, calculators) is used following freshness, verifiability, and latency rules. Agents record all tool provenance in an evidence ledger for traceable attribution.
- *Parallel Gathering and Interim Synthesis.* Worker agents operate concurrently while the orchestrator monitors coverage, resolves conflicts, and launches micro-delegations to close residual gaps or trigger deeper reasoning where needed.
- *Final Report and Attribution.* The orchestrator integrates verified findings into a coherent report, programmatically linking claims to sources and ensuring schema compliance, factual grounding, and transparent citation.

Overall, Anthropic's system exemplifies a scalable, interpretable multi-agent research paradigm that achieves high-quality synthesis through modular delegation, explicit budgeting, and verifiable reasoning.

 doi:10.20944/preprints202511.2077.v1

4.2. Supervised Fine-Tuning

Definition. Supervised fine-tuning (SFT) is a widely adopted approach that trains models to imitate desired behaviors using input–output pairs under a supervised learning objective. Within DR, SFT is commonly employed as the *cold start*, *e.g.,* a warm-up process, before online reinforcement learning [45,46,59,148,392]. It aims to endow agents with basic task-solving skills [158,393].

Since manual collection of expert trajectories is labor-intensive, costly, and difficult to scale, a key challenge lies in automatically constructing high-quality SFT datasets. This has been widely explored by prior work [394–397]. Below, we categorize representative work into two main paradigms: (i) *strong-to-weak distillation*, distilling correct task-solving trajectories from powerful LLMs (*e.g.,* GPT-5 and DeepSeek-V3.1) into smaller, weak models; and (ii) *iterative self-evolution*, iteratively fine-tuning the model on the dataset produced by itself, leading to a progressive improvement.

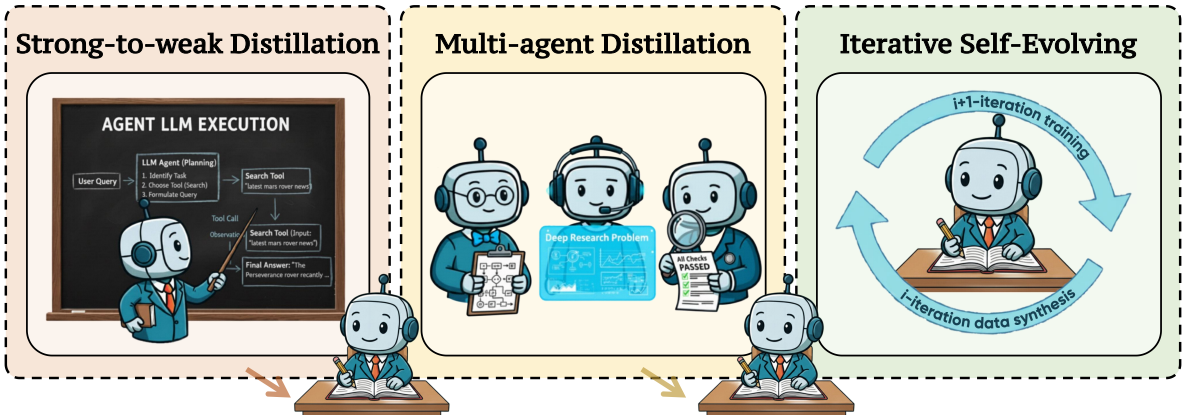

Figure 7. Comparisons among three types of data synthesis approaches, including: (i) Strong-to-Weak Distillation, (ii) Multi-Agent Distillation, and (iii) Iterative Self-Evolving. Each type is illustrated through the process of how agents perform tasks, learn, and refine their abilities.

4.2.1. Strong-to-Weak Distillation

Definition. Strong-to-weak distillation transfers high-quality decision trajectories from a powerful *teacher* system to smaller, weaker *student* models. Early work predominantly uses a single LLM-based agent to synthesize trajectories; more recent research employs multi-agent teacher systems to elicit more diverse, higher-complexity trajectories. We detail these two lines of work below.

Single-agent distillation. Representative systems instantiate this pipeline in various ways. WebDancer [142] provides the agent with search and click tools. A strong non-reasoning model generates short CoT, while a large reasoning model (LRM) generates long CoT. The agent learns from both, using rejection sampling for quality control. WebSailor [143] uses an expert LRM to generate action-observation trajectories, then reconstructs short CoT with a non-reasoning model, ensuring the final reasoning chain is compact enough for long-horizon tasks. WebShaper [144] uses search and visit tools in a ReAct-style trajectory. It performs 5 rollouts per task and filters out repeated or speculative answers using a reviewing LLM. WebThinker [12] augments SFT with policy-gradient refinement and WebSynthesis [145] leverages a learned world model to simulate virtual web environments and employs MCTS to synthesize diverse, controllable web interaction trajectories entirely offline.

Multi-agent distillation. Multi-agent distillation synthesizes training data using an agentic teacher system composed of specialized, collaborating agents (*e.g.,* a planner, a tool caller, and a verifier), with the goal of transferring emergent problem-solving behaviors into a single end-to-end student model [147,398]. This paradigm tends to produce diverse trajectories, richer tool-use patterns, and explicit self-correction signals.

A representative work is MaskSearch [148], which constructs a multi-agent pipeline that includes a planner, a query rewriter, and an observer, generating 58k verified chain-of-thought trajectories.

 doi:10.20944/preprints202511.2077.v1

Similarly, Chain-of-Agents [149] builds on the expert multi-agent system OAgents [399] to synthesize task-solving trajectories, and after a four-stage filtering pipeline that removes trivial or incorrect cases, it yields 16,433 high-quality trajectories for agent training. More recently, AgentFounder [150] propose the agentic continual pre-training, which scales up the data generation process by constructing large-scale planning traces, tool-invocation sequences, and step-by-step reasoning data.

Comparing Two Types of Distillation. Single-agent distillation provides a simple and easy-to-deploy pipeline, but it is limited by the bias of a single teacher model and the relatively shallow nature of its synthesized trajectories [400–403]. Such trajectories often emphasize token-level action sequences rather than higher-level reasoning, which can restrict the student model's generalization ability in complex tasks. In contrast, multi-agent distillation generates longer and more diverse trajectories that expand the action space to include strategic planning, task decomposition, iterative error correction, and self-reflection [404,405]. This broader behavioral coverage equips student models with stronger capabilities for multi-step and knowledge-intensive reasoning [149].

Despite these advantages, multi-agent distillation introduces notable trade-offs. The pipelines require careful system design, substantial inference cost, and dedicated infrastructure for logging and verification. Data quality can also be brittle as the system's sensitivity to prompting [406–408].

4.2.2. Iterative Self-Evolving

Definition. Iterative self-evolving data generation is an autonomous, cyclic process in which a model continuously generates new training data to fine-tune itself, progressively enhancing its capabilities [110,152,153,394].

Representative Work. Early evidence that large language models can improve themselves comes from self-training methods [151,394,409], where a model bootstraps from a small set of seed tasks to synthesize instruction–input–output triples, filters the synthetic data, and then fine-tunes itself on the resulting corpus. These approaches deliver substantial gains in instruction following with minimal human supervision. Yuan et al. [151] further introduces self-rewarding language models, in which the model generates its own rewards through LLM-as-a-Judge prompting. More recently, Zhao et al. [152] extends this idea to the zero-data regime by framing self-play as an autonomous curriculum. The model creates code-style reasoning tasks, solves them, and relies on an external code executor as a verifiable environment to validate both tasks and solutions. In the context of DR, EvolveSearch [153] iteratively selects high-performing rollouts (*i.e.,* task-solving trajectories) and re-optimizes the model on these data via supervised fine-tuning.

Advantages & Disadvantages. A key advantage of iterative self-evolving frameworks is their closed-loop design, where the model progressively improves its capabilities by tightly interleaving data generation with training. This autonomy enables scalable training without heavy reliance on external models or human annotations, and it allows exploration of data distributions that extend beyond handcrafted knowledge [151,394,410–412].

However, self-improvement also introduces significant risks. Previous studies have shown that, as iterations progress, distributional drift, reward hacking, and self-reinforcing errors may accumulate and degrade data quality, potentially leading to training collapse [413–415]. In addition, without robust validation mechanisms, the process may converge prematurely to narrow modes with limited performance ceilings [416–419].

4.3. End-to-End Agentic Reinforcement Learning

Definition. In this section, we dive into the application of end-to-end agentic reinforcement learning (RL) in DR, *i.e.,* using RL algorithms to incentivize DR agents that can flexibly plan, act, and generate a final answer. We start with a brief overview, including commonly used RL algorithms and reward design for optimizing DR systems. For a clear explanation, we provide a glossary table in Table 3

 doi:10.20944/preprints202511.2077.v1

to formally introduce the key variable in this section 4.3. Then we discuss two training practices: (i) *specific module optimization* and (ii) *entire pipeline optimization*.

Table 3. Summary of key notations used in proximal policy optimization and group-relative policy optimization algorithms.

Symbol	Definition	Description
π_θ	Current policy	Parameterized LLM policy that generates actions (tokens or sequences) conditioned on a given state.
$\pi_{\theta_{\text{old}}}$	Reference (old) policy	A frozen snapshot of the policy before the current update, used for computing probability ratios and ensuring stable optimization.
q	Input query	Input question or prompt to the agent.
o	Model output	Final answer produced by the policy model.
o^t	Action at step t	The token generated by the policy model conditioned on state s_t.
s_t	State at step t	Context of the policy model at time step t.
$\mathcal{R}(o\|\cdot)$	Reward function	Scalar score assigned to output o for the input query q.
$r_t(\theta)$	Probability ratio	Ratio between current and reference policy probabilities, computed as $\frac{\pi_\theta(o^t\|s_t)}{\pi_{\theta_{\text{old}}}(o^t\|s_t)}$.
ϵ	Clipping threshold	Stability constant that limits update magnitude in PPO or adds numerical robustness in GRPO.
$\mathcal{G}$	Response group	A collection of multiple sampled responses corresponding to the same query s_t in GRPO.
m	Group size	The number of candidate responses in a response group $\mathcal{G}$.
o_j	j-th response in group	The j-th sampled output candidate among the m responses in group $\mathcal{G}$.

4.3.1. Preliminary

RL algorithms in Deep Research. In DR, LLMs are trained to act as autonomous agents that generate comprehensive reports through complex query decomposition, multi-step reasoning, and extensive tool use. The primary RL algorithms used to train these agents include Proximal Policy Optimization (PPO) from OpenAI [259,420], Group Relative Policy Optimization (GRPO) from DeepSeek [332,421], and their variants [422].

Proximal Policy Optimization. PPO [259] is a clipped policy-gradient method that constrains updates within a trust region [423]. Given a current policy π_θ and a old policy $\pi_{\theta_{\text{old}}}$, the objective is to maximize the clipped surrogate:

$$L^{\text{PPO}}(\theta) = \mathbb{E}_t\big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\big], \tag{1}$$

$$r_t(\theta) = \frac{\pi_\theta(o^t \mid s_t)}{\pi_{\theta_{\text{old}}}(o^t \mid s_t)}. \tag{2}$$

where ϵ bounds the policy update and $\hat{A}_t$ is the estimated advantage. The advantage is computed using discounted returns or generalized advantage estimation (GAE) [424] as:

$$\hat{A}_t = \sum_{l=0}^{T-t} \gamma^l \cdot r_{t+l} + \gamma^{T-t+1} \cdot V_\phi(s_{T+1}) - V_\phi(s_t). \tag{3}$$

Here r_{t+l} denotes the immediate reward at time step $t+l$, $\gamma \in [0,1)$ is the discount factor balancing the importance of long-term and short-term returns; T is the terminal time step of the current trajectory (episode); s_{T+1} is the next state used for bootstrapping after termination, $V_\phi(s_t)$ is the value function

 doi:10.20944/preprints202511.2077.v1

predicted by the value network parameterized by ϕ. We define the empirical return $\hat{R}_t$ purely from rewards as:

$$\hat{R}_t = \sum_{l=0}^{T-t} \gamma^l r_{t+l}, \tag{4}$$

which represents the cumulative discounted rewards from time step t until the end of the episode. In PPO, the value function parameters ϕ are updated by minimizing the squared error between the predicted value and the empirical return:

$$\mathcal{L}^{\text{value}}(\phi) = \frac{1}{2}\mathbb{E}_t\Big[\big(V_\phi(s_t) - \hat{R}_t\big)^2\Big]. \tag{5}$$

Group Relative Policy Optimization. Group Relative Policy Optimization (GRPO) [332] extends PPO by normalizing rewards *within groups of responses* to the same query. Formally, given a group $\mathcal{G}$ of m responses $\{o_1, o_2, \ldots, o_m\}$ sampled for the same query s_t, each response is assigned a scalar reward R_j. The *group-relative advantage* for the j-th response is:

$$\hat{A}_j^{\mathcal{G}} = \frac{\mathcal{R}_j - \text{mean}(\{\mathcal{R}_i \mid i \in [m]\})}{\text{std}(\{\mathcal{R}_i \mid i \in [m]\}) + \epsilon}, \tag{6}$$

where $\text{mean}_{\mathcal{G}}$ and $\text{std}_{\mathcal{G}}$ denote the mean and standard deviation of rewards within group $\mathcal{G}$, and ϵ prevents numerical instability when the variance is small. The GRPO objective mirrors PPO's clipping mechanism but replaces $\hat{A}_t^{\mathcal{G}}$ with the group-relative advantage $\hat{A}_j^{\mathcal{G}}$:

$$\mathcal{L}^{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{|\mathcal{G}|}\sum_{j=1}^{|\mathcal{G}|}\min\Big\{\frac{\pi_\theta(o_j \mid q)}{\pi_{\theta_{\text{old}}}(o_j \mid q)}\hat{A}_j^{\mathcal{G}},\ \text{clip}\Big(\frac{\pi_\theta(o_j \mid q)}{\pi_{\theta_{\text{old}}}(o_j \mid q)}, 1-\epsilon, 1+\epsilon\Big)\hat{A}_j^{\mathcal{G}}\Big\}\right]. \tag{7}$$

Comparison between PPO and GRPO in Deep Research. In PPO, each sampled output is optimized using an advantage signal derived from a value model. While this approach is effective, its performance is highly reliant on accurate value estimation and requires additional resources for training the value model. In contrast, GRPO optimizes by contrasting each response against others within the same group. This shifts the focus to a relative-quality comparison among competing hypotheses, simplifying implementation while maintaining strong performance.

Reward Design in Deep Research Agents. During the RL training of DR agents, the reward model, denoted as $\mathcal{R}(\cdot)$, assesses the quality (*e.g.,* correctness) of the agents' outputs and produces scalar signals to enable policy optimization algorithms such as PPO and GRPO. Reward design takes a critical role in training LLM. There are two common reward design paradigms in DR systems, *i.e., rule-based rewards* and *LLM-as-judge rewards*.

- *Rule-based Rewards* $\mathcal{R}_{rule}(\cdot)$. Rule-based rewards are derived from deterministic, task-specific metrics such as Exact Match (EM) and F1 score [425]. In the context of research agents, EM is a commonly used binary score that indicates whether a generated answer perfectly matches a ground-truth string [45,69,273]. Alternatively, the F1 score (*i.e.,* the harmonic mean of precision and recall calculated over token overlap) is also used to reward outputs [37,53]. However, a key limitation of rule-based rewards is that they are primarily suited for tasks with well-defined, short-span ground truths (*e.g.,* a specific entity name) and struggle to evaluate multi-answer or open-ended questions effectively.
- *LLM-as-judge Rewards* $\mathcal{R}_{LLMs}(\cdot)$. The LLM-as-judge approach uses an external LLM to evaluate the quality of an agent's output and assign a scalar score based on a predefined rubric [426]. Formally, for an output o to an input query q, the reward assigned by an LLM judge ϕ can be formulated as:

$$\mathcal{R}_{\text{LLMs}}(o \mid q) = \mathbb{E}_{\text{criteria}\in\mathcal{C}}[\phi(o, q, \text{criteria})]$$

 doi:10.20944/preprints202511.2077.v1

where $\mathcal{C}$ is the set of evaluation criteria (*e.g.*, accuracy, completeness, citation quality, clarity, etc) and $\phi(\cdot)$ returns a scalar score for each criterion.

4.3.2. End-to-end Optimization of a Specific Module

Definition. End-to-end optimization of a specific module focuses on applying RL techniques to improve individual components within a DR system, such as the query planning, document ranking, or planning modules.

Representative work. Within DR, most existing work trains the query planner [43,427–429] while freezing the parameters, leaving components such as retrieval. MAO-ARAG [53] treats DR as a multi-turn process where a planning agent orchestrates sub-agents for information seeking. PPO propagates a holistic reward (*e.g.*, final F1 minus token and latency penalties) across all steps, enabling end-to-end learning of the trade-offs between answer quality and computational cost. AI-SearchPlanner [44] decouples a lightweight search planner from a frozen QA generator. PPO optimizes the planner with dual rewards: an outcome reward for improving answer quality and a process reward for reasoning rationality. A Pareto-regularized objective balances utility with real-world cost, guiding the planner on when to query or stop.

Advantages & Disadvantages. Single-module optimization usually focuses on training a single core component (*e.g.*, the planning module) while keeping the others fixed. Optimizing this critical module can improve the performance of a DR system by enabling more accurate credit assignment, more sophisticated algorithm design for the target module, and reduced training data and computational costs. However, this approach restricts the optimization space and may be inadequate when other frozen modules contain significant design or performance flaws.

4.3.3. End-to-end Optimization of an Entire Pipeline

Definition. End-to-end pipeline optimization involves jointly optimizing all components and processes from input to output (*e.g.*, query decomposition, search, reading, and report generation) to achieve the best overall performance across the DR workflow.

Representative work on Multi-Hop Search. Some work focuses on enhancing the capability of multi-hop search by training the entire DR systems end-to-end [17,46,47,154,430]. For example, Jin et al. [45,273] present Search-R1, the first work to formulate search-augmented reasoning as a fully observable Markov Decision Process and to optimize the entire pipeline via RL, containing query planning, retrieval, and extracting the final answer. By masking retrieved tokens in the policy-gradient loss, the model learns to autonomously decide when and what to search while keeping the training signal on its own generated tokens. Meanwhile, Song et al. [46] introduces R1-Searcher, a two-stage RL method in which the DR agent learns when to invoke external searches and how to use retrieved knowledge via outcome rewards. However, it has been observed that pure RL training often leads to over-reliance on external retrieval, resulting in over-searching [47]. To mitigate this issue, R1-Searcher++ [47] first cold-starts the DR agent via an SFT, then applies a knowledge-assimilation RL process to encourage the agent to internalize previously retrieved documents and avoid redundant retrievals.

Besides the above early effort, recent work extends the naive Search-R1 by integrating multi-reward signals or improving the training environment. *For the reward design*, R-Search [164] trains models to decide when to retrieve and how to integrate external knowledge in both single-hop and multi-hop question answering. The framework improves answer quality and evidence reliability by optimizing reasoning–search trajectories under a multi-objective reward design. *For the training environment*, ZeroSearch [157] and O^2-Searcher [160] simulate a search engine to develop retrieval capabilities without accessing the actual web, providing a more controllable setting for RL training. In contrast, DeepResearcher [17] operates directly in real-world web environments, learning to plan, search, verify, and autonomously answer open-domain questions.

 doi:10.20944/preprints202511.2077.v1

Besides a basic document retrieval tool, some work also integrates additional information-seeking tools, teaching the DR agent to flexibly combine them. MMSearch-R1 [154] stands out as the first RL-trained multimodal model that learns when and how to search the text or image from the web on demand. HierSearch [162] introduces a hierarchical DR framework for enterprise scenarios that involve both local and web knowledge sources. Other work [23,163,430] integrates knowledge graphs into DR agents to achieve efficient multi-hop reasoning.

Representative work on Long-Chain Web Search. Besides the relatively simple multi-hop QA tasks, more recent work also applies end-to-end pipeline optimization to address longer-chain web search problems. Prior works, such as WebDancer [142], WebSailor [143], and Kimi-K2 [155], have focused on advancing more intricate multi-hop tasks, including GAIA [178] and BrowseComp [184]. These approaches combine data synthesis with end-to-end reinforcement learning training, thereby enabling more extensive iterations in the DR process.

Furthermore, Gao et al. [156] presents ASearcher, which scales end-to-end RL to extreme long-horizon search. A fully asynchronous RL engine removes the 10-turn ceiling that plagued earlier systems, allowing trajectories of 40+ turns and 150k tokens to be optimized without blocking GPU updates. Coupled with an autonomous QA-synthesis agent that injects noise and fuzzes questions for difficulty, the whole pipeline is operated end-to-end, from synthetic data creation to multi-turn policy optimization. SimpleDeepSearcher [159] leverages real-web simulation and distilled SFT to deliver agentic search capability without heavy RL, yet stays fully compatible with lightweight RL refinement. WebAgent-R1 [158] and DeepDiver [161] are training web agents through end-to-end multi-turn RL algorithms.

In addition, some works [149,165–167] have studied Deep Research systems across multiple tool-calling scenarios. For example, Li et al. [149] introduces Chain-of-Agents (CoA), a novel paradigm that distills the capabilities of multi-agent systems into a single LLM. CoA enables native, end-to-end complex problem-solving by dynamically orchestrating multiple role-playing and tool agents within one model. Through multi-agent distillation and agentic RL, the authors train Agent Foundation Models (AFMs) in an end-to-end approach that achieves excellent performance on diverse web search benchmarks, while significantly reducing computational overhead compared to traditional multi-agent systems. Tool-Star [165] and ARPO [166] investigate how to effectively leverage tools in long-horizon tasks such as Deep Research, and use the GRPO algorithm to optimize the entire pipeline end-to-end. Additionally, AEPO [167] further improves rollout efficiency and, based on ARPO, optimizes both performance and efficiency for the end-to-end tool-use pipeline.

Advantages & Disadvantages. These end-to-end methods model the entire DR system as a multi-turn search process, achieving comprehensive optimization across reasoning, query rewriting, knowledge retrieval, tool invocation, and answer generation. This modeling and optimization approach is not only flexible but also allows for different objectives to be emphasized through the design of reward functions. However, these methods also have drawbacks, including sparse rewards, excessively long responses, and unstable training. Continuous optimization is needed to further enhance the effectiveness, stability, and efficiency of DR systems.

Takeaway

- **RL Algorithms**: PPO provides stable updates based on absolute rewards, while GRPO leverages group-relative advantages to reduce resource requirements.
- **Specific Module End-to-End Optimization**: Targets a critical component (*e.g.*, planner or searcher) for RL training, improving overall performance at lower cost, though limitations in other frozen modules cannot be addressed.
- **Entire Pipeline End-to-End Optimization**: Optimizes the full DR workflow, including retrieval, reasoning, tool use, and answer generation, yielding holistic gains but facing sparse rewards, long outputs, and training instability.

 doi:10.20944/preprints202511.2077.v1

5. Evaluation of Deep Research System

DR techniques have been applied to a wide range of downstream tasks, including healthcare [426], financial report generation [431], and survey generation [117]. In this section, we systematically review common benchmarks and evaluation protocols for DR systems across three representative scenarios: (i) *information seeking*, (ii) *report generation*, and (iii) *AI for research*. These scenarios reflect the most prevalent applications of DR agents. Each category poses distinct challenges, illuminating the limitations of current systems while offering practical insights to guide future advances.

5.1. Agentic Information Seeking

Evaluating the effectiveness of agentic information-seeking is a critical component of assessing DR systems. In DR scenarios, information seeking is not a single QA task but a multi-stage, iterative, and cross-domain process in which agents must continuously explore, reformulate, and synthesize information from diverse sources. To capture this complexity, benchmark design has evolved from early static single-hop retrieval tasks such as Natural Questions (NQ) [168] to dynamic web environments requiring multi-hop reasoning and complex interactions, *e.g.*, BrowseComp [184] and HotpotQA [171]. In this section, we review representative benchmarks and evaluation frameworks along two dimensions: query complexity and interaction environment complexity.

Table 4. Comprehensive overview of existing and emerging benchmarks for Deep Research Systems that focus on question answering scenarios.

Benchmark (with link)	Date	Aspect	Data size (train/dev/test)	Evaluation metrics
NQ	2019	QA	307373/7830/7842	Exact Match / F1 / Accuracy
SimpleQA	2024	QA	4,326	Exact Match / F1 / Accuracy
HotpotQA	2019	QA	90124 / 5617 / 5813	Exact Match / F1 / Accuracy
2WikiMultihopQA	2020	QA	167454/12576/12576	Exact Match / F1 / Accuracy
Bamboogle	2023	QA	8600	Exact Match / F1 / Accuracy
MultiHop-RAG	2024	QA	2556	Exact Match / F1 / Accuracy
MuSiQue	2022	QA	25K	Exact Match / F1 / Accuracy
GPQA	2023	QA	448	Accuracy
GAIA	2023	QA	450	Exact Match
BrowseComp	2025	QA	1266	Exact Match
BrowseComp-Plus	2025	QA	830	Accuracy / Recall / Search Call / Calibration Error
HLE	2025	QA	2500	Exact Match / Accuracy

5.1.1. Complex Queries

The evolution of benchmarks for agentic information seeking has closely followed the increasing complexity of query demands. Early benchmarks such as NQ [168], TriviaQA [169], and SimpleQA [170] established the foundation for question answering research. These datasets focused on single-hop queries, where answers could be retrieved with a single lookup or were already contained within the LLM's parameters. While such tasks provided a controlled starting point, they could not capture the reasoning and synthesis required in DR.

As research questions grew more complex, benchmarks evolved from simple fact retrieval to multi-step reasoning challenges. Multi-hop QA datasets assess an agent's ability to reformulate queries and build reasoning chains across documents. HotpotQA [171], one of the earliest and most widely used multi-hop datasets, requires reasoning across multiple Wikipedia articles using supporting facts to derive the answer. 2WikiMultihopQA [172] extends this by integrating information from two separate Wikipedia pages per question, emphasizing cross-document reasoning. Bamboogle [173] consists of 125 two-hop questions generated from random Wikipedia articles, testing the ability to decompose and reason over complex queries. MultiHop-RAG [174] is the RAG dataset designed specifically for multi-hop queries, categorizing questions into four types: inference, comparison, temporal, and null queries. MuSiQue [175] adopts a bottom-up approach, systematically pairing composable single-hop questions where one reasoning step depends on another. FRAMES [176] simulates realistic multi-document queries to evaluate an LLM's ability to retrieve relevant facts, reason accurately, and synthesize information into coherent responses. However, most of these benchmarks rely on

 doi:10.20944/preprints202511.2077.v1

structured, linear reasoning paths, which fall short of reflecting the inherent ambiguity and branching, non-linear exploration required in real-world research scenarios.

Recent benchmarks have begun to capture this growing complexity, placing greater emphasis on the in-depth and progressive exploration of complex topics. For instance, GPQA [177] is a graduate-level dataset in physics, chemistry, and biology that tests both domain experts and skilled non-experts, requiring extensive reasoning and problem-solving. Similarly, GAIA [178] provides 466 carefully designed questions that require multi-step reasoning, real-world knowledge retrieval, and complex generation. HLE [179] aims to be a comprehensive, fully closed academic benchmark across dozens of disciplines, including mathematics, humanities, and natural sciences, designed to advance reasoning skills. Its questions cannot be quickly answered through an online search. These recent datasets challenge agents to operate in environments that better reflect the ambiguity, branching evidence paths, and iterative synthesis characteristic of real-world DR systems.

Table 5. Comprehensive overview of existing and emerging benchmarks for Deep Research Systems that focus on more boarder scenarios.

Benchmark (with link)	Date	Aspect	Data size (train/dev/test)	Evaluation metrics
FRAMES	2024	QA	824	Exact Match / F1 / Accuracy
InfoDeepSeek	2025	QA	245	Accuracy / Utilization / Compactness
AssistantBench	2025	QA	214	F1 / Similarity
Mind2Web	2025	QA	2350	Accuracy / F1 / Step Success Rate
Mind2Web 2	2025	QA	130	Agent-as-a-Judge
Deep Research Bench	2025	QA	89	Precision / Recall / F1
DeepResearchGym	2025	QA	96,000	Report Relevance / Retrieval Faithfulness / Report Quality
WebArena	2024	Complex Task	812	Correctness
WebWalkerQA	2025	QA	680	Accuracy / Action Count
WideSearch	2025	QA	200	LLM Judge
MMInA	2025	Complex Task	1050	Success Rate
AutoSurvey	2024	Survey Generation	530,000	Citation Quality / Content Quality
ReportBench	2025	Survey Generation	600	Content Quality / Cited Statement / Non-Cited Statements
SurveyGen	2025	Survey Generation	4200	Topical Relevance / Academic Impact / Content Diversity
Deep Research Comparator	2025	Report Generation	176	BradleyTerry Score
DeepResearch Bench	2025	Report Generation	100	LLM Judge
ResearcherBench	2025	Report Generation	65	Rubric Assessment / Factual Assessment
LiveDRBench	2025	Report Generation	100	Precision / Recall / F1
PROXYQA	2025	Report Generation	100	LLM Judge
SCHOLARQABENCH	2025	Report Generation	2967	Accuracy / Citations / Rubrics
Paper2Poster	2025	Poster Generation	100	Visual Quality / Textual Coherence / VLM Judge
PosterGen	2025	Poster Generation	10	Poster Content / Poster Design
P2PInstruct	2025	Poster Generation	121	LLM Judge
Doc2PPT	2022	Slides Generation	6000	ROUGE / Figure Subsequence / Text-Figure Relevance
SLIDESBENCH	2025	Slides Generation	7000/0/585	Text / Image / Layout / Color / LLM Judge
Zenodo10K	2025	Slides Generation	10,448	Content / Design / Coherence
TSBench	2025	Slides Generation	379	Editing Success / Efficiency
AI Idea Bench	2025	Idea Generation	0/0/3495	LLM Judge
Scientist-Bench	2025	Idea Generation, Experimental Execution	0/0/52	LLM Judge, Human Judge
PaperBench	2025	Experimental Execution	0/0/20	LLM Judge
ASAP-Review	2021	Peer Review	0/0/8877	Human / ROUGE / BERTScore
DeepReview	2025	Peer Review	13378/0/1286	LLM Judge
SWE-Bench	2023	Software Engineering	0/0/500	Environment
ScienceWorld	2022	Scientific Discovery	3600/1800/1800	Environment
GPT-Simulator	2024	Scientific Discovery	0/0/76369	LLM Judge
DiscoveryWorld	2024	Scientific Discovery	0/0/120	LLM Judge
CORE-Bench	2024	Scientific Discovery	0/0/270	Environment
MLE	2024	Machine Learning Engineering	0/0/75	Environment
RE-Bench	2024	Machine Learning Engineering	0/0/7	Environment
DSBench	2024	Data Science	0/0/540	Environment
Spider2-V	2024	Data Science	0/0/494	Environment
DSEval	2024	Data Science	0/0/513	LLM Judge
UnivEARTH	2025	Earth Observation	0/0/140	Exact Match
Commit0	2024	Software Engineering	0/0/54	Unit test

5.1.2. Interaction Environment

As agent capabilities have advanced, evaluation based solely on static environments and fixed corpora is no longer sufficient. Consequently, a series of benchmarks has been developed to reflect the

 doi:10.20944/preprints202511.2077.v1

scale and dynamics of real-world web environments, requiring agents to interact with, navigate, and creatively explore web pages to obtain complex or hard-to-find information.

Some studies have incorporated browsing tools such as Google and Bing into benchmarks, enabling agents to directly retrieve and extract information from the live web. For example, InfoDeepSeek [180] and AssistantBench [181] present challenging tasks that require agents to integrate multiple search and browsing tools in real-time web environments, testing their ability to operate dynamically. Mind2Web [182] replaces the overly simplified, simulated environments common in other datasets with authentic, dynamic, and unpredictable real-world websites, providing complete records of user interactions, webpage snapshots, and network traffic. Its successor, Mind2Web 2 [183], was subsequently introduced to more rigorously evaluate agent-based search systems on realistic, long-horizon tasks that involve live web search and browsing. BrowseComp [184] and BrowseComp-Plus [185] demand persistent navigation to locate hard-to-find, entangled information across multiple sites. Moreover, DeepResearchBench [432] offers a large-scale RetroSearch environment that reduces task degradation and network randomness while evaluating LLM agents on complex real-world web research tasks. DeepResearchGym [186] complements this by providing an open-source sandbox with a reproducible search API and a rigorous evaluation protocol, promoting transparency and reproducibility in DR area.

Building on this trend, subsequent datasets have increasingly emphasized the authenticity and complexity of interactive environments. WebArena [187] provides a highly realistic and reproducible environment for language-guided agents, built from fully functional websites across four domains. WebWalkerQA [188] assesses LLMs' ability to systematically traverse website subpages and extract high-quality data through interactive actions such as clicking, specifically testing complex, multi-step web interactions. WideSearch [189] focuses on a critical yet under-evaluated task: requiring agents to thoroughly and accurately acquire all large-scale atomic information that meets a set of criteria and organize it into a structured output. MMInA [190] extends these challenges by providing a multi-hop, multi-modal benchmark for embodied agents performing integrated internet tasks on realistic, evolving websites, ensuring high realism and applicability to natural user tasks. Together, these benchmarks illustrate a clear trend: web-oriented evaluation environments are becoming increasingly human-like, visually grounded, diverse, complex, and realistic, pushing the limits of agentic information seeking and DR in dynamic, real-world settings.

5.2. Comprehensive Report Generation

Another critical dimension in evaluating DR systems is their capacity to generate comprehensive reports. Unlike single-point answers or brief summaries, comprehensive reports require systems to integrate information from multiple sources and modalities into structured, logically coherent, and broadly informative outputs. This process involves information aggregation, content organization, factual consistency verification, and clarity of expression, and is therefore regarded as a core indicator of a DR system's overall capability. Below, we introduce the relevant benchmarks by task type.

5.2.1. Survey Generation

A closely related task is survey generation, which involves producing structured overviews or syntheses of a specific scientific topic by aggregating information from diverse sources. Thanks to the clear citation structure provided by gold-standard references, survey generation has been widely used to evaluate the capabilities of DR systems. AutoSurvey [117] gathers arXiv articles of varying lengths and uses a multi-LLM-as-judge framework to evaluate survey generation in terms of speed, citation quality, and content quality. Moreover, ReportBench [191] is a systematic benchmark for evaluating research reports generated by large language models. It focuses on two key aspects: the relevance of citations and the reliability and accuracy of the report's statements. The evaluation corpus is constructed using high-quality survey papers published on arXiv as the gold standard. SurveyGen [192] is another survey-generation dataset, containing over 4,200 human-written surveys

 doi:10.20944/preprints202511.2077.v1

with chapter-level structure, cited references, and rich metadata. It enables comprehensive evaluation of content quality, citation accuracy, and structural consistency.

5.2.2. Long-Form Report Generation

Other benchmarks focus on different types of report generation tasks and introduce alternative evaluation frameworks. For example, Deep Research Comparator [193] provides a unified evaluation platform for DR agents, enabling systematic assessment of long-form reports and their intermediate reasoning processes through side-by-side comparison, fine-grained human feedback, and ranking mechanisms. DeepResearch Bench [16] is a benchmark of 100 PhD-level research tasks, introducing two evaluation methods for generated reports: a reference-based assessment of overall quality and a citation-based evaluation of retrieval accuracy. ResearcherBench [194] comprises 65 research questions focused on evaluating the capabilities of advanced agent systems on cutting-edge AI science problems, using an evaluation framework that combines rubric assessment and factual evaluation. LiveDR-Bench [195] is a benchmark for DR tasks, offering challenging science and world-event queries and evaluating systems via intermediate reasoning steps and factual sub-propositions. PROXYQA [196] uses human-designed meta-questions and corresponding proxy questions to indirectly assess knowledge coverage and information richness, providing an objective measure of long-form text generation quality. SCHOLARQABENCH [197] is a benchmark for scientific literature synthesis tasks in multiple formats, comprising 2,967 expert-written queries and 208 long-form answers across the field of computer science. Evaluating research reports is particularly challenging because there is no single gold-standard answer, and multiple valid perspectives exist for assessing quality. The diversity of acceptable content, reasoning approaches, and presentation styles makes it difficult to define objective metrics. As a result, most benchmarks rely on LLM-as-judge methods [433], leveraging large language models' reasoning and knowledge to provide scalable, consistent, and nuanced evaluations of content quality, factual accuracy, citation relevance, and structural coherence.

5.2.3. Poster Generation

Poster generation can be viewed as a highly condensed and visually structured variant of the comprehensive report generation task. Unlike multi-page reports, a scientific poster is typically a single-page, high-density summary that concisely presents the research motivation, methodology, results, and conclusions in a format that is both navigable and visually engaging. For DR systems, this task imposes unique challenges: not only must the system aggregate and synthesize information from multiple heterogeneous sources, such as research papers, notes, and presentation slides, but it must also transform that content into an effective visual layout. Evaluation of poster generation typically focuses on three main aspects: factual completeness, visual communication effectiveness, and readability. For example, Paper2Poster [198] focuses exclusively on AI research papers. The dataset comprises 100 paper-poster pairs, covering 280 distinct topics across subfields. A comprehensive evaluation framework is introduced, comprising four key dimensions: visual quality, text coherence, VLM-based quality judgment, and PaperQuiz, which is a novel metric designed to assess how effectively a poster communicates the core knowledge of the original paper. PosterGen [199] adopts a two-dimensional evaluation protocol, dividing the assessment into poster content and poster design. It introduces a VLM-based metric to evaluate key design aspects, including layout balance, readability, and aesthetic consistency. P2PInstruct [200] is a large-scale instruction dataset for paper-to-poster generation, containing over 30,000 high-quality instruction–response pairs. It covers the full pipeline from image element processing and text generation to final layout formatting.

5.2.4. Slides Generation

Slide generation represents another critical pathway for evaluating the comprehensive capabilities of DR systems. This task challenges a system not only to comprehend and summarize large volumes of heterogeneous information sources, but also to transform the distilled content into a structured, slide-based presentation format. The objectives of slide generation encompass information distillation,

 doi:10.20944/preprints202511.2077.v1

logical structuring, and presentation-oriented expression, making it a strong indicator of a system's ability to coordinate across multiple dimensions. Common benchmark tasks and datasets for this evaluation often involve generating slides from meeting transcripts, research papers, long-form reports, or collections of web documents. These benchmarks typically assess whether a model can maintain content integrity and factual accuracy while performing high-quality information selection and organization. Doc2PPT [201] collects paired documents and their corresponding slide decks from academic proceedings. It conducts detailed post-processing for evaluation, using metrics such as Slide-Level ROUGE, Longest Common Figure Subsequence, Text-Figure Relevance, and Mean Intersection over Union. For Example, SLIDESBENCH [202] is a benchmark comprising 7,000 training examples and 585 test examples, derived from 310 slide decks across 10 distinct domains. It supports two types of evaluation: reference-based evaluation, which measures similarity to target (gold) slides, and reference-free evaluation, which assesses the design quality of the generated slides independently. Zhang et al. introduced Zenodo10K [203], a new dataset collected from Zenodo, a platform hosting diverse, openly licensed artifacts across various domains. They also proposed PPTEval [203], an evaluation framework leveraging GPT-4.0 as the judge to assess presentation quality along three dimensions: content, design, and coherence. TSBench [204] is a benchmark dataset specifically designed to evaluate the slide editing capabilities of models and frameworks. It includes 379 distinct editing instructions along with the corresponding slide modifications.

5.3. AI for Research

AI for Research seeks to harness artificial intelligence to advance scientific discovery, either by automating processes or by assisting researchers in accelerating their work [21]. Its applications and corresponding benchmarks include (i) *idea generation*, (ii) *experimental execution*, (iii) *academic writing*, and (iv) *peer review*. Unlike report generation, research goes beyond producing extended outputs; it requires the creation of new perspectives, conclusions, and knowledge, thereby necessitating mechanisms for evaluating novelty.

5.3.1. Idea Generation

A key challenge in research lies in generating genuinely novel ideas and, more importantly, in reliably assessing their novelty. Such evaluation is typically conducted by human experts, but it remains difficult and resource-intensive. Existing approaches generally fall into two categories. The first is human- or LLM-based evaluation. Si et al. [205] recruited over 100 NLP researchers to evaluate the novelty of ideas generated by humans, LLMs, and human–LLM collaboration. However, this process is not easily scalable and proves difficult even for domain experts. Moreover, they investigated LLMs' ability to assess novelty, finding that LLM judgments show lower agreement with expert reviewers than human evaluations. Li et al. [206] and Gao et al. [211] leverage LLMs through direct prompting to evaluate novelty. To enhance the reliability of LLMs' judgments, Lu et al. [207] and Su et al. [208] integrate LLMs with the Semantic Scholar API and web access, enabling them to evaluate ideas against related literature. AI Idea Bench 2025 [209] provides a benchmark for quantifying and comparing ideas generated by LLMs. It incorporates 3,495 representative papers published in AI-related conferences after October 10, 2023, together with their corresponding inspiration papers. Furthermore, it introduces an evaluation framework to assess whether the ideas derived from inspiration papers are consistent with the ground-truth papers. The second category is density-based evaluation, which relies on the absolute local density in the semantic embedding space to measure novelty. Wang et al. [210] introduces the Relative Neighbor Density (RND) algorithm, which evaluates novelty by examining the distributional patterns of semantic neighbors rather than relying solely on absolute local density. Moreover, they construct large-scale semantic embedding databases for novelty assessment, encompassing more than 30 million publications across two distinct domains.

 doi:10.20944/preprints202511.2077.v1

5.3.2. Experimental Execution

Evaluation of experimental execution typically involves both objective and subjective assessments. Objective evaluation generally emphasizes the outcomes produced in specific environments, such as benchmark performance or compiler outputs. For example, Lu et al. [207] and Weng et al. [212] directly adopt the results of downstream tasks as evaluation metrics, while Tang et al. [213] leverages compiler outputs to assist LLMs in refining experiments and correcting code errors. Subjective evaluation involves leveraging either humans or LLMs to assess the quality of experimental designs. For example, Tang et al. [213] employs LLMs to compare code implementations with atomic research ideas, thereby verifying whether the code satisfies the intended requirements of the ideas. Starace et al. [214] employs LLMs to evaluate source code, documentation, and configuration files against human-designed rubrics to derive a final grade.

5.3.3. Academic Writing

The evaluation of academic writing differs substantially from general report generation. It requires not only factual accuracy, logical coherence, and clarity, but also alignment with underlying ideas and experimental results, proper integration of citations from related work, and effective visualization of findings. To capture these multifaceted criteria, Lu et al. [207] employ LLMs to assess writing along dimensions such as originality, quality, clarity, and significance. Similarly, Höpner et al. [215] train a domain-specific LLM to predict citation counts and review scores as proxies for paper quality, addressing the limitations of generic LLMs in academic evaluation. Building on this direction, Starace et al. [214] introduce PaperBench, a benchmark designed to assess AI agents' ability to replicate AI papers. The benchmark includes 20 ICML 2024 Spotlight and Oral papers, and evaluates replication quality using LLMs guided by manually constructed rubrics that hierarchically decompose each task into graded subtasks. Tang et al. [213] propose Scientist-Bench, a comprehensive benchmark built from top-cited papers published between 2022 and 2024 across 16 research areas and multiple expertise levels. It evaluates dimensions such as technical novelty, methodological rigor, empirical validation, and potential impact, closely reflecting the criteria used in the ICLR review process. To further push the boundaries of scientific evaluation, Xu et al. [194] present ResearcherBench, a more challenging benchmark for evaluating DR systems, which consists of 65 research questions carefully curated from real-world scientific contexts across 35 AI subfields. They also propose a dual evaluation framework that combines rubric-based assessment to evaluate the quality of insights with factual evaluation that measures citation faithfulness and evidence coverage.

5.3.4. Peer Review

AI for peer review seeks to leverage an AI agent to generate feedback on scientific papers. However, evaluating such feedback is challenging, as reviews are typically lengthy and inherently subjective. Yuan et al. [216] introduced ASAP-Review, a large-scale benchmark that collects 8,877 AI papers from ICLR (2017–2022) via OpenReview and NeurIPS papers (2016–2019) via the official proceedings, along with their corresponding reviews. The dataset is annotated across multiple dimensions, including Motivation, Originality, Soundness, Substance, Replicability, Clarity, and Comparison. To evaluate generated reviews, they employ automatic metrics such as ROUGE and BERTScore [434], as well as human judgments. Lu et al. [207] collect 500 ICLR 2022 papers from OpenReview to establish a benchmark for the peer review task, subsequently employing self-reflection, few-shot examples, and response ensembling with LLMs to assess the quality of the generated reviews. Weng et al. [217] introduces the REVIEW-5k dataset, which contains 782 test samples collected from ICLR 2024. Each sample includes the paper title, abstract, LaTeX or Markdown source, and the corresponding review comments. The dataset also provides structured review information, including summaries, strengths and weaknesses, clarification questions, and review scores. For evaluation, they employ Proxy Mean Squared Error (Proxy MSE) and Proxy Mean Absolute Error (Proxy MAE), which leverage multiple independent reviews of the same submission as unbiased estimators of its true

 doi:10.20944/preprints202511.2077.v1

rating [435]. Similarly, Gao et al. [218] constructed REVIEWER2, a dataset comprising 27,805 papers and reviews collected from CONLL-16, ACL-17, COLING-20, ARR-22, ICLR-17–23, and NeurIPS-16–22. Zhu et al. [219] introduces DeepReview-Bench, a dataset of 1.2K ICLR 2024–2025 submissions collected from OpenReview. The dataset includes textual reviewer assessments, interactive rebuttal-stage discussions, and standardized scoring information. For quantitative evaluation, they employ MAE, MSE, accuracy, F1, and Spearman correlation, while qualitative evaluation is conducted under the LLM-as-a-judge paradigm [433,436,437] across five dimensions: constructive value, analytical depth, plausibility, technical accuracy, and overall quality.

5.4. Software Engineering

In addition to the above scenarios, DR agents can also be applied to software engineering, representing a shift from assisting with isolated code snippets to autonomously executing complex software development tasks [438]. A pioneering work is SWE-Bench [439], a benchmark designed to evaluate whether AI agents can resolve real-world GitHub issues. Although SWE-Bench does not yet cover full end-to-end software development, it marks an important step toward bridging idealized benchmarks with practical scenarios. Meanwhile, DR agents have been deployed in a wide range of complex software engineering domains, including scientific discovery [440–446], machine learning experimentation [447–450], data science [451–453], earth observation [454], and software library completion [246,455].

6. Challenges and Outlook

6.1. Retrieval Timing

Although determining when to retrieve has become a standard feature of various DR systems, several fundamental challenges remain. Existing DR systems, such as Search-R1 Jin et al. [45], rely too heavily on answer correctness to guide the entire search pipeline and lack fine-grained guidance on when to retrieve, leading to both over-retrieval and under-retrieval Wu et al. [264]. Moreover, even with continued retrieval, the model may still produce an incorrect answer, and when no relevant evidence can be retrieved, generating an answer regardless risks misleading users, particularly in safety-critical domains such as healthcare and finance.

Future research could explore fine-grained reward designs that assess, at each step, whether the model lacks the knowledge needed to answer the question [264] and whether relevant documents can be retrieved [456]. Such signals would help determine when retrieval is necessary. Beyond deciding when to retrieve, the system should also evaluate whether the model's post-retrieval answer is correct and, after completing the entire process, estimate the uncertainty of the final output to avoid misleading users.

6.2. Memory Evolution

DR systems aim to mimic the research process of human experts by integrating autonomous planning, multi-source information acquisition, dynamic memory management, and deep knowledge synthesis. However, existing memory modules face significant challenges in fulfilling this vision. To develop more capable and DR systems, it is essential to re-examine the role of memory and identify future directions in personalization, structurization, adaptivity, and goal-driven optimization [340,457, 458].

6.2.1. Proactive Personalization Memory Evolution

Recap of previous work. Personalized memory in current systems often serves as a *passive knowledge buffer*, primarily designed to record user interaction histories and preferences for enhancing retrieval-based responses [459,460]. While effective for maintaining conversational consistency, this potentially limits the agent to a reactive stance [460]. The memory's primary function is to serve as a repository of past events, such as the fine-grained, timestamped interactions stored in episodic memory or

 doi:10.20944/preprints202511.2077.v1

the consolidated user traits in semantic memory [459,461]. Even advanced management techniques, such as the reflective mechanisms proposed by RMM [462], are chiefly focused on optimizing the organization and retrieval of this historical data to improve the relevance of future responses, rather than enabling forward-looking planning.

A necessary paradigm shift is emerging, moving from memory as a historical archive to memory as a dynamic, predictive user model [460]. To transition from mere assistants to true collaborators, future agents must leverage memory to engage in proactive reasoning. The foundation for such a model is a comprehensive, multi-dimensional user profile, as conceptualized in benchmarks like PersonaLens, which integrates demographics, detailed cross-domain preferences, and summaries of past interactions to form a holistic view of the user [463]. Early steps in this direction can be seen in goal-oriented systems like MemGuide, which employs proactive reasoning by using the user's task intent and analyzing missing information *slots* to strategically filter memories [464]. The ultimate vision is for future memory modules to empower agents as proactive partners by capturing not only explicit preferences but also implicit signals, such as communication styles and latent intents. The PaRT framework exemplifies this future, using its dynamic user profile to actively guide conversations by generating personalized new topics and retrieving real-time external information [460]. A blueprint for the underlying architecture can be found in systems like MIRIX, whose multi-component design could support diverse proactive functions; for instance, its Procedural Memory could store workflows to anticipate a user's next steps in a complex task [461]. By integrating these capabilities, the system can anticipate user needs, proactively acquire and present relevant information, and adapt its interaction style in real time, thus shifting from reactive responses to proactive planning for more effective, intuitive, personalized support [460].

6.2.2. Cognitive-Inspired Structured Memory Evolution

The predominant memory architecture in current systems (*e.g.*, vector stores of text chunks) follows a *flat* storage paradigm, which lacks the capacity to capture deep logical or relational structures between knowledge elements. This architectural deficiency fundamentally hinders complex multi-hop reasoning, as the system cannot traverse explicit relationships between concepts. Recent work has begun to address this by moving towards structured representations like knowledge graphs, where entities are explicitly linked by semantic relationships, thereby providing a scaffold for more sophisticated inference [107,108,250]. Moreover, memory is often treated as a static snapshot, making it incapable of addressing the temporal dynamics of knowledge. This is a critical failure point in real-world scenarios where information evolves. Pioneering work has introduced bi-temporal models into knowledge graphs [107], allowing memory to track not only when a fact was recorded but also the period during which it was valid in the real world, using non-destructive updates that preserve historical context [107,108].

A key future direction is to integrate these structured memory representations with dynamic, autonomous update mechanisms, drawing inspiration from cognitive science. Agents should be capable of autonomously transforming unstructured inputs into structured representations (*e.g.*, knowledge graphs [107,108], operator trees [465], or multi-faceted memory fragments [104]) in real time during interaction. Importantly, this is not a one-time conversion, but a continuous *stream-processing* procedure. As new information arrives, the memory structure must dynamically expand, prune, and reorganize itself [99,105,108]. This vision is partially realized in systems that employ agentic, cognitive-inspired operations such as `INSERT`, `FORGET`, and `MERGE` to refine memory content [99], or processes like *memory evolution*, in which new memories trigger updates and recontextualization of existing, linked memories [105]. The ultimate goal is to create a unified cognitive framework that addresses the dual challenges of representational depth and timeliness of knowledge. This framework would likely emulate the distinction between human episodic and semantic memory, a principle already explored in several advanced architectures [104,107,466,467], allowing an agent to both ground its knowledge in specific experiences and evolve a generalized, abstract understanding of the world.

 doi:10.20944/preprints202511.2077.v1

6.2.3. Goal-Driven Reinforced Memory Evolution

Existing strategies for memory retention are primarily heuristic-based, relying on static signals such as recency or semantic relevance [111,112]. However, these heuristics fail to guarantee that preserved memories are truly useful for achieving the final task goal, as they often ignore the interconnected memory cycle effect of storage, retrieval, and utilization [112]. A more powerful paradigm is to formulate memory management as a decision-making problem within a RL framework [111,115,354,468,469]. In this approach, the agent learns an optimal policy for memory operations, such as updating a fixed-length internal state [115,468] or executing structured commands like `ADD`, `UPDATE`, and `DELETE` on a memory store [111]. The learning process is guided solely by the reward from the final task outcome, forcing memory management to emerge as a goal-aligned, adaptive capability [111,115,468].

A key direction lies in extending this RL paradigm to jointly optimize the entire memory cycle, where agents learn not just to store information but to dynamically retrieve and utilize it through sophisticated strategies, such as multi-round reasoning [354] and experience reuse [112,469]. This goal is becoming increasingly practical due to two key advances. First, emerging frameworks enable policy learning for memory management at low cost and in real time, without requiring expensive LLM fine-tuning [469]. Second, the data efficiency of RL training makes this approach viable even in data-scarce domains [111]. However, despite these promising developments, a fundamental obstacle remains: the long-term credit assignment problem, which involves developing reliable algorithms to attribute a final outcome to a long sequence of intermediate memory decisions [468].

6.3. Instability in Training Algorithms

In DR systems, multiple rounds of interaction with the environment are required. Although RL algorithms such as PPO [259] and GRPO [332] exhibit stable behavior in single-turn scenarios, they often become unstable when extended to multi-turn settings. This instability typically appears as a gradual or abrupt drop in reward, the generation of invalid responses, and symptoms such as entropy collapse and gradient explosion [45,470–472]. These issues remain persistent challenges for training agentic RL systems. Below, we examine two newly emerging solutions and outline future directions for further study.

6.3.1. Existing Solutions

Filtering void turns. The first representative solution is proposed by Xue et al. [470], who identify *void turns* as a major cause of collapse in multi-turn RL. Void turns refer to responses that do not advance the task, such as fragmented text, repetitive content, or premature termination; and once produced, they propagate through later turns, creating a harmful feedback loop. These errors largely stem from the distribution shift between pre-training and multi-turn inference, where the model must process external tool outputs or intermediate signals that were not present during pre-training, increasing the chance of malformed generations. To address this, SimpleTIR [470] filters out trajectories containing void turns, effectively removing corrupted supervision and stabilizing multi-turn RL training.

Mitigating the Echo Trap. Wang et al. [471] identify the *Echo Trap* as a central cause of collapse in multi-turn RL. The Echo Trap refers to rapid policy homogenization, where the model abandons exploration and repeatedly produces conservative outputs that yield short-term rewards. Once this happens, reward variance and policy entropy drop sharply, forming a self-reinforcing degenerative loop. The root cause is a misalignment between reward-driven optimization and reasoning quality. In multi-turn settings, sparse binary rewards cannot distinguish coincidental success from genuine high-quality reasoning, encouraging reward hacking behaviors such as hallucinated reasoning or skipping essential steps. To address this, the proposed StarPO-S [471] uses uncertainty-based trajectory filtering to retain trajectories exhibiting meaningful exploration. This breaks the Echo Trap cycle and stabilizes multi-turn RL training.

 doi:10.20944/preprints202511.2077.v1

6.3.2. Future Directions

Beyond the solutions discussed above, we highlight two additional directions for achieving more stable agentic RL training.

Cold-start methods that preserve exploration. SFT is a practical cold-start strategy for multi-turn RL, yet it introduces a significant drawback: it rapidly reduces output entropy, constraining the model's ability to explore and develop new reasoning strategies [473]. A promising research direction is to design cold-start methods that improve initial task performance while maintaining exploratory behavior. Such techniques should aim to avoid early entropy collapse and preserve the model's capacity for innovation in multi-turn reasoning.

Denser and smoother reward design. Although StarPO-S [471] effectively mitigates training collapse in PPO-based multi-turn RL, its benefit is more limited for GRPO. The critic module inherent to PPO algorithm [259] naturally smooths reward signals, while GRPO relies on group-wise normalization, which makes it more sensitive to reward variance and extreme values. Developing denser, smoother, and more informative reward functions for multi-turn scenarios, especially for GRPO-style algorithms, remains an important direction for future research.

6.4. Evaluation of Deep Research System

Evaluation of DR generally falls into two complementary aspects: (i) the evaluation of agentic information-seeking capabilities and (ii) the evaluation of long-form generation [171,191]. Considerable progress has been made in the former, with benchmarks such as HotpotQA [171], GAIA [178]. The more recent Deep Research Bench [16,474] further provides increasingly complex and interactive settings for evaluating agents' abilities to retrieve, navigate, and synthesize information across dynamic web environments. However, reliably evaluating model-generated long-form outputs, especially research-style reports in response to open-ended and high-level queries, remains an open and pressing challenge [237]. Most existing approaches rely on LLM-as-a-Judge to directly evaluate general dimensions such as content factuality, structural coherence, and readability [194]. While effective for scalable comparisons, these evaluation strategies ignore crucial dimensions and are subject to several limitations.

6.4.1. Logical Evaluation

Since DR typically requires long-form context generation, maintaining logical coherence throughout the text is essential [475]. Existing studies suggest that while LLMs demonstrate strong capabilities for recognizing logical patterns, such as in summarization tasks or the detection of inconsistencies in short passages, their ability to create rigorous logical chains during DR remains uncertain [476]. In particular, when required to synthesize insights from multiple retrieved supporting documents, models often fail to consistently transform fragmented evidence into a logically connected narrative. The generated reasoning may contain gaps, abrupt leaps, or even circular justifications, compromising the argument's fidelity [477]. This limitation underscores a key challenge for DR: the task is not merely to produce fluent and factually plausible text, but to articulate insights that are logically well-founded and epistemically defensible [478].

Accordingly, robust logical evaluation emerges as a central challenge. However, most existing research on logical assessment remains narrowly scoped. Current benchmarks typically address limited logical tasks, such as solving symbolic logic puzzles, identifying entailment in short sentences, or handling deductive reasoning in synthetic settings [479,480]. While these tasks provide valuable insights into basic reasoning abilities, they fall short of capturing the complexities of long-form logical consistency. Specifically, they do not address whether models can sustain coherent argumentative structures across extended contexts, reconcile conflicting sources, or systematically avoid introducing unsupported claims. One potential approach is to design evaluation frameworks that assess coherence across multiple granularities (*e.g.*, sentence-level, paragraph-level, and document-level), capturing both local and global logical dependencies [481].

 doi:10.20944/preprints202511.2077.v1

6.4.2. Boundary between Novelty and Hallucination

In DR, progressing beyond faithful summarization toward the generation of genuinely novel hypotheses or perspectives is a central goal [482]. However, in practice, outputs that appear original may embed unverifiable claims, fabricated connections between sources, or spurious inferences lacking epistemic grounding [483]. This challenge is exacerbated in open-ended settings, where no single ground truth exists and retrieval broadens the hypothesis space, increasing the likelihood that superficially plausible but unsupported statements evade detection, especially by surface-level or style-sensitive evaluation methods [484]. Current practices often depend on density-based novelty scores or LLM-as-judge assessments of originality [210,485], yet these alone do not ensure verifiability or differentiate between creative recombination and unfounded speculation.

A potential solution is to differentiate between two types of novelty. Generative novelty refers to new combinations or perspectives, while deductive novelty refers to conclusions logically derived from known facts. To achieve this, novelty scoring can be combined with validity-checking mechanisms [486–488]. For example, researchers can pre-register testable claims along with verification plans, ensure that each claim is linked to clear sources, and systematically ablate sources to determine which are necessary or sufficient [489,490]. Additionally, inserting control examples or testing the system with false information can reveal how often it generates incorrect but seemingly original results [491]. Another useful method is to restrict the system to documents published before a certain cutoff date and then examine whether its outputs are later validated by subsequently published sources—providing insight into the independence and robustness of novel ideas [492,493].

6.4.3. Bias and Efficiency of LLM-as-Judge

LLM-as-Judge has become a mainstream approach for evaluating long-form model outputs. However, this practice introduces two major challenges. **The first challenge is bias.** LLM judges may prefer longer responses, be affected by answer ordering, reward particular writing styles, or favor systems that resemble themselves [433,436]. Such biases may reduce the robustness and fairness of existing evaluation protocols. **The second challenge is efficiency.** Large-scale pairwise evaluation is resource-intensive, especially when relying on paid APIs and applying costly comparison methods to long outputs [494]. These limitations motivate two directions for improvement: mitigating bias and improving efficiency.

Mitigating bias. Bias can be reduced by incorporating human evaluators for critical or ambiguous cases, providing a grounded reference for calibration [433]. Another direction is to fine-tune judge models using datasets that highlight diverse reasoning styles and explicit debiasing signals [494]. Such training may lessen systematic preferences for particular formats or linguistic patterns.

Improving efficiency. Efficiency can be improved by adopting open-source, general-purpose judge models, which reduce evaluation cost while offering greater transparency and reproducibility [495]. Further improvements may come from smarter candidate selection algorithms that focus on the most informative comparisons [496]. By lowering the number of required pairwise evaluations without sacrificing quality, such methods enable LLM-based evaluation in more resource-constrained settings.

7. Open Discussion: Deep Research to General Intelligence

As DR systems advance, they must navigate key challenges that bridge specialized task-solving with broader cognitive capabilities. This subsection examines three pivotal areas (*i.e.*, creativity, fairness, safety, and reliability) that may shape the development from current DR paradigms to AGI-level autonomy, ensuring these systems not only augment human inquiry but also foster equitable, innovative, and trustworthy ecosystems.

7.1. Creativity

Despite the considerable attention and rapid development in both academia and industry, Previous studies highlight fundamental limitations in LLM creativity based on next-token prediction

 doi:10.20944/preprints202511.2077.v1

[497]. While they excel at recombination [111,250], emotion [498,499], imitation [433,500], and logical reasoning [501,502], the question remains whether AI can evolve from these capabilities to achieve genuine innovation and novel concept generation. This transition may require mechanisms beyond statistical learning, drawing on psychological theories of human creativity, such as *insight* or *eureka moments*, which involve sudden restructuring of mental representations and are not easily explained by probabilistic models [503]. Some argue that hallucinations in AI could be interpreted as a form of creativity [497], potentially bridging this gap, but this perspective needs careful examination to distinguish between productive divergence and erroneous output.

7.2. Fairness

As noted in prior work [504], DR powered by autonomous agents may inadvertently inherit and amplify existing biases in academia. For example, they could favor mainstream fields, methodologies, or prominent researchers, thereby overlooking emerging interdisciplinary work or contributions from non-mainstream regions.r To mitigate this, such systems should incorporate built-in fairness frameworks that ensure comprehensive and impartial evaluation of all data, prevent the reinforcement of academic hierarchies, and provide equitable support to researchers from diverse backgrounds. A critical consideration is the impact of each agent's decision step on the overall fairness of outcomes: how much does bias in early steps shape subsequent decision spaces during interactions with the environment? Recent work [505] indicates that this cascading effect could limit exploration and perpetuate inequities if not addressed through debiasing techniques at every stage.

7.3. Safety and Reliability

Although some studies suggest that AI hallucinations can spark diversity [497,506], they also pose risks of disseminating serious academic errors. To enhance safety and reliability, Deep research system should ensure conclusions are supported by clear, traceable evidence chains; offer highly transparent reasoning processes to avoid "black-box" decisions; and implement robust validation mechanisms to curb the spread of hallucinated science [205,507]. These measures are essential for maintaining trust in AI-assisted research and preventing misinformation in scholarly pursuits.

8. Conclusions and Future Outlook

Deep research (DR) stands at the frontier of transforming large language models from passive responders into autonomous investigators capable of iterative reasoning, evidence synthesis, and verifiable knowledge creation. This survey consolidates recent advances in architectures, optimization methods, and evaluation frameworks, providing a unified roadmap for understanding and building future DR systems. By investigating relevant works, this survey facilitates future research and accelerates the advancement of DR systems toward more general, reliable, and interpretable intelligence. Given the rapid evolution of this field, we will continuously update this survey to encompass emerging paradigms such as multimodal reasoning, self-evolving memory, and agentic reinforcement learning. This effort aims to provide a comprehensive and up-to-date understanding of deep research systems.

References

1. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology* **2025**.
2. Gao, C.; Lan, X.; Li, N.; Yuan, Y.; Ding, J.; Zhou, Z.; Xu, F.; Li, Y. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications* **2024**.
3. Zhang, Z.; Chen, Z.; Li, M.; Tu, Z.; Li, X. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *arXiv preprint arXiv:2507.22844* **2025**.

 doi:10.20944/preprints202511.2077.v1

4. Shi, Z.; Ma, R.; Huang, J.t.; Ma, X.; Chen, X.; Wang, M.; Yang, Q.; Wang, Y.; Ye, F.; Chen, Z.; et al. Social Welfare Function Leaderboard: When LLM Agents Allocate Social Welfare. *arXiv preprint arXiv:2510.01164* **2025**.
5. He, Z.; Liang, T.; Xu, J.; Liu, Q.; Chen, X.; Wang, Y.; Song, L.; Yu, D.; Liang, Z.; Wang, W.; et al. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. *arXiv preprint arXiv:2504.11456* **2025**.
6. Zhang, Z.; Xu, J.; He, Z.; Liang, T.; Liu, Q.; Li, Y.; Song, L.; Liang, Z.; Zhang, Z.; Wang, R.; et al. Deeptheorem: Advancing llm reasoning for theorem proving through natural language and reinforcement learning. *arXiv preprint arXiv:2505.23754* **2025**.
7. G'omez-Rodr'iguez, C.; Williams, P. A Confederacy of Models: a Comprehensive Evaluation of LLMs on Creative Writing. *arXiv preprint arXiv:2310.08433* **2023**.
8. Hou, X.; Zhao, Y.; Liu, Y.; Yang, Z.; Wang, K.; Li, L.; Luo, X.; Lo, D.; Grundy, J.C.; Wang, H. Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Transactions on Software Engineering and Methodology* **2023**.
9. Jimenez, C.E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; Narasimhan, K. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770* **2023**.
10. Lee, C.; Xia, C.S.; Yang, L.; Huang, J.t.; Zhu, Z.; Zhang, L.; Lyu, M.R. UniDebugger: Hierarchical Multi-Agent Framework for Unified Software Debugging. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025.
11. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems* **2023**.
12. Li, X.; Jin, J.; Dong, G.; Qian, H.; Zhu, Y.; Wu, Y.; Wen, J.R.; Dou, Z. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776* **2025**.
13. Shi, Z.; Yan, L.; Yin, D.; Verberne, S.; de Rijke, M.; Ren, Z. Iterative self-incentivization empowers large language models as agentic searchers. *arXiv preprint arXiv:2505.20128* **2025**.
14. OpenAI. Deep Research System Card. Technical report, OpenAI, 2025.
15. Google. Deep research is now available on Gemini 2.5 Pro Experimental. https://blog.google/products/gemini/deep-research-gemini-2-5-pro-experimental/, 2025.
16. Du, M.; Xu, B.; Zhu, C.; Wang, X.; Mao, Z. DeepResearch Bench: A Comprehensive Benchmark for Deep Research Agents. *arXiv preprint arXiv:2506.11763* **2025**.
17. Zheng, Y.; Fu, D.; Hu, X.; Cai, X.; Ye, L.; Lu, P.; Liu, P. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160* **2025**.
18. Huang, Y.; Chen, Y.; Zhang, H.; Li, K.; Zhou, H.; Fang, M.; Yang, L.; Li, X.; Shang, L.; Xu, S.; et al. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096* **2025**.
19. Lyu, Y.; Zhang, X.; Yan, L.; de Rijke, M.; Ren, Z.; Chen, X. DeepShop: A Benchmark for Deep Research Shopping Agents. *arXiv preprint arXiv:2506.02839* **2025**.
20. Zhang, W.; Li, X.; Zhang, Y.; Jia, P.; Wang, Y.; Guo, H.; Liu, Y.; Zhao, X. Deep Research: A Survey of Autonomous Research Agents. *arXiv preprint arXiv:2508.12752* **2025**.
21. Chen, Q.; Yang, M.; Qin, L.; Liu, J.; Yan, Z.; Guan, J.; Peng, D.; Ji, Y.; Li, H.; Hu, M.; et al. AI4Research: A Survey of Artificial Intelligence for Scientific Research. *arXiv preprint arXiv:2507.01903* **2025**.
22. Xi, Y.; Lin, J.; Xiao, Y.; Zhou, Z.; Shan, R.; Gao, T.; Zhu, J.; Liu, W.; Yu, Y.; Zhang, W. A Survey of LLM-based Deep Search Agents: Paradigm, Optimization, Evaluation, and Challenges. *arXiv preprint arXiv:2508.05668* **2025**.
23. Luo, H.; E, H.; Chen, G.; Lin, Q.; Guo, Y.; Xu, F.; Kuang, Z.; Song, M.; Wu, X.; Zhu, Y.; et al. Graph-R1: Towards Agentic GraphRAG Framework via End-to-end Reinforcement Learning. *arXiv preprint arXiv:2507.21892* **2025**.
24. Shi, Z.; Gao, S.; Chen, X.; Feng, Y.; Yan, L.; Shi, H.; Yin, D.; Ren, P.; Verberne, S.; Ren, Z. Learning to Use Tools via Cooperative and Interactive Agents. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024.
25. Zhang, W.; Li, Y.; Bei, Y.Q.; Luo, J.; Wan, G.; Yang, L.; Xie, C.; Yang, Y.; Huang, W.C.; Miao, C.; et al. From Web Search towards Agentic Deep Research: Incentivizing Search with Reasoning Agents. *arXiv preprint arXiv:2506.18959* **2025**.
26. Sun, W.; Shi, Z.; Gao, S.; Ren, P.; de Rijke, M.; Ren, Z. Contrastive learning reduces hallucination in conversations. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2023.

 doi:10.20944/preprints202511.2077.v1

27. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Guo, Q.; Wang, M.; et al. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* **2023**.
28. Fan, W.; Ding, Y.; bo Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.S.; Li, Q. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* **2024**.
29. Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N.A.; Lewis, M. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350* **2022**.
30. Yao, S.; et al.. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations, 2023.
31. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Proceedings of the Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020.
32. Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* **2021**.
33. Packer, C.; Fang, V.; Patil, S.; Lin, K.; Wooders, S.; Gonzalez, J. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560* **2023**.
34. Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* **2022**.
35. Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495* **2023**.
36. Ma, X.; Gong, Y.; He, P.; Zhao, H.; Duan, N. Query rewriting in retrieval-augmented large language models. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
37. Chen, Y.; Yan, L.; Sun, W.; Ma, X.; Zhang, Y.; Wang, S.; Yin, D.; Yang, Y.; Mao, J. Improving retrieval-augmented generation through multi-agent reinforcement learning. *arXiv preprint arXiv:2501.15228* **2025**.
38. Jiang, P.; Lin, J.; Cao, L.; Tian, R.; Kang, S.; Wang, Z.; Sun, J.; Han, J. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223* **2025**.
39. Gong, P.; Zhu, F.; Yin, Y.; Dai, C.; Zhang, C.; Zheng, K.; Bao, W.; Mao, J.; Zhang, Y. CardRewriter: Leveraging Knowledge Cards for Long-Tail Query Rewriting on Short-Video Platforms. *arXiv preprint arXiv:2510.10095* **2025**.
40. Li, X.; Zhu, C.; Li, L.; Yin, Z.; Sun, T.; Qiu, X. Llatrieval: Llm-verified retrieval for verifiable generation. *arXiv preprint arXiv:2311.07838* **2023**.
41. Su, W.; Tang, Y.; Ai, Q.; Wu, Z.; Liu, Y. DRAGIN: dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081* **2024**.
42. Jiang, Z.; Sun, M.; Liang, L.; Zhang, Z. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. In Proceedings of the Companion Proceedings of the ACM on Web Conference 2025, 2025.
43. Jiang, P.; Xu, X.; Lin, J.; Xiao, J.; Wang, Z.; Sun, J.; Han, J. s3: You Don't Need That Much Data to Train a Search Agent via RL. *arXiv preprint arXiv:2505.14146* **2025**.
44. Mei, L.; Yang, Z.; Chen, C. AI-SearchPlanner: Modular Agentic Search via Pareto-Optimal Multi-Objective Reinforcement Learning. *arXiv preprint arXiv:2508.20368* **2025**.
45. Jin, B.; Zeng, H.; Yue, Z.; Yoon, J.; Arik, S.; Wang, D.; Zamani, H.; Han, J. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516* **2025**.
46. Song, H.; Jiang, J.; Min, Y.; Chen, J.; Chen, Z.; Zhao, W.X.; Fang, L.; Wen, J.R. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592* **2025**.
47. Song, H.; Jiang, J.; Tian, W.; Chen, Z.; Wu, Y.; Zhao, J.; Min, Y.; Zhao, W.X.; Fang, L.; Wen, J.R. R1-Searcher++: Incentivizing the Dynamic Knowledge Acquisition of LLMs via Reinforcement Learning. *arXiv preprint arXiv:2505.17005* **2025**.
48. Oh, M.; Kim, J.; Lee, N.; Seo, D.; Kim, T.; Lee, J. RAISE: Enhancing Scientific Reasoning in LLMs via Step-by-Step Retrieval. *arXiv preprint arXiv:2506.08625* **2025**.
49. Jiang, J.; Chen, J.; Li, J.; Ren, R.; Wang, S.; Zhao, W.X.; Song, Y.; Zhang, T. Rag-star: Enhancing deliberative reasoning with retrieval augmented verification and refinement. *arXiv preprint arXiv:2412.12881* **2024**.

 doi:10.20944/preprints202511.2077.v1

50. Zhu, D.; Shi, W.; Shi, Z.; Ren, Z.; Wang, S.; Yan, L.; Yin, D. Divide-Then-Aggregate: An Efficient Tool Learning Method via Parallel Tool Invocation. *arXiv preprint arXiv:2501.12432* **2025**.
51. Guo, M.; Zeng, Q.; Zhao, X.; Liu, Y.; Yu, W.; Du, M.; Chen, H.; Cheng, W. DeepSieve: Information Sieving via LLM-as-a-Knowledge-Router. *arXiv preprint arXiv:2507.22050* **2025**.
52. Guan, X.; Zeng, J.; Meng, F.; Xin, C.; Lu, Y.; Lin, H.; Han, X.; Sun, L.; Zhou, J. DeepRAG: Thinking to Retrieve Step by Step for Large Language Models. *arXiv preprint arXiv:2502.01142* **2025**.
53. Chen, Y.; Zhang, E.; Yan, L.; Wang, S.; Huang, J.; Yin, D.; Mao, J. MAO-ARAG: Multi-Agent Orchestration for Adaptive Retrieval-Augmented Generation. *arXiv preprint arXiv:2508.01005* **2025**.
54. Trivedi, H.; Balasubramanian, N.; Khot, T.; Sabharwal, A. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509* **2022**.
55. Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active retrieval augmented generation. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 7969–7992.
56. Ni, S.; Bi, K.; Guo, J.; Cheng, X. When Do LLMs Need Retrieval Augmentation? Mitigating LLMs' Overconfidence Helps Retrieval Augmentation. *arXiv preprint arXiv:2402.11457* **2024**.
57. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
58. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations, 2023.
59. Li, X.; Dong, G.; Jin, J.; Zhang, Y.; Zhou, Y.; Zhu, Y.; Zhang, P.; Dou, Z. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366* **2025**.
60. Ding, H.; Pang, L.; Wei, Z.; Shen, H.; Cheng, X. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612* **2024**.
61. Huanshuo, L.; Zhang, H.; Guo, Z.; Wang, J.; Dong, K.; Li, X.; Lee, Y.Q.; Zhang, C.; Liu, Y. CtrlA: Adaptive Retrieval-Augmented Generation via Inherent Control. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025.
62. Cheng, Q.; Li, X.; Li, S.; Zhu, Q.; Yin, Z.; Shao, Y.; Li, L.; Sun, T.; Yan, H.; Qiu, X. Unified active retrieval for retrieval augmented generation. *arXiv preprint arXiv:2406.12534* **2024**.
63. Yao, Z.; Qi, W.; Pan, L.; Cao, S.; Hu, L.; Liu, W.; Hou, L.; Li, J. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215* **2024**.
64. Robertson, S.; Zaragoza, H. *The Probabilistic Relevance Framework: BM25 and Beyond*; Now Publishers Inc., 2009.
65. Formal, T.; Lassance, C.; Piwowarski, B.; Clinchant, S. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086* **2021**.
66. Formal, T.; Piwowarski, B.; Clinchant, S. SPLADE: Sparse lexical and expansion model for first stage ranking. In Proceedings of the Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 2288–2292.
67. Khattab, O.; Zaharia, M. ColBERT: Efficient and Effective Passage Search via Late Interaction over BERT. In Proceedings of the SIGIR, 2020.
68. Santhanam, K.; Khattab, O.; Saad-Falcon, J.; Potts, C.; Zaharia, M. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488* **2021**.
69. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.t. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020.
70. Qu, Y.; Ding, Y.; Liu, J.; Liu, K.; Ren, R.; Zhao, W.X.; Dong, D.; Wu, H.; Wang, H. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191* **2020**.
71. Mathew, M.; Pal, U.; Jawahar, C. DocVQA: A Dataset for Document Visual Question Answering. In Proceedings of the WACV Workshops, 2021.
72. Cheng, Z.Q.; Dai, Q.; Li, S.; Sun, J.; Mitamura, T.; Hauptmann, A.G. ChartReader: A Unified Framework for Chart Derendering and Comprehension without Heuristic Rules. *arXiv preprint arXiv:2304.02173* **2023**.

 doi:10.20944/preprints202511.2077.v1

73. Dong, G.; Zhu, Y.; Zhang, C.; Wang, Z.; Wen, J.R.; Dou, Z. Understand what LLM needs: Dual preference alignment for retrieval-augmented generation. In Proceedings of the Proceedings of the ACM on Web Conference 2025, 2025, pp. 4206–4225.
74. Yu, Y.; Ping, W.; Liu, Z.; Wang, B.; You, J.; Zhang, C.; Shoeybi, M.; Catanzaro, B. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems* **2024**.
75. Qin, Z.; Jagerman, R.; Hui, K.; Zhuang, H.; Wu, J.; Yan, L.; Shen, J.; Liu, T.; Liu, J.; Metzler, D.; et al. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563* **2023**.
76. Sun, W.; Yan, L.; Ma, X.; Wang, S.; Ren, P.; Chen, Z.; Yin, D.; Ren, Z. Is ChatGPT good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542* **2023**.
77. Chen, Y.; Liu, Q.; Zhang, Y.; Sun, W.; Ma, X.; Yang, W.; Shi, D.; Mao, J.; Yin, D. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. In Proceedings of the Proceedings of the ACM on Web Conference 2025, 2025.
78. Wang, Y.; Ren, R.; Li, J.; Zhao, W.X.; Liu, J.; Wen, J.R. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering. *arXiv preprint arXiv:2402.17497* **2024**.
79. Xu, F.; Shi, W.; Choi, E. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408* **2023**.
80. Yoon, S.; Choi, E.; Kim, J.; Yun, H.; Kim, Y.; Hwang, S.w. Listt5: Listwise reranking with fusion-in-decoder improves zero-shot retrieval. *arXiv preprint arXiv:2402.15838* **2024**.
81. Wei, Z.; Chen, W.L.; Meng, Y. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. *arXiv preprint arXiv:2406.13629* **2024**.
82. Zhuang, S.; Ma, X.; Koopman, B.; Lin, J.; Zuccon, G. Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning. *arXiv preprint arXiv:2503.06034* **2025**.
83. Jin, J.; Zhu, Y.; Zhou, Y.; Dou, Z. Bider: Bridging knowledge inconsistency for efficient retrieval-augmented llms via key supporting evidence. *arXiv preprint arXiv:2402.12174* **2024**.
84. Liu, W.; Ma, X.; Sun, W.; Zhu, Y.; Li, Y.; Yin, D.; Dou, Z. ReasonRank: Empowering Passage Ranking with Strong Reasoning Ability. *arXiv preprint arXiv:2508.07050* **2025**.
85. Yu, W.; Zhang, H.; Pan, X.; Ma, K.; Wang, H.; Yu, D. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210* **2023**.
86. Wu, M.; Liu, Z.; Yan, Y.; Li, X.; Yu, S.; Zeng, Z.; Gu, Y.; Yu, G. RankCoT: Refining Knowledge for Retrieval-Augmented Generation through Ranking Chain-of-Thoughts. *arXiv preprint arXiv:2502.17888* **2025**.
87. Ge, T.; Hu, J.; Wang, L.; Wang, X.; Chen, S.Q.; Wei, F. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945* **2023**.
88. Rau, D.; Wang, S.; Déjean, H.; Clinchant, S. Context embeddings for efficient answer generation in rag. *arXiv preprint arXiv:2407.09252* **2024**.
89. Cheng, X.; Wang, X.; Zhang, X.; Ge, T.; Chen, S.Q.; Wei, F.; Zhang, H.; Zhao, D. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems* **2024**.
90. Guo, S.; Ren, Z. Dynamic Context Compression for Efficient RAG. *arXiv preprint arXiv:2507.22931* **2025**.
91. Cao, Z.; Cao, Q.; Lu, Y.; Peng, N.; Huang, L.; Cheng, S.; Su, J. Retaining key information under high compression ratios: Query-guided compressor for llms. *arXiv preprint arXiv:2406.02376* **2024**.
92. Tan, J.; Dou, Z.; Wang, W.; Wang, M.; Chen, W.; Wen, J.R. Htmlrag: Html is better than plain text for modeling retrieved knowledge in rag systems. In Proceedings of the Proceedings of the ACM on Web Conference 2025, 2025.
93. Chen, S.A.; Miculicich, L.; Eisenschlos, J.; Wang, Z.; Wang, Z.; Chen, Y.; Fujii, Y.; Lin, H.T.; Lee, C.Y.; Pfister, T. Tablerag: Million-token table understanding with language models. *Advances in Neural Information Processing Systems* **2024**.
94. Lu, J.; An, S.; Lin, M.; Pergola, G.; He, Y.; Yin, D.; Sun, X.; Wu, Y. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239* **2023**.
95. Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; Wang, Y. Memorybank: Enhancing large language models with long-term memory. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
96. Wang, Q.; Fu, Y.; Cao, Y.; Wang, S.; Tian, Z.; Ding, L. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing* **2025**, *639*, 130193.

 doi:10.20944/preprints202511.2077.v1

97. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative agents: Interactive simulacra of human behavior. In Proceedings of the Proceedings of the 36th annual acm symposium on user interface software and technology, 2023.
98. Zhu, X.; Chen, Y.; Tian, H.; Tao, C.; Su, W.; Yang, C.; Huang, G.; Li, B.; Lu, L.; Wang, X.; et al. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144* **2023**.
99. Liu, L.; Yang, X.; Shen, Y.; Hu, B.; Zhang, Z.; Gu, J.; Zhang, G. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719* **2023**.
100. Hu, C.; Fu, J.; Du, C.; Luo, S.; Zhao, J.; Zhao, H. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901* **2023**.
101. Anokhin, P.; Semenov, N.; Sorokin, A.; Evseev, D.; Kravchenko, A.; Burtsev, M.; Burnaev, E. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. *arXiv preprint arXiv:2407.04363* **2024**.
102. Jimenez Gutierrez, B.; Shu, Y.; Gu, Y.; Yasunaga, M.; Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems* **2024**.
103. Rezazadeh, A.; Li, Z.; Wei, W.; Bao, Y. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *arXiv preprint arXiv:2410.14052* **2024**.
104. Zhang, G.; Wang, B.; Ma, Y.; Zhao, D.; Yu, Z. Multiple Memory Systems for Enhancing the Long-term Memory of Agent. *arXiv preprint arXiv:2508.15294* **2025**.
105. Xu, W.; Mei, K.; Gao, H.; Tan, J.; Liang, Z.; Zhang, Y. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110* **2025**.
106. Ong, K.T.i.; Kim, N.; Gwak, M.; Chae, H.; Kwon, T.; Jo, Y.; Hwang, S.w.; Lee, D.; Yeo, J. Towards lifelong dialogue agents via timeline-based memory management. *arXiv preprint arXiv:2406.10996* **2024**.
107. Rasmussen, P.; Paliychuk, P.; Beauvais, T.; Ryan, J.; Chalef, D. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956* **2025**.
108. Chhikara, P.; Khant, D.; Aryan, S.; Singh, T.; Yadav, D. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413* **2025**.
109. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* **2023**, *36*.
110. Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* **2023**.
111. Yan, S.; Yang, X.; Huang, Z.; Nie, E.; Ding, Z.; Li, Z.; Ma, X.; Schütze, H.; Tresp, V.; Ma, Y. Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning. *arXiv preprint arXiv:2508.19828* **2025**.
112. Zhang, Z.; Dai, Q.; Li, R.; Bo, X.; Chen, X.; Dong, Z. Learn to Memorize: Optimizing LLM-based Agents with Adaptive Memory Framework. *arXiv preprint arXiv:2508.16629* **2025**.
113. Wei, R.; Cao, J.; Wang, J.; Kai, J.; Guo, Q.; Zhou, B.; Lin, Z. MLP Memory: Language Modeling with Retriever-pretrained External Memory. *arXiv preprint arXiv:2508.01832* **2025**.
114. Cao, J.; Wang, J.; Wei, R.; Guo, Q.; Chen, K.; Zhou, B.; Lin, Z. Memory Decoder: A Pretrained, Plug-and-Play Memory for Large Language Models. *arXiv preprint arXiv:2508.09874* **2025**.
115. Zhou, Z.; Qu, A.; Wu, Z.; Kim, S.; Prakash, A.; Rus, D.; Zhao, J.; Low, B.K.H.; Liang, P.P. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. *arXiv preprint arXiv:2506.15841* **2025**.
116. Gu, T.; Huang, K.; Luo, R.; Yao, Y.; Yang, Y.; Teng, Y.; Wang, Y. Meow: Memory supervised llm unlearning via inverted facts. *arXiv preprint arXiv:2409.11844* **2024**.
117. Wang, Y.; Guo, Q.; Yao, W.; Zhang, H.; Zhang, X.; Wu, Z.; Zhang, M.; Dai, X.; Wen, Q.; Ye, W.; et al. Autosurvey: Large language models can automatically write surveys. *Advances in neural information processing systems* **2024**.
118. Deng, B.; Wang, W.; Zhu, F.; Wang, Q.; Feng, F. Cram: Credibility-aware attention modification in llms for combating misinformation in rag. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 23760–23768.
119. Wang, H.; Prasad, A.; Stengel-Eskin, E.; Bansal, M. Retrieval-Augmented Generation with Conflicting Evidence. In Proceedings of the Second Conference on Language Modeling, 2025.
120. Wang, Y.; Ren, R.; Wang, Y.; Zhao, W.X.; Liu, J.; Wu, H.; Wang, H. Reinforced Informativeness Optimization for Long-Form Retrieval-Augmented Generation, 2025.

 doi:10.20944/preprints202511.2077.v1

121. Bai, Y.; Zhang, J.; Lv, X.; Zheng, L.; Zhu, S.; Hou, L.; Dong, Y.; Tang, J.; Li, J. LongWriter: Unleashing 10,000+ Word Generation from Long Context LLMs. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
122. Wu, Y.; Bai, Y.; Hu, Z.; Li, J.; Lee, R.K.W. SuperWriter: Reflection-Driven Long-Form Generation with Large Language Models, 2025, [arXiv:cs.CL/2506.04180].
123. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* **2022**.
124. Gu, H.; Li, D.; Dong, K.; Zhang, H.; Lv, H.; Wang, H.; Lian, D.; Liu, Y.; Chen, E. RAPID: Efficient Retrieval-Augmented Long Text Generation with Writing Planning and Information Discovery. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025.
125. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools, 2023. https://doi.org/10.48550/ARXIV.2302.04761.
126. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International Conference on Machine Learning, 2023.
127. Dai, W.; Li, J.; Li, D.; Tiong, A.M.H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; Hoi, S. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In Proceedings of the Proceedings of the 37th International Conference on Neural Information Processing Systems, 2023.
128. Zhu, D.; Chen, J.; Shen, X.; Li, X.; Elhoseiny, M. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
129. Jingwei, S.; Zeyu, Z.; Biao, W.; Yanjie, L.; Meng, F.; Ling, C.; Yang, Z. PresentAgent: Multimodal Agent for Presentation Video Generation. *arXiv preprint arXiv:2507.04036* **2025**.
130. Zheng, H.; Guan, X.; Kong, H.; Zheng, J.; Zhou, W.; Lin, H.; Lu, Y.; He, B.; Han, X.; Sun, L. PPTAgent: Generating and Evaluating Presentations Beyond Text-to-Slides, 2025, [arXiv:cs.AI/2501.03936].
131. Zhu, Z.; Lin, K.Q.; Shou, M.Z. Paper2Video: Automatic Video Generation from Scientific Papers. *arXiv preprint arXiv:2510.05096* **2025**.
132. LLC, G. Gemini Deep Research, 2024.
133. xAI. Grok DeepSearch, 2025.
134. OpenAI. Deep Research, 2025.
135. Liu, X.; Qin, B.; Liang, D.; Dong, G.; Lai, H.; Zhang, H.; Zhao, H.; Iong, I.L.; Sun, J.; Wang, J.; et al. Autoglm: Autonomous foundation agents for guis. *arXiv preprint arXiv:2411.00820* **2024**.
136. AI, S. Skywork-DeepResearch. https://github.com/SkyworkAI/Skywork-DeepResearch, 2025.
137. AI, P. Perplexity Deep Research, 2025.
138. Ltd.), M.A.B.E.P. Manus AI, 2025.
139. AI, K. Suna: Open-Source Generalist AI Agent, 2025.
140. Qiu, J.; Qi, X.; Zhang, T.; Juan, X.; Guo, J.; Lu, Y.; Wang, Y.; Yao, Z.; Ren, Q.; Jiang, X.; et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286* **2025**.
141. H2O.ai. H2O.ai Deep Research, 2025.
142. Wu, J.; Li, B.; Fang, R.; Yin, W.; Zhang, L.; Tao, Z.; Zhang, D.; Xi, Z.; Fu, G.; Jiang, Y.; et al. WebDancer: Towards Autonomous Information Seeking Agency. *arXiv preprint arXiv:2505.22648* **2025**.
143. Li, K.; Zhang, Z.; Yin, H.; Zhang, L.; Ou, L.; Wu, J.; Yin, W.; Li, B.; Tao, Z.; Wang, X.; et al. WebSailor: Navigating Super-human Reasoning for Web Agent. *arXiv preprint arXiv:2507.02592* **2025**.
144. Tao, Z.; Wu, J.; Yin, W.; Zhang, J.; Li, B.; Shen, H.; Li, K.; Zhang, L.; Wang, X.; Jiang, Y.; et al. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061* **2025**.
145. Gao, Y.; Ye, J.; Wang, J.; Sang, J. WebSynthesis: World-Model-Guided MCTS for Efficient WebUI-Trajectory Synthesis. *arXiv preprint arXiv:2507.04370* **2025**.
146. Hu, M.; Fang, T.; Zhang, J.; Ma, J.; Zhang, Z.; Zhou, J.; Zhang, H.; Mi, H.; Yu, D.; King, I. WebCoT: Enhancing Web Agent Reasoning by Reconstructing Chain-of-Thought in Reflection, Branching, and Rollback. *arXiv preprint arXiv:2505.20013* **2025**.
147. Tang, S.; Pang, X.; Liu, Z.; Tang, B.; Ye, R.; Jin, T.; Dong, X.; Wang, Y.; Chen, S. Synthesizing post-training data for llms through multi-agent simulation. *arXiv preprint arXiv:2410.14251* **2024**.

 doi:10.20944/preprints202511.2077.v1

148. Wu, W.; Guan, X.; Huang, S.; Jiang, Y.; Xie, P.; Huang, F.; Cao, J.; Zhao, H.; Zhou, J. MASKSEARCH: A Universal Pre-Training Framework to Enhance Agentic Search Capability. *arXiv preprint arXiv:2505.20285* **2025**.
149. Li, W.; Lin, J.; Jiang, Z.; Cao, J.; Liu, X.; Zhang, J.; Huang, Z.; Chen, Q.; Sun, W.; Wang, Q.; et al. Chain-of-Agents: End-to-End Agent Foundation Models via Multi-Agent Distillation and Agentic RL. *arXiv preprint arXiv:2508.13167* **2025**.
150. Su, L.; Zhang, Z.; Li, G.; Chen, Z.; Wang, C.; Song, M.; Wang, X.; Li, K.; Wu, J.; Chen, X.; et al. Scaling agents via continual pre-training. *arXiv preprint arXiv:2509.13310* **2025**.
151. Yuan, W.; Pang, R.Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; Weston, J. Self-rewarding language models. *arXiv preprint arXiv:2401.10020* **2024**, *3*.
152. Zhao, A.; Wu, Y.; Yue, Y.; Wu, T.; Xu, Q.; Lin, M.; Wang, S.; Wu, Q.; Zheng, Z.; Huang, G. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335* **2025**.
153. Zhang, D.; Zhao, Y.; Wu, J.; Li, B.; Yin, W.; Zhang, L.; Jiang, Y.; Li, Y.; Tu, K.; Xie, P.; et al. EvolveSearch: An Iterative Self-Evolving Search Agent. *arXiv preprint arXiv:2505.22501* **2025**.
154. Wu, J.; Deng, Z.; Li, W.; Liu, Y.; You, B.; Li, B.; Ma, Z.; Liu, Z. MMSearch-R1: Incentivizing LMMs to Search. *arXiv preprint arXiv:2506.20670* **2025**.
155. Team, K.; Bai, Y.; Bao, Y.; Chen, G.; Chen, J.; Chen, N.; Chen, R.; Chen, Y.; Chen, Y.; Chen, Y.; et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534* **2025**.
156. Gao, J.; Fu, W.; Xie, M.; Xu, S.; He, C.; Mei, Z.; Zhu, B.; Wu, Y. Beyond Ten Turns: Unlocking Long-Horizon Agentic Search with Large-Scale Asynchronous RL. *arXiv preprint arXiv:2508.07976* **2025**.
157. Sun, H.; Qiao, Z.; Guo, J.; Fan, X.; Hou, Y.; Jiang, Y.; Xie, P.; Zhang, Y.; Huang, F.; Zhou, J. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588* **2025**.
158. Wei, Z.; Yao, W.; Liu, Y.; Zhang, W.; Lu, Q.; Qiu, L.; Yu, C.; Xu, P.; Zhang, C.; Yin, B.; et al. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *arXiv preprint arXiv:2505.16421* **2025**.
159. Sun, S.; Song, H.; Wang, Y.; Ren, R.; Jiang, J.; Zhang, J.; Bai, F.; Deng, J.; Zhao, W.X.; Liu, Z.; et al. SimpleDeepSearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834* **2025**.
160. Mei, J.; Hu, T.; Fu, D.; Wen, L.; Yang, X.; Wu, R.; Cai, P.; Cai, X.; Gao, X.; Yang, Y.; et al. O2-Searcher: A Searching-based Agent Model for Open-Domain Open-Ended Question Answering. *arXiv preprint arXiv:2505.16582* **2025**.
161. Shi, W.; Tan, H.; Kuang, C.; Li, X.; Ren, X.; Zhang, C.; Chen, H.; Wang, Y.; Shang, L.; Yu, F.; et al. Pangu deep-diver: Adaptive search intensity scaling via open-web reinforcement learning. *arXiv preprint arXiv:2505.24332* **2025**.
162. Tan, J.; Dou, Z.; Yu, Y.; Cheng, J.; Ju, Q.; Xie, J.; Wen, J.R. HierSearch: A Hierarchical Enterprise Deep Search Framework Integrating Local and Web Searches. *arXiv preprint arXiv:2508.08088* **2025**.
163. Hao, C.; Feng, W.; Zhang, Y.; Wang, H. Dynasearcher: Dynamic knowledge graph augmented search agent via multi-reward reinforcement learning. *arXiv preprint arXiv:2507.17365* **2025**.
164. Zhao, Q.; Wang, R.; Xu, D.; Zha, D.; Liu, L. R-Search: Empowering LLM Reasoning with Search via Multi-Reward Reinforcement Learning. *arXiv preprint arXiv:2506.04185* **2025**.
165. Dong, G.; Chen, Y.; Li, X.; Jin, J.; Qian, H.; Zhu, Y.; Mao, H.; Zhou, G.; Dou, Z.; Wen, J.R. Tool-Star: Empowering LLM-Brained Multi-Tool Reasoner via Reinforcement Learning. *arXiv preprint arXiv:2505.16410* **2025**.
166. Dong, G.; Mao, H.; Ma, K.; Bao, L.; Chen, Y.; Wang, Z.; Chen, Z.; Du, J.; Wang, H.; Zhang, F.; et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849* **2025**.
167. Dong, G.; Bao, L.; Wang, Z.; Zhao, K.; Li, X.; Jin, J.; Yang, J.; Mao, H.; Zhang, F.; Gai, K.; et al. Agentic Entropy-Balanced Policy Optimization. *arXiv preprint arXiv:2510.14545* **2025**.
168. Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* **2019**.
169. Joshi, M.; Choi, E.; Weld, D.S.; Zettlemoyer, L. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Proceedings of the Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017.
170. Wei, J.; Karina, N.; Chung, H.W.; Jiao, Y.J.; Papay, S.; Glaese, A.; Schulman, J.; Fedus, W. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368* **2024**.

 doi:10.20944/preprints202511.2077.v1

171. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* **2018**.
172. Ho, X.; Nguyen, A.K.D.; Sugawara, S.; Aizawa, A. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In Proceedings of the Proceedings of the 28th International Conference on Computational Linguistics, 2020.
173. Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N.A.; Lewis, M. Measuring and Narrowing the Compositionality Gap in Language Models. In Proceedings of the The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
174. Tang, Y.; Yang, Y. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391* **2024**.
175. Trivedi, H.; Balasubramanian, N.; Khot, T.; Sabharwal, A. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics* **2022**, *10*.
176. Krishna, S.; Krishna, K.; Mohananey, A.; Schwarcz, S.; Stambler, A.; Upadhyay, S.; Faruqui, M. Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, 2025.
177. Rein, D.; Hou, B.L.; Stickland, A.C.; Petty, J.; Pang, R.Y.; Dirani, J.; Michael, J.; Bowman, S.R. Gpqa: A graduate-level google-proof q&a benchmark. In Proceedings of the First Conference on Language Modeling, 2024.
178. Mialon, G.; Fourrier, C.; Wolf, T.; LeCun, Y.; Scialom, T. Gaia: a benchmark for general ai assistants. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
179. Phan, L.; Gatti, A.; Han, Z.; Li, N.; Hu, J.; Zhang, H.; Zhang, C.B.C.; Shaaban, M.; Ling, J.; Shi, S.; et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249* **2025**.
180. Xi, Y.; Lin, J.; Zhu, M.; Xiao, Y.; Ou, Z.; Liu, J.; Wan, T.; Chen, B.; Liu, W.; Wang, Y.; et al. InfoDeepSeek: Benchmarking Agentic Information Seeking for Retrieval-Augmented Generation. *arXiv preprint arXiv:2505.15872* **2025**.
181. Yoran, O.; Amouyal, S.J.; Malaviya, C.; Bogin, B.; Press, O.; Berant, J. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711* **2024**.
182. Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; Su, Y. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* **2023**.
183. Gou, B.; Huang, Z.; Ning, Y.; Gu, Y.; Lin, M.; Qi, W.; Kopanev, A.; Yu, B.; Gutiérrez, B.J.; Shu, Y.; et al. Mind2Web 2: Evaluating Agentic Search with Agent-as-a-Judge. *arXiv preprint arXiv:2506.21506* **2025**.
184. Wei, J.; Sun, Z.; Papay, S.; McKinney, S.; Han, J.; Fulford, I.; Chung, H.W.; Passos, A.T.; Fedus, W.; Glaese, A. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516* **2025**.
185. Chen, Z.; Ma, X.; Zhuang, S.; Nie, P.; Zou, K.; Liu, A.; Green, J.; Patel, K.; Meng, R.; Su, M.; et al. BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent. *arXiv preprint arXiv:2508.06600* **2025**.
186. Coelho, J.; Ning, J.; He, J.; Mao, K.; Paladugu, A.; Setlur, P.; Jin, J.; Callan, J.; Magalhães, J.; Martins, B.; et al. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253* **2025**.
187. Zhou, S.; Xu, F.F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
188. Wu, J.; Yin, W.; Jiang, Y.; Wang, Z.; Xi, Z.; Fang, R.; Zhang, L.; He, Y.; Zhou, D.; Xie, P.; et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572* **2025**.
189. Wong, R.; Wang, J.; Zhao, J.; Chen, L.; Gao, Y.; Zhang, L.; Zhou, X.; Wang, Z.; Xiang, K.; Zhang, G.; et al. WideSearch: Benchmarking Agentic Broad Info-Seeking. *arXiv preprint arXiv:2508.07999* **2025**.
190. Tian, S.; Zhang, Z.; Chen, L.; Liu, Z. Mmina: Benchmarking multihop multimodal internet agents. *arXiv preprint arXiv:2404.09992* **2024**.
191. Li, M.; Zeng, Y.; Cheng, Z.; Ma, C.; Jia, K. ReportBench: Evaluating Deep Research Agents via Academic Survey Tasks, 2025.
192. Bao, T.; Nayeem, M.T.; Rafiei, D.; Zhang, C. SurveyGen: Quality-Aware Scientific Survey Generation with Large Language Models. *arXiv preprint arXiv:2508.17647* **2025**.

 doi:10.20944/preprints202511.2077.v1

193. Chandrahasan, P.; Jin, J.; Zhang, Z.; Wang, T.; Tang, A.; Mo, L.; Ziyadi, M.; Ribeiro, L.F.; Qiu, Z.; Dreyer, M.; et al. Deep research comparator: A platform for fine-grained human annotations of deep research agents. *arXiv preprint arXiv:2507.05495* **2025**.
194. Xu, T.; Lu, P.; Ye, L.; Hu, X.; Liu, P. Researcherbench: Evaluating deep ai research systems on the frontiers of scientific inquiry. *arXiv preprint arXiv:2507.16280* **2025**.
195. Java, A.; Khandelwal, A.; Midigeshi, S.; Halfaker, A.; Deshpande, A.; Goyal, N.; Gupta, A.; Natarajan, N.; Sharma, A. Characterizing Deep Research: A Benchmark and Formal Definition. *arXiv preprint arXiv:2508.04183* **2025**.
196. Tan, H.; Guo, Z.; Shi, Z.; Xu, L.; Liu, Z.; Feng, Y.; Li, X.; Wang, Y.; Shang, L.; Liu, Q.; et al. Proxyqa: An alternative framework for evaluating long-form text generation with large language models. *arXiv preprint arXiv:2401.15042* **2024**.
197. Asai, A.; He, J.; Shao, R.; Shi, W.; Singh, A.; Chang, J.C.; Lo, K.; Soldaini, L.; Feldman, S.; D'arcy, M.; et al. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199* **2024**.
198. Pang, W.; Lin, K.Q.; Jian, X.; He, X.; Torr, P. Paper2Poster: Towards Multimodal Poster Automation from Scientific Papers. *arXiv preprint arXiv:2505.21497* **2025**.
199. Zhang, Z.; Zhang, X.; Wei, J.; Xu, Y.; You, C. PosterGen: Aesthetic-Aware Paper-to-Poster Generation via Multi-Agent LLMs. *arXiv preprint arXiv:2508.17188* **2025**.
200. Sun, T.; Pan, E.; Yang, Z.; Sui, K.; Shi, J.; Cheng, X.; Li, T.; Huang, W.; Zhang, G.; Yang, J.; et al. P2P: Automated Paper-to-Poster Generation and Fine-Grained Benchmark. *arXiv preprint arXiv:2505.17104* **2025**.
201. Fu, T.J.; Wang, W.Y.; McDuff, D.; Song, Y. Doc2ppt: Automatic presentation slides generation from scientific documents. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 634–642.
202. Ge, J.; Wang, Z.Z.; Zhou, X.; Peng, Y.H.; Subramanian, S.; Tan, Q.; Sap, M.; Suhr, A.; Fried, D.; Neubig, G.; et al. Autopresent: Designing structured visuals from scratch. In Proceedings of the Proceedings of the Computer Vision and Pattern Recognition Conference, 2025.
203. Zheng, H.; Guan, X.; Kong, H.; Zheng, J.; Zhou, W.; Lin, H.; Lu, Y.; He, B.; Han, X.; Sun, L. Pptagent: Generating and evaluating presentations beyond text-to-slides. *arXiv preprint arXiv:2501.03936* **2025**.
204. Jung, K.; Cho, H.; Yun, J.; Yang, S.; Jang, J.; Choo, J. Talk to Your Slides: Language-Driven Agents for Efficient Slide Editing. *arXiv preprint arXiv:2505.11604* **2025**.
205. Si, C.; Yang, D.; Hashimoto, T. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109* **2024**.
206. Li, R.; Jing, L.; Han, C.; Zhou, J.; Du, X. Learning to generate research idea with dynamic control. *arXiv preprint arXiv:2412.14626* **2024**.
207. Lu, C.; Lu, C.; Lange, R.T.; Foerster, J.; Clune, J.; Ha, D. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292* **2024**.
208. Su, H.; Chen, R.; Tang, S.; Zheng, X.; Li, J.; Yin, Z.; Ouyang, W.; Dong, N. Two Heads Are Better Than One: A Multi-Agent System Has the Potential to Improve Scientific Idea Generation. *CoRR* **2024**, [2410.09403].
209. Qiu, Y.; Zhang, H.; Xu, Z.; Li, M.; Song, D.; Wang, Z.; Zhang, K. Ai idea bench 2025: Ai research idea generation benchmark. *arXiv preprint arXiv:2504.14191* **2025**.
210. Wang, Y.; Cui, M.; Jiang, A. Enabling ai scientists to recognize innovation: A domain-agnostic algorithm for assessing novelty. *arXiv preprint arXiv:2503.01508* **2025**.
211. Gao, X.; Zhang, Z.; Xie, M.; Liu, T.; Fu, Y. Graph of AI Ideas: Leveraging Knowledge Graphs and LLMs for AI Research Idea Generation. *arXiv preprint arXiv:2503.08549* **2025**.
212. Weng, Y.; Zhu, M.; Xie, Q.; Sun, Q.; Lin, Z.; Liu, S.; Zhang, Y. DeepScientist: Advancing Frontier-Pushing Scientific Findings Progressively. *arXiv preprint arXiv:2509.26603* **2025**.
213. Tang, J.; Xia, L.; Li, Z.; Huang, C. AI-Researcher: Autonomous Scientific Innovation. *arXiv preprint arXiv:2505.18705* **2025**.
214. Starace, G.; Jaffe, O.; Sherburn, D.; Aung, J.; Chan, J.S.; Maksin, L.; Dias, R.; Mays, E.; Kinsella, B.; Thompson, W.; et al. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848* **2025**.
215. Höpner, N.; Eshuijs, L.; Alivanistos, D.; Zamprogno, G.; Tiddi, I. Automatic Evaluation Metrics for Artificially Generated Scientific Research. *arXiv preprint arXiv:2503.05712* **2025**.
216. Yuan, W.; Liu, P.; Neubig, G. Can we automate scientific reviewing? *Journal of Artificial Intelligence Research* **2022**.

 doi:10.20944/preprints202511.2077.v1

217. Weng, Y.; Zhu, M.; Bao, G.; Zhang, H.; Wang, J.; Zhang, Y.; Yang, L. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816* **2024**.
218. Gao, Z.; Brantley, K.; Joachims, T. Reviewer2: Optimizing review generation through prompt generation. *arXiv preprint arXiv:2402.10886* **2024**.
219. Zhu, M.; Weng, Y.; Yang, L.; Zhang, Y. DeepReview: Improving LLM-based Paper Review with Human-like Deep Thinking Process. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, 2025.
220. Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; Deng, L. Ms marco: A human-generated machine reading comprehension dataset. In Proceedings of the International Conference on Learning Representations, 2016.
221. Roy, N.; Ribeiro, L.F.R.; Blloshmi, R.; Small, K. Learning When to Retrieve, What to Rewrite, and How to Respond in Conversational QA. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024.
222. Zaib, M.; Zhang, W.E.; Sheng, Q.Z.; Mahmood, A.; Zhang, Y. Conversational question answering: A survey. *Knowledge and Information Systems* **2022**, *64*, 3151–3195.
223. Tanjim, M.M.; In, Y.; Chen, X.; Bursztyn, V.S.; Rossi, R.A.; Kim, S.; Ren, G.J.; Muppala, V.; Jiang, S.; Kim, Y.; et al. Disambiguation in Conversational Question Answering in the Era of LLM: A Survey. *arXiv preprint arXiv:2505.12543* **2025**.
224. Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; Auli, M. ELI5: Long form question answering. *arXiv preprint arXiv:1907.09190* **2019**.
225. Chernyshevich, M. Core Intelligence at SemEval-2025 Task 8: Multi-hop LLM Agent for Tabular Question Answering. In Proceedings of the Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), 2025.
226. Nguyen, G.; Brugere, I.; Sharma, S.; Kariyappa, S.; Nguyen, A.T.; Lecue, F. Interpretable llm-based table question answering. *arXiv preprint arXiv:2412.12386* **2024**.
227. Masry, A.; Long, D.X.; Tan, J.Q.; Joty, S.; Hoque, E. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022; Muresan, S.; Nakov, P.; Villavicencio, A., Eds., 2022.
228. Zhao, Q.; Wang, J.; Zhang, Y.; Jin, Y.; Zhu, K.; Chen, H.; Xie, X. Competeai: Understanding the competition dynamics in large language model-based agents. *arXiv preprint arXiv:2310.17512* **2023**.
229. Vinogradova, A.; Vinogradov, V.; Radkevich, D.; Yasny, I.; Kobyzev, D.; Izmailov, I.; Yanchanka, K.; Doronin, R.; Doronichev, A. LLM-Based Agents for Competitive Landscape Mapping in Drug Asset Due Diligence. *arXiv preprint arXiv:2508.16571* **2025**.
230. Wang, M.; Colby, E.; Okwara, J.; Nagaraj Rao, V.; Liu, Y.; Monroy-Hernández, A. PolicyPulse: LLM-Synthesis Tool for Policy Researchers. In Proceedings of the Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, 2025.
231. Tang, Y.; Wang, Z.; Qu, A.; Yan, Y.; Wu, Z.; Zhuang, D.; Kai, J.; Hou, K.; Guo, X.; Zheng, H.; et al. ItiNera: Integrating spatial optimization with large language models for open-domain urban itinerary planning. *arXiv preprint arXiv:2402.07204* **2024**.
232. Chern, I.; Chern, S.; Chen, S.; Yuan, W.; Feng, K.; Zhou, C.; He, J.; Neubig, G.; Liu, P.; et al. FacTool: Factuality Detection in Generative AI–A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. *arXiv preprint arXiv:2307.13528* **2023**.
233. Min, S.; Krishna, K.; Lyu, X.; Lewis, M.; Yih, W.t.; Koh, P.; Iyyer, M.; Zettlemoyer, L.; Hajishirzi, H. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
234. Sun, H.; Cai, H.; Wang, B.; Hou, Y.; Wei, X.; Wang, S.; Zhang, Y.; Yin, D. Towards Verifiable Text Generation with Evolving Memory and Self-Reflection. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024.
235. Gao, T.; Yen, H.; Yu, J.; Chen, D. Enabling Large Language Models to Generate Text with Citations. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
236. Cao, B.; Ren, M.; Lin, H.; Han, X.; Zhang, F.; Zhan, J.; Sun, L. StructEval: Deepen and broaden large language model assessment via structured evaluation. *arXiv preprint arXiv:2408.03281* **2024**.
237. Wei, J.; Yang, C.; Song, X.; Lu, Y.; Hu, N.; Huang, J.; Tran, D.; Peng, D.; Liu, R.; Huang, D.; et al. Long-form factuality in large language models. *Advances in Neural Information Processing Systems* **2024**.

 doi:10.20944/preprints202511.2077.v1

238. Zhou, Y.; Liu, H.; Srivastava, T.; Mei, H.; Tan, C. Hypothesis Generation with Large Language Models. In Proceedings of the Proceedings of the 1st Workshop on NLP for Science (NLP4Science), 2024.
239. Nathani, D.; Madaan, L.; Roberts, N.; Bashlykov, N.; Menon, A.; Moens, V.; Budhiraja, A.; Magka, D.; Vorotilov, V.; Chaurasia, G.; et al. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499* **2025**.
240. Zhuang, Z.; Chen, J.; Xu, H.; Jiang, Y.; Lin, J. Large language models for automated scholarly paper review: A survey. *Information Fusion* **2025**, p. 103332.
241. Peng, Q.; Liu, H.; Xu, H.; Yang, Q.; Shao, M.; Wang, W. Review-llm: Harnessing large language models for personalized review generation. *arXiv preprint arXiv:2407.07487* **2024**.
242. Zhang, Y.; Chen, X.; Jin, B.; Wang, S.; Ji, S.; Wang, W.; Han, J. A comprehensive survey of scientific large language models and their applications in scientific discovery. *arXiv preprint arXiv:2406.10833* **2024**.
243. Shojaee, P.; Nguyen, N.H.; Meidani, K.; Farimani, A.B.; Doan, K.D.; Reddy, C.K. Llm-srbench: A new benchmark for scientific equation discovery with large language models. *arXiv preprint arXiv:2504.10415* **2025**.
244. Shojaee, P.; Meidani, K.; Gupta, S.; Farimani, A.B.; Reddy, C.K. Llm-sr: Scientific equation discovery via programming with large language models. *arXiv preprint arXiv:2404.18400* **2024**.
245. Wang, X.; Li, B.; Song, Y.; Xu, F.F.; Tang, X.; Zhuge, M.; Pan, J.; Song, Y.; Li, B.; Singh, J.; et al. Openhands: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741* **2024**.
246. Zhao, W.; Jiang, N.; Lee, C.; Chiu, J.T.; Cardie, C.; Gallé, M.; Rush, A.M. Commit0: Library generation from scratch. *arXiv preprint arXiv:2412.01769* **2024**.
247. Oche, A.J.; Folashade, A.G.; Ghosal, T.; Biswas, A. A systematic review of key retrieval-augmented generation (rag) systems: Progress, gaps, and future directions. *arXiv preprint arXiv:2507.18910* **2025**.
248. Neha, F.; Bhati, D. Traditional RAG vs. Agentic RAG: A Comparative Study of Retrieval-Augmented Systems. *Authorea Preprints* **2025**.
249. Zhou, S.; Xu, F.F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854* **2023**.
250. Xu, M.; Liang, G.; Chen, K.; Wang, W.; Zhou, X.; Yang, M.; Zhao, T.; Zhang, M. Memory-augmented query reconstruction for llm-based knowledge graph reasoning. *arXiv preprint arXiv:2503.05193* **2025**.
251. Hadi, M.U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M.B.; Akhtar, N.; Wu, J.; Mirjalili, S.; et al. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints* **2023**.
252. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A survey of large language models. *arXiv preprint arXiv:2303.18223* **2023**.
253. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* **2025**, *43*, 1–55.
254. Bang, Y.; Ji, Z.; Schelten, A.; Hartshorn, A.; Fowler, T.; Zhang, C.; Cancedda, N.; Fung, P. HalluLens: LLM Hallucination Benchmark. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025.
255. Cossio, M. A comprehensive taxonomy of hallucinations in large language models. *arXiv preprint arXiv:2508.01781* **2025**.
256. Shi, Z.; Zhang, S.; Sun, W.; Gao, S.; Ren, P.; Chen, Z.; Ren, Z. Generate-then-Ground in Retrieval-Augmented Generation for Multi-hop Question Answering. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024.
257. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems* **2023**.
258. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*.
259. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* **2017**.
260. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to algorithms*; MIT press, 2022.
261. Browne, C.B.; Powley, E.; Whitehouse, D.; Lucas, S.M.; Cowling, P.I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* **2012**.

 doi:10.20944/preprints202511.2077.v1

262. Kocsis, L.; Szepesvári, C. Bandit based monte-carlo planning. In Proceedings of the European conference on machine learning. Springer, 2006.
263. Zhang, G.; Niu, L.; Fang, J.; Wang, K.; Bai, L.; Wang, X. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180* **2025**.
264. Wu, P.; Zhang, M.; Zhang, X.; Du, X.; Chen, Z. Search Wisely: Mitigating Sub-optimal Agentic Searches By Reducing Uncertainty. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025.
265. Zhang, W.; Liao, J.; Li, N.; Du, K.; Lin, J. Agentic information retrieval. *arXiv preprint arXiv:2410.09713* **2024**.
266. Singhal, A.; et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* **2001**.
267. Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Chen, H.; Liu, Z.; Dou, Z.; Wen, J.R. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107* **2023**.
268. Yang, Q.; Ye, M.; Cai, Z.; Su, K.; Du, B. Composed image retrieval via cross relation network with hierarchical aggregation transformer. *IEEE Transactions on Image Processing* **2023**, *32*, 4543–4554.
269. Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; Wei, F. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* **2022**.
270. Shi, Y.; Li, S.; Wu, C.; Liu, Z.; Fang, J.; Cai, H.; Zhang, A.; Wang, X. Search and refine during think: Autonomous retrieval-augmented reasoning of llms. *arXiv e-prints* **2025**, pp. arXiv–2505.
271. He, J.; Fan, J.; Jiang, B.; Houine, I.; Roth, D.; Ribeiro, A. Self-give: Associative thinking from limited structured knowledge for enhanced large language model reasoning. *arXiv preprint arXiv:2505.15062* **2025**.
272. Shi, Z.; Gao, S.; Zhang, Z.; Chen, X.; Chen, Z.; Ren, P.; Ren, Z. Towards a unified framework for reference retrieval and related work generation. In Proceedings of the Findings of the Association for Computational Linguistics, 2023.
273. Jin, B.; Yoon, J.; Kargupta, P.; Arik, S.O.; Han, J. An Empirical Study on Reinforcement Learning for Reasoning-Search Interleaved LLM Agents. *arXiv preprint arXiv:2505.15117* **2025**.
274. Tao, Z.; Shen, H.; Li, B.; Yin, W.; Wu, J.; Li, K.; Zhang, Z.; Yin, H.; Ye, R.; Zhang, L.; et al. WebLeaper: Empowering Efficiency and Efficacy in WebAgent via Enabling Info-Rich Seeking. *arXiv preprint arXiv:2510.24697* **2025**.
275. Fang, R.; Cai, S.; Li, B.; Wu, J.; Li, G.; Yin, W.; Wang, X.; Wang, X.; Su, L.; Zhang, Z.; et al. Towards General Agentic Intelligence via Environment Scaling. *arXiv preprint arXiv:2509.13311* **2025**.
276. Ou, L.; Li, K.; Yin, H.; Zhang, L.; Zhang, Z.; Wu, X.; Ye, R.; Qiao, Z.; Xie, P.; Zhou, J.; et al. BrowseConf: Confidence-Guided Test-Time Scaling for Web Agents. *arXiv preprint arXiv:2510.23458* **2025**.
277. Chen, X.; Qiao, Z.; Chen, G.; Su, L.; Zhang, Z.; Wang, X.; Xie, P.; Huang, F.; Zhou, J.; Jiang, Y. AgentFrontier: Expanding the Capability Frontier of LLM Agents with ZPD-Guided Data Synthesis. *arXiv preprint arXiv:2510.24695* **2025**.
278. Xu, R.; Feng, Y.; Chen, H. Chatgpt vs. google: A comparative study of search performance and user experience. *arXiv preprint arXiv:2307.01135* **2023**.
279. Tang, X.; Qin, T.; Peng, T.; Zhou, Z.; Shao, D.; Du, T.; Wei, X.; Xia, P.; Wu, F.; Zhu, H.; et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229* **2025**.
280. Qiao, Z.; Chen, G.; Chen, X.; Yu, D.; Yin, W.; Wang, X.; Zhang, Z.; Li, B.; Yin, H.; Li, K.; et al. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309* **2025**.
281. Xu, Y.; et al.. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In Proceedings of the KDD, 2020.
282. Kim, G.; et al.. Donut: Document Understanding Transformer without OCR. In Proceedings of the ECCV, 2022.
283. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the International Conference on Machine Learning, 2021.
284. Zhai, X.; et al.. SigLIP: Scaling Up Visual Pre-training with Semantics-aware Contrastive Learning. *arXiv preprint arXiv:2303.15343* **2023**.
285. Li, J.; et al.. BLIP: Bootstrapping Language-Image Pre-training. In Proceedings of the International Conference on Machine Learning, 2022.
286. Yang, Q.; Ye, M.; Cai, Z.; Su, K.; Du, B. Composed image retrieval via cross relation network with hierarchical aggregation transformer. *IEEE Transactions on Image Processing* **2023**.
287. Zheng, H.; Wang, S.; Thomas, C.; Huang, L. Advancing Chart Question Answering with Robust Chart Component Recognition. *arXiv preprint arXiv:2407.21038* **2024**.

 doi:10.20944/preprints202511.2077.v1

288. Cormack, G.V.; Clarke, C.L.A.; Buettcher, S. Reciprocal Rank Fusion outperforms Condorcet and individual rank learning methods. In Proceedings of the SIGIR, 2009.
289. Meng, F.; Shao, W.; Lu, Q.; Gao, P.; Zhang, K.; Qiao, Y.; Luo, P. ChartAssistant: A Universal Chart Multimodal Language Model via Chart-to-Table Pre-training and Multitask Instruction Tuning. *arXiv preprint arXiv:2401.02384* **2024**.
290. Carbune, V.; Mansoor, H.; Liu, F.; Aralikatte, R.; Baechler, G.; Chen, J.; Sharma, A. Chart-based Reasoning: Transferring Capabilities from LLMs to VLMs (ChartPaLI-5B). *arXiv preprint arXiv:2403.12596* **2024**.
291. Zhang, L.; Hu, A.; Xu, H.; Yan, M.; Xu, Y.; Jin, Q.; Zhang, J.; Huang, F. TinyChart: Efficient Chart Understanding with Visual Token Merging and Program-of-Thoughts Learning. *arXiv preprint arXiv:2404.16635* **2024**.
292. Masry, A.; Kavehzadeh, P.; Do, X.L.; Hoque, E.; Joty, S. UniChart: A Universal Vision-language Pretrained Model for Chart Comprehension and Reasoning. *arXiv preprint arXiv:2305.14761* **2023**.
293. Tang, L.; Kim, G.; Zhao, X.; Lake, T.; Ding, W.; Yin, F.; Singhal, P.; Wadhwa, M.; Liu, Z.L.; Sprague, Z.; et al. ChartMuseum: Testing Visual Reasoning Capabilities of Large Vision-Language Models. *arXiv preprint arXiv:2505.13444* **2025**.
294. Xia, R.; Zhang, B.; Ye, H.; Yan, X.; Liu, Q.; Zhou, H.; Chen, Z.; Ye, P.; Dou, M.; Shi, B.; et al. ChartX & ChartVLM: A Versatile Benchmark and Foundation Model for Complicated Chart Reasoning. *arXiv preprint arXiv:2402.12185* **2024**.
295. Jeong, S.; Baek, J.; Cho, S.; Hwang, S.J.; Park, J.C. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403* **2024**.
296. Mallen, A.; Asai, A.; Zhong, V.; Das, R.; Khashabi, D.; Hajishirzi, H. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023.
297. Zhao, S.; Huang, Y.; Song, J.; Wang, Z.; Wan, C.; Ma, L. Towards understanding retrieval accuracy and prompt quality in rag systems. *arXiv preprint arXiv:2411.19463* **2024**.
298. Li, M.; Zhao, Y.; Zhang, W.; Li, S.; Xie, W.; Ng, S.K.; Chua, T.S.; Deng, Y. Knowledge boundary of large language models: A survey. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 5131–5157.
299. Zhang, Q.; Fu, Y.; Wang, Y.; Yan, L.; Wei, T.; Xu, K.; Huang, M.; Qiu, H. On the Self-awareness of Large Reasoning Models' Capability Boundaries. *arXiv preprint arXiv:2509.24711* **2025**.
300. Xiao, C.; Chan, H.P.; Zhang, H.; Aljunied, M.; Bing, L.; Moubayed, N.A.; Rong, Y. Analyzing LLMs' Knowledge Boundary Cognition Across Languages Through the Lens of Internal Representations. *arXiv preprint arXiv:2504.13816* **2025**.
301. Ren, R.; Wang, Y.; Qu, Y.; Zhao, W.X.; Liu, J.; Wu, H.; Wen, J.R.; Wang, H. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In Proceedings of the Proceedings of the 31st International Conference on Computational Linguistics, 2025.
302. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the International conference on machine learning. PMLR, 2017, pp. 1321–1330.
303. Desai, S.; Durrett, G. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892* **2020**.
304. Jiang, Z.; Araki, J.; Ding, H.; Neubig, G. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics* **2021**.
305. Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221* **2022**.
306. Si, C.; Gan, Z.; Yang, Z.; Wang, S.; Wang, J.; Boyd-Graber, J.; Wang, L. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150* **2022**.
307. Kuhn, L.; Gal, Y.; Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664* **2023**.
308. Duan, J.; Cheng, H.; Wang, S.; Zavalny, A.; Wang, C.; Xu, R.; Kailkhura, B.; Xu, K. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. *arXiv preprint arXiv:2307.01379* **2023**.
309. Fomicheva, M.; Sun, S.; Yankovskaya, L.; Blain, F.; Guzmán, F.; Fishel, M.; Aletras, N.; Chaudhary, V.; Specia, L. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics* **2020**, *8*, 539–555.

 doi:10.20944/preprints202511.2077.v1

310. Manakul, P.; Liusie, A.; Gales, M.J. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896* **2023**.
311. Zhang, J.; Li, Z.; Das, K.; Malin, B.A.; Kumar, S. SAC3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. *arXiv preprint arXiv:2311.01740* **2023**.
312. Azaria, A.; Mitchell, T. The internal state of an LLM knows when it's lying. *arXiv preprint arXiv:2304.13734* **2023**.
313. Su, W.; Wang, C.; Ai, Q.; Hu, Y.; Wu, Z.; Zhou, Y.; Liu, Y. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448* **2024**.
314. Chen, C.; Liu, K.; Chen, Z.; Gu, Y.; Wu, Y.; Tao, M.; Fu, Z.; Ye, J. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. *arXiv preprint arXiv:2402.03744* **2024**.
315. Wang, Y.; Li, H.; Zou, H.; Zhang, J.; He, X.; Li, Q.; Xu, K. Hidden question representations tell non-factuality within and across large language models. *arXiv preprint arXiv:2406.05328* **2024**.
316. Ni, S.; Bi, K.; Guo, J.; Yu, L.; Bi, B.; Cheng, X. Towards Fully Exploiting LLM Internal States to Enhance Knowledge Boundary Perception. *arXiv preprint arXiv:2502.11677* **2025**.
317. Ni, S.; Bi, K.; Guo, J.; Tang, M.; Wu, J.; Han, Z.; Cheng, X. Annotation-Efficient Universal Honesty Alignment. *arXiv preprint arXiv:2510.17509* **2025**.
318. Lin, S.; Hilton, J.; Evans, O. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research* **2022**.
319. Yin, Z.; Sun, Q.; Guo, Q.; Wu, J.; Qiu, X.; Huang, X. Do Large Language Models Know What They Don't Know? *arXiv preprint arXiv:2305.18153* **2023**.
320. Tian, K.; Mitchell, E.; Zhou, A.; Sharma, A.; Rafailov, R.; Yao, H.; Finn, C.; Manning, C.D. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975* **2023**.
321. Xiong, M.; Hu, Z.; Lu, X.; Li, Y.; Fu, J.; He, J.; Hooi, B. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063* **2023**.
322. Zhang, H.; Diao, S.; Lin, Y.; Fung, Y.; Lian, Q.; Wang, X.; Chen, Y.; Ji, H.; Zhang, T. R-tuning: Instructing large language models to say 'I don't know'. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2024.
323. Yang, Y.; Chern, E.; Qiu, X.; Neubig, G.; Liu, P. Alignment for honesty. *arXiv preprint arXiv:2312.07000* **2023**.
324. Yoran, O.; Wolfson, T.; Ram, O.; Berant, J. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558* **2023**.
325. Wu, J.; Cai, H.; Yan, L.; Sun, H.; Li, X.; Wang, S.; Yin, D.; Gao, M. Pa-rag: Rag alignment via multi-perspective preference optimization. *arXiv preprint arXiv:2412.14510* **2024**.
326. Fang, F.; Bai, Y.; Ni, S.; Yang, M.; Chen, X.; Xu, R. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. *arXiv preprint arXiv:2405.20978* **2024**.
327. Jin, B.; Yoon, J.; Han, J.; Arik, S.O. Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2024.
328. Xiao, S.; Liu, Z.; Zhang, P.; Muennighoff, N.; Lian, D.; Nie, J.Y. C-pack: Packed resources for general chinese embeddings. In Proceedings of the Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval, 2024, pp. 641–649.
329. Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; Grave, E. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* **2021**.
330. Izacard, G.; Grave, E. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* **2020**.
331. Shengyu, Z.; Linfeng, D.; Xiaoya, L.; Sen, Z.; Xiaofei, S.; Shuhe, W.; Jiwei, L.; Hu, R.; Tianwei, Z.; Wu, F.; et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792* **2023**.
332. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* **2024**.
333. Wang, L.; Yang, N.; Wei, F. Learning to Retrieve In-Context Examples for Large Language Models. In Proceedings of the Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, 2024.
334. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* **2023**.

 doi:10.20944/preprints202511.2077.v1

335. Zhu, K.; Feng, X.; Du, X.; Gu, Y.; Yu, W.; Wang, H.; Chen, Q.; Chu, Z.; Chen, J.; Qin, B. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. *arXiv preprint arXiv:2406.01549* **2024**.
336. Mu, J.; Li, X.; Goodman, N. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems* **2023**.
337. Chevalier, A.; Wettig, A.; Ajith, A.; Chen, D. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788* **2023**.
338. Wu, Y.; Liang, S.; Zhang, C.; Wang, Y.; Zhang, Y.; Guo, H.; Tang, R.; Liu, Y. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv preprint arXiv:2504.15965* **2025**.
339. Du, Y.; Huang, W.; Zheng, D.; Wang, Z.; Montella, S.; Lapata, M.; Wong, K.F.; Pan, J.Z. Rethinking memory in ai: Taxonomy, operations, topics, and future directions. *arXiv preprint arXiv:2505.00675* **2025**.
340. Jiang, X.; Li, F.; Zhao, H.; Qiu, J.; Wang, J.; Shao, J.; Xu, S.; Zhang, S.; Chen, W.; Tang, X.; et al. Long term memory: The foundation of ai self-evolution. *arXiv preprint arXiv:2410.15665* **2024**.
341. He, Z.; Lin, W.; Zheng, H.; Zhang, F.; Jones, M.W.; Aitchison, L.; Xu, X.; Liu, M.; Kristensson, P.O.; Shen, J. Human-inspired Perspectives: A Survey on AI Long-term Memory. *arXiv preprint arXiv:2411.00489* **2024**.
342. Zhang, Z.; Dai, Q.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Zhu, J.; Dong, Z.; Wen, J.R. A survey on the memory mechanism of large language model based agents. *ACM Transactions on Information Systems* **2024**.
343. Sun, W.; Lu, M.; Ling, Z.; Liu, K.; Yao, X.; Yang, Y.; Chen, J. Scaling Long-Horizon LLM Agent via Context-Folding. *arXiv preprint arXiv:2510.11967* **2025**.
344. Squire, L.R.; Genzel, L.; Wixted, J.T.; Morris, R.G. Memory consolidation. *Cold Spring Harbor perspectives in biology* **2015**, *7*, a021766.
345. Wang, Y.; Han, C.; Wu, T.; He, X.; Zhou, W.; Sadeq, N.; Chen, X.; He, Z.; Wang, W.; Haffari, G.; et al. Towards lifespan cognitive systems. *arXiv preprint arXiv:2409.13265* **2024**.
346. Zhao, Z.; Zhang, S.; Du, Y.; Liang, B.; Wang, B.; Li, Z.; Li, B.; Wong, K.F. Eventweave: A dynamic framework for capturing core and supporting events in dialogue systems. *arXiv preprint arXiv:2503.23078* **2025**.
347. Maekawa, A.; Kamigaito, H.; Funakoshi, K.; Okumura, M. Generative replay inspired by hippocampal memory indexing for continual language learning. In Proceedings of the Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, 2023.
348. Mehta, S.V.; Gupta, J.; Tay, Y.; Dehghani, M.; Tran, V.Q.; Rao, J.; Najork, M.; Strubell, E.; Metzler, D. Dsi++: Updating transformer memory with new documents. *arXiv preprint arXiv:2212.09744* **2022**.
349. Datar, M.; Immorlica, N.; Indyk, P.; Mirrokni, V.S. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Proceedings of the twentieth annual symposium on Computational geometry, 2004.
350. Malkov, Y.A.; Yashunin, D.A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* **2018**, *42*.
351. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* **2019**, *7*.
352. Sun, H.; Zeng, S. Hierarchical Memory for High-Efficiency Long-Term Reasoning in LLM Agents. *arXiv preprint arXiv:2507.22925* **2025**.
353. Wu, D.; Wang, H.; Yu, W.; Zhang, Y.; Chang, K.W.; Yu, D. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813* **2024**.
354. Long, L.; He, Y.; Ye, W.; Pan, Y.; Lin, Y.; Li, H.; Zhao, J.; Li, W. Seeing, Listening, Remembering, and Reasoning: A Multimodal Agent with Long-Term Memory. *arXiv preprint arXiv:2508.09736* **2025**.
355. Wang, J.; Zhao, R.; Wei, W.; Wang, Y.; Yu, M.; Zhou, J.; Xu, J.; Xu, L. ComoRAG: A Cognitive-Inspired Memory-Organized RAG for Stateful Long Narrative Reasoning. *arXiv preprint arXiv:2508.10419* **2025**.
356. Wang, S.; Zhu, Y.; Liu, H.; Zheng, Z.; Chen, C.; Li, J. Knowledge editing for large language models: A survey. *ACM Computing Surveys* **2024**.
357. Tack, J.; Kim, J.; Mitchell, E.; Shin, J.; Teh, Y.W.; Schwarz, J.R. Online adaptation of language models with a memory of amortized contexts. *Advances in Neural Information Processing Systems* **2024**.
358. Wang, Y.; Gao, Y.; Chen, X.; Jiang, H.; Li, S.; Yang, J.; Yin, Q.; Li, Z.; Li, X.; Yin, B.; et al. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624* **2024**.
359. Wang, P.; Li, Z.; Zhang, N.; Xu, Z.; Yao, Y.; Jiang, Y.; Xie, P.; Huang, F.; Chen, H. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems* **2024**.

 doi:10.20944/preprints202511.2077.v1

360. De Cao, N.; Aziz, W.; Titov, I. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164* **2021**.
361. Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; Manning, C.D. Fast model editing at scale. *arXiv preprint arXiv:2110.11309* **2021**.
362. Zhou, H.; Chen, Y.; Guo, S.; Yan, X.; Lee, K.H.; Wang, Z.; Lee, K.Y.; Zhang, G.; Shao, K.; Yang, L.; et al. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153* **2025**.
363. Shao, Y.; Li, L.; Dai, J.; Qiu, X. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158* **2023**.
364. Kim, Y.; Lee, H.; Shin, J.; Jung, K. Improving neural question generation using answer separation. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2019, Vol. 33, pp. 6602–6609.
365. Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; van den Driessche, G.; Lespiau, J.B.; Damoc, B.; Clark, A.; et al. Improving language models by retrieving from trillions of tokens, 2022.
366. Chen, L.; Tong, P.; Jin, Z.; Sun, Y.; Ye, J.; Xiong, H. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *Advances in Neural Information Processing Systems* **2024**, *37*, 37665–37691.
367. Yu, F.X.; Adam, G.; Bastian, N.D.; Lan, T. Optimizing prompt sequences using monte carlo tree search for LLM-based optimization. *arXiv preprint arXiv:2508.05995* **2025**.
368. Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Mittal, A. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In Proceedings of the Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018.
369. Ge, Z.; Wu, Y.; Chin, D.W.K.; Lee, R.K.W.; Cao, R. Resolving Conflicting Evidence in Automated Fact-Checking: A Study on Retrieval-Augmented LLMs, 2025.
370. Wang, H.; Prasad, A.; Stengel-Eskin, E.; Bansal, M. Retrieval-augmented generation with conflicting evidence. *arXiv preprint arXiv:2504.13079* **2025**.
371. Sun, J.; Zhong, X.; Zhou, S.; Han, J. DynamicRAG: Leveraging Outputs of Large Language Model as Feedback for Dynamic Reranking in Retrieval-Augmented Generation, 2025, [arXiv:cs.CL/2505.07233].
372. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
373. Luo, Y.; Shi, L.; Li, Y.; Zhuang, A.; Gong, Y.; Liu, L.; Lin, C. From intention to implementation: automating biomedical research via LLMs. *Science China Information Sciences* **2025**.
374. Yang, L.; Weng, Y. ResearStudio: A Human-Intervenable Framework for Building Controllable Deep-Research Agents. *arXiv preprint arXiv:2510.12194* **2025**.
375. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems* **2022**.
376. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In Proceedings of the International Conference on Learning Representations, 2023.
377. Hu, M.; Zhou, Y.; Fan, W.; Nie, Y.; Xia, B.; Sun, T.; Ye, Z.; Jin, Z.; Li, Y.; Chen, Q.; et al. OWL: Optimized Workforce Learning for General Multi-Agent Assistance in Real-World Task Automation, 2025. Accessed: 2025-11-13.
378. Weng, Y.; Zhu, M.; Bao, G.; Zhang, H.; Wang, J.; Zhang, Y.; Yang, L. CycleResearcher: Improving Automated Research via Automated Review. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
379. Shi, Z.; Gao, S.; Yan, L.; Feng, Y.; Chen, X.; Chen, Z.; Yin, D.; Verberne, S.; Ren, Z. Tool learning in the wild: Empowering language models as automatic tool agents. In Proceedings of the Proceedings of the ACM on Web Conference 2025, 2025, pp. 2222–2237.
380. Yi, G.; Nan, C.; Xiaoyu, Q.; Haoyang, L.; Danqing, S.; Jing, Z.; Qing, C.; and, Daniel, W. Urania: Visualizing Data Analysis Pipelines for Natural Language-Based Data Exploration. *arXiv preprint arXiv:2306.07760* **2023**.
381. Xie, Q.; Feng, Q.; Zhang, Y.; Feng, R.; Zhang, T.; Gao, S. ControlCap: Controllable Captioning via No-Fuss Lexicon. In Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024, pp. 8326–8330.
382. Xia, F.; Li, B.; Weng, Y.; He, S.; Liu, K.; Sun, B.; Li, S.; Zhao, J. MedConQA: Medical Conversational Question Answering System based on Knowledge Graphs. In Proceedings of the Proceedings of the 2022 Conference

 doi:10.20944/preprints202511.2077.v1

on Empirical Methods in Natural Language Processing: System Demonstrations; Che, W.; Shutova, E., Eds., 2022.
383. Victor, D. LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models. *arXiv preprint arXiv:2303.02927* **2023**.
384. Yuan, T.; Weiwei, C.; Dazhen, D.; Xinjing, Y.; Yurun, Y.; Haidong, Z.; Yingcai, W. ChartGPT: Leveraging LLMs to Generate Charts from Abstract Natural Language. *arXiv preprint arXiv:2311.01920* **2023**.
385. Jin, X.; Zhifang, G.; Jinzheng, H.; Hangrui, H.; Ting, H.; Shuai, B.; Keqin, C.; Jialin, W.; Yang, F.; Kai, D.; et al. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215v1* **2025**.
386. Zineng, T.; Ziyi, Y.; Chenguang, Z.; Michael, Z.; Mohit, B. Any-to-Any Generation via Composable Diffusion. *arXiv preprint arXiv:2305.11846* **2023**.
387. Li, S.; Li, B.; Sun, B.; Weng, Y. Towards Visual-Prompt Temporal Answer Grounding in Instructional Video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**.
388. Weng, Y.; Li, B. Visual Answer Localization with Cross-Modal Mutual Knowledge Transfer. In Proceedings of the ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023.
389. Xie, Q.; Weng, Y.; Zhu, M.; Shen, F.; Huang, S.; Lin, Z.; Zhou, J.; Mao, Z.; Yang, Z.; Yang, L.; et al. How Far Are AI Scientists from Changing the World? *arXiv preprint arXiv:2507.23276* **2025**.
390. OpenManus. OpenManus: An Open Multi-Agent Research Framework. https://openmanus.github.io/, 2025. Accessed: 2025-11-13.
391. Team, A.E. How we built our multi-agent research system, 2025. Accessed 2025-08-27.
392. Li, K.; Zhang, Z.; Yin, H.; Ye, R.; Zhao, Y.; Zhang, L.; Ou, L.; Zhang, D.; Wu, X.; Wu, J.; et al. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *arXiv preprint arXiv:2509.13305* **2025**.
393. Qi, Z.; Liu, X.; Iong, I.L.; Lai, H.; Sun, X.; Zhao, W.; Yang, Y.; Yang, X.; Sun, J.; Yao, S.; et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337* **2024**.
394. Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N.A.; Khashabi, D.; Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560* **2022**.
395. Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems* **2023**.
396. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Stanford alpaca: An instruction-following llama model, 2023.
397. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* **2024**, *25*, 1–53.
398. Ge, T.; Chan, X.; Wang, X.; Yu, D.; Mi, H.; Yu, D. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094* **2024**.
399. Zhu, H.; Qin, T.; Zhu, K.; Huang, H.; Guan, Y.; Xia, J.; Yao, Y.; Li, H.; Wang, N.; Liu, P.; et al. Oagents: An empirical study of building effective agents. *arXiv preprint arXiv:2506.15741* **2025**.
400. Ojha, U.; Li, Y.; Sundara Rajan, A.; Liang, Y.; Lee, Y.J. What knowledge gets distilled in knowledge distillation? *Advances in Neural Information Processing Systems* **2023**, *36*, 11037–11048.
401. Lukasik, M.; Bhojanapalli, S.; Menon, A.K.; Kumar, S. Teacher's pet: understanding and mitigating biases in distillation. *arXiv preprint arXiv:2106.10494* **2021**.
402. Zhu, Y.; Liu, N.; Xu, Z.; Liu, X.; Meng, W.; Wang, L.; Ou, Z.; Tang, J. Teach less, learn more: On the undistillable classes in knowledge distillation. *Advances in Neural Information Processing Systems* **2022**, *35*, 32011–32024.
403. Nagarajan, V.; Menon, A.K.; Bhojanapalli, S.; Mobahi, H.; Kumar, S. On student-teacher deviations in distillation: does it pay to disobey? *Advances in Neural Information Processing Systems* **2023**, *36*, 5961–6000.
404. Zhou, X.; Huang, H.; Liao, L. Debate, Reflect, and Distill: Multi-Agent Feedback with Tree-Structured Preference Optimization for Efficient Language Model Enhancement. *arXiv preprint arXiv:2506.03541* **2025**.
405. Chen, J.C.Y.; Saha, S.; Stengel-Eskin, E.; Bansal, M. Magdi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. *arXiv preprint arXiv:2402.01620* **2024**.
406. Qiu, J.; Juan, X.; Wang, Y.; Yang, L.; Qi, X.; Zhang, T.; Guo, J.; Lu, Y.; Yao, Z.; Wang, H.; et al. AgentDistill: Training-Free Agent Distillation with Generalizable MCP Boxes. *arXiv preprint arXiv:2506.14728* **2025**.
407. Reid, A.; O'Callaghan, S.; Carroll, L.; Caetano, T. Risk Analysis Techniques for Governed LLM-based Multi-Agent Systems. *arXiv preprint arXiv:2508.05687* **2025**.

 doi:10.20944/preprints202511.2077.v1

408. Shen, X.; Liu, Y.; Dai, Y.; Wang, Y.; Miao, R.; Tan, Y.; Pan, S.; Wang, X. Understanding the Information Propagation Effects of Communication Topologies in LLM-based Multi-Agent Systems. *arXiv preprint arXiv:2505.23352* **2025**.
409. Chen, Z.; Deng, Y.; Yuan, H.; Ji, K.; Gu, Q. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335* **2024**.
410. Zeng, W.; Xu, C.; Zhao, Y.; Lou, J.G.; Chen, W. Automatic instruction evolving for large language models. *arXiv preprint arXiv:2406.00770* **2024**.
411. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *nature* **2017**.
412. Fawzi, A.; Balog, M.; Huang, A.; Hubert, T.; Romera-Paredes, B.; Barekatain, M.; Novikov, A.; R. Ruiz, F.J.; Schrittwieser, J.; Swirszcz, G.; et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* **2022**, *610*, 47–53.
413. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Gal, Y.; Papernot, N.; Anderson, R. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493* **2023**.
414. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; Gal, Y. AI models collapse when trained on recursively generated data. *Nature* **2024**.
415. Herel, D.; Mikolov, T. Collapse of self-trained language models. *arXiv preprint arXiv:2404.02305* **2024**.
416. Alemohammad, S.; Casco-Rodriguez, J.; Luzi, L.; Humayun, A.I.; Babaei, H.; LeJeune, D.; Siahkoohi, A.; Baraniuk, R.G. Self-consuming generative models go mad. In Proceedings of the International Conference on Learning Representations, 2024.
417. Bertrand, Q.; Bose, A.J.; Duplessis, A.; Jiralerspong, M.; Gidel, G. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arXiv:2310.00429* **2023**.
418. Dohmatob, E.; Feng, Y.; Yang, P.; Charton, F.; Kempe, J. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043* **2024**.
419. Briesch, M.; Sobania, D.; Rothlauf, F. Large Language Models Suffer From Their Own Output: An Analysis of the Self-Consuming Training Loop. *arXiv preprint arXiv:2311.16822* **2023**.
420. OpenAI. ChatGPT: Language Model. https://openai.com/, 2023.
421. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* **2025**.
422. Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476* **2025**.
423. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* **2022**.
424. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.I.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *CoRR* **2015**.
425. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to information retrieval*; Vol. 39, Cambridge University Press Cambridge, 2008.
426. Arora, R.K.; Wei, J.; Hicks, R.S.; Bowman, P.; Candela, J.Q.; Tsimpourlas, F.; Sharman, M.; Shah, M.; Vallone, A.; Beutel, A.; et al. HealthBench: Evaluating Large Language Models Towards Improved Human Health. *ArXiv* **2025**, *abs/2505.08775*.
427. Zhu, S.; Jiang, Y.; Sang, H.; Tang, S.; Song, Q.; He, B.; Jain, R.; Wang, Z.; Geramifard, A. Planner-R1: Reward Shaping Enables Efficient Agentic RL with Smaller LLMs. *arXiv preprint arXiv:2509.25779* **2025**.
428. Zhu, C.; Wang, S.; Feng, R.; Song, K.; Qiu, X. ConvSearch-R1: Enhancing Query Reformulation for Conversational Search with Reasoning via Reinforcement Learning. *arXiv preprint arXiv:2505.15776* **2025**.
429. Liu, Y.; Liu, Y.; Yuan, F.; Cao, C.; Sun, Y.; Peng, K.; Chen, W.; Li, J.; Ma, Z. OPERA: A Reinforcement Learning–Enhanced Orchestrated Planner-Executor Architecture for Reasoning-Oriented Multi-Hop Retrieval. *arXiv preprint arXiv:2508.16438* **2025**.
430. Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; Tang, S. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921* **2024**.
431. Tian, Y.E.; Tang, Y.C.; Wang, K.D.; Yen, A.Z.; Peng, W.C. Template-Based Financial Report Generation in Agentic and Decomposed Information Retrieval. *arXiv preprint arXiv:2504.14233* **2025**.
432. Bosse, N.I.; Evans, J.; Gambee, R.G.; Hnyk, D.; Mühlbacher, P.; Phillips, L.; Schwarz, D.; Wildman, J.; et al. Deep Research Bench: Evaluating AI Web Research Agents. *arXiv preprint arXiv:2506.06287* **2025**.

 doi:10.20944/preprints202511.2077.v1

433. Li, H.; Dong, Q.; Chen, J.; Su, H.; Zhou, Y.; Ai, Q.; Ye, Z.; Liu, Y. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579* **2024**.
434. Zhang*, T.; Kishore*, V.; Wu*, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations, 2020.
435. Su, B.; Zhang, J.; Collina, N.; Yan, Y.; Li, D.; Cho, K.; Fan, J.; Roth, A.; Su, W. The ICML 2023 ranking experiment: Examining author self-assessment in ML/AI peer review. *Journal of the American Statistical Association* **2025**, pp. 1–16.
436. Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594* **2024**.
437. Li, J.; Sun, S.; Yuan, W.; Fan, R.Z.; Zhao, H.; Liu, P. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470* **2023**.
438. Liu, J.; Wang, K.; Chen, Y.; Peng, X.; Chen, Z.; Zhang, L.; Lou, Y. Large language model-based agents for software engineering: A survey. *arXiv preprint arXiv:2409.02977* **2024**.
439. Jimenez, C.E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; Narasimhan, K. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770* **2023**.
440. Wang, R.; Jansen, P.; Côté, M.A.; Ammanabrolu, P. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540* **2022**.
441. Wang, R.; Todd, G.; Xiao, Z.; Yuan, X.; Côté, M.A.; Clark, P.; Jansen, P. Can language models serve as text-based world simulators? *arXiv preprint arXiv:2406.06485* **2024**.
442. Jansen, P.; Côté, M.A.; Khot, T.; Bransom, E.; Dalvi Mishra, B.; Majumder, B.P.; Tafjord, O.; Clark, P. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in Neural Information Processing Systems* **2024**.
443. Fan, R.Z.; Wang, Z.; Liu, P. MegaScience: Pushing the Frontiers of Post-Training Datasets for Science Reasoning. *arXiv preprint arXiv:2507.16812* **2025**.
444. Siegel, Z.S.; Kapoor, S.; Nagdir, N.; Stroebl, B.; Narayanan, A. Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363* **2024**.
445. Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, M.P.; Dupont, E.; Ruiz, F.J.R.; Ellenberg, J.S.; Wang, P.; Fawzi, O.; et al. Mathematical discoveries from program search with large language models. *Nature* **2023**.
446. Sun, W.; Feng, S.; Li, S.; Yang, Y. CO-Bench: Benchmarking Language Model Agents in Algorithm Search for Combinatorial Optimization. *arXiv preprint arXiv: 2504.04310* **2025**.
447. Huang, Q.; Vora, J.; Liang, P.; Leskovec, J. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302* **2023**.
448. Chan, J.S.; Chowdhury, N.; Jaffe, O.; Aung, J.; Sherburn, D.; Mays, E.; Starace, G.; Liu, K.; Maksin, L.; Patwardhan, T.; et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095* **2024**.
449. Wijk, H.; Lin, T.; Becker, J.; Jawhar, S.; Parikh, N.; Broadley, T.; Chan, L.; Chen, M.; Clymer, J.; Dhyani, J.; et al. Re-bench: Evaluating frontier ai r&d capabilities of language model agents against human experts. *arXiv preprint arXiv:2411.15114* **2024**.
450. Li, S.; Sun, W.; Li, S.; Talwalkar, A.; Yang, Y. Towards Community-Driven Agents for Machine Learning Engineering. *arXiv preprint arXiv: 2506.20640* **2025**.
451. Jing, L.; Huang, Z.; Wang, X.; Yao, W.; Yu, W.; Ma, K.; Zhang, H.; Du, X.; Yu, D. DSBench: How Far Are Data Science Agents from Becoming Data Science Experts? *arXiv preprint arXiv:2409.07703* **2024**.
452. Cao, R.; Lei, F.; Wu, H.; Chen, J.; Fu, Y.; Gao, H.; Xiong, X.; Zhang, H.; Hu, W.; Mao, Y.; et al. Spider2-v: How far are multimodal agents from automating data science and engineering workflows? *Advances in Neural Information Processing Systems* **2024**.
453. Zhang, Y.; Jiang, Q.; Han, X.; Chen, N.; Yang, Y.; Ren, K. Benchmarking data science agents. *arXiv preprint arXiv:2402.17168* **2024**.
454. Kao, C.H.; Zhao, W.; Revankar, S.; Speas, S.; Bhagat, S.; Datta, R.; Phoo, C.P.; Mall, U.; Vondrick, C.; Bala, K.; et al. Towards llm agents for earth observation. *arXiv preprint arXiv:2504.12110* **2025**.
455. Ouyang, A.; Guo, S.; Arora, S.; Zhang, A.L.; Hu, W.; R'e, C.; Mirhoseini, A. KernelBench: Can LLMs Write Efficient GPU Kernels? *arXiv preprint arXiv: 2502.10517* **2025**.
456. Wang, Z.; Zheng, X.; An, K.; Ouyang, C.; Cai, J.; Wang, Y.; Wu, Y. StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization. *arXiv preprint arXiv:2505.15107* **2025**.

 doi:10.20944/preprints202511.2077.v1

457. Gao, H.a.; Geng, J.; Hua, W.; Hu, M.; Juan, X.; Liu, H.; Liu, S.; Qiu, J.; Qi, X.; Wu, Y.; et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046* **2025**.
458. Fang, J.; Peng, Y.; Zhang, X.; Wang, Y.; Yi, X.; Zhang, G.; Xu, Y.; Wu, B.; Liu, S.; Li, Z.; et al. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407* **2025**.
459. Zhang, W.; Zhang, X.; Zhang, C.; Yang, L.; Shang, J.; Wei, Z.; Zou, H.P.; Huang, Z.; Wang, Z.; Gao, Y.; et al. Personaagent: When large language model agents meet personalization at test time. *arXiv preprint arXiv:2506.06254* **2025**.
460. Niu, Z.; Xie, Z.; Cao, S.; Lu, C.; Ye, Z.; Xu, T.; Liu, Z.; Gao, Y.; Chen, J.; Xu, Z.; et al. PaRT: Enhancing Proactive Social Chatbots with Personalized Real-Time Retrieval. In Proceedings of the Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025.
461. Wang, Y.; Chen, X. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957* **2025**.
462. Tan, Z.; Yan, J.; Hsu, I.; Han, R.; Wang, Z.; Le, L.T.; Song, Y.; Chen, Y.; Palangi, H.; Lee, G.; et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. *arXiv preprint arXiv:2503.08026* **2025**.
463. Zhao, Z.; Vania, C.; Kayal, S.; Khan, N.; Cohen, S.B.; Yilmaz, E. PersonaLens: A Benchmark for Personalization Evaluation in Conversational AI Assistants. *arXiv preprint arXiv:2506.09902* **2025**.
464. Du, Y.; Wang, B.; He, Y.; Liang, B.; Wang, B.; Li, Z.; Gui, L.; Pan, J.Z.; Xu, R.; Wong, K.F. Bridging the Long-Term Gap: A Memory-Active Policy for Multi-Session Task-Oriented Dialogue. *arXiv preprint arXiv:2505.20231* **2025**.
465. Christmann, P.; Weikum, G. Recursive Question Understanding for Complex Question Answering over Heterogeneous Personal Data. *arXiv preprint arXiv:2505.11900* **2025**.
466. Nan, J.; Ma, W.; Wu, W.; Chen, Y. Nemori: Self-Organizing Agent Memory Inspired by Cognitive Science. *arXiv preprint arXiv:2508.03341* **2025**.
467. Yang, W.; Xiao, J.; Zhang, H.; Zhang, Q.; Wang, Y.; Xu, B. Coarse-to-Fine Grounded Memory for LLM Agent Planning. *arXiv preprint arXiv:2508.15305* **2025**.
468. Yu, H.; Chen, T.; Feng, J.; Chen, J.; Dai, W.; Yu, Q.; Zhang, Y.Q.; Ma, W.Y.; Liu, J.; Wang, M.; et al. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent. *arXiv preprint arXiv:2507.02259* **2025**.
469. Zhou, H.; Chen, Y.; Guo, S.; Yan, X.; Lee, K.H.; Wang, Z.; Lee, K.Y.; Zhang, G.; Shao, K.; Yang, L.; et al. AgentFly: Fine-tuning LLM Agents without Fine-tuning LLMs. *arXiv preprint arXiv:2508.16153* **2025**.
470. Xue, Z.; Zheng, L.; Liu, Q.; Li, Y.; Ma, Z.; An, B. SimpleTIR: End-to-End Reinforcement Learning for Multi-Turn Tool-Integrated Reasoning, 2025. Notion Blog.
471. Wang, Z.; Wang, K.; Wang, Q.; Zhang, P.; Li, L.; Yang, Z.; Jin, X.; Yu, K.; Nguyen, M.N.; Liu, L.; et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073* **2025**.
472. supermancmk. when running the official gsm8k with tool, multi turn async rollout sglang example without any modifications, the model crashes and appears Nan. https://github.com/volcengine/verl/issues/1581, 2025.
473. Zhu, W.; Xie, R.; Wang, R.; Sun, X.; Wang, D.; Liu, P. Proximal Supervised Fine-Tuning. *arXiv preprint arXiv:2508.17784* **2025**.
474. Wang, J.; Ming, Y.; Dulepet, R.; Chen, Q.; Xu, A.; Ke, Z.; Sala, F.; Albarghouthi, A.; Xiong, C.; Joty, S. LiveResearchBench: A Live Benchmark for User-Centric Deep Research in the Wild. *arXiv preprint arXiv:2510.14240* **2025**.
475. Zheng, D.; Lapata, M.; Pan, J.Z. Long-Form Information Alignment Evaluation Beyond Atomic Facts. *arXiv preprint arXiv:2505.15792* **2025**.
476. Li, A.; Yu, L. Summary Factual Inconsistency Detection Based on LLMs Enhanced by Universal Information Extraction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025.
477. Que, H.; Duan, F.; He, L.; Mou, Y.; Zhou, W.; Liu, J.; Rong, W.; Wang, Z.M.; Yang, J.; Zhang, G.; et al. Hellobench: Evaluating long text generation capabilities of large language models. *arXiv preprint arXiv:2409.16191* **2024**.
478. Liu, X.; Dong, P.; Hu, X.; Chu, X. Longgenbench: Long-context generation benchmark. *arXiv preprint arXiv:2410.04199* **2024**.

 doi:10.20944/preprints202511.2077.v1

479. Wei, A.; Wu, Y.; Wan, Y.; Suresh, T.; Tan, H.; Zhou, Z.; Koyejo, S.; Wang, K.; Aiken, A. SATBench: Benchmarking LLMs' Logical Reasoning via Automated Puzzle Generation from SAT Formulas. *arXiv preprint arXiv:2505.14615* **2025**.
480. Parmar, M.; Patel, N.; Varshney, N.; Nakamura, M.; Luo, M.; Mashetty, S.; Mitra, A.; Baral, C. Towards systematic evaluation of logical reasoning ability of large language models. *CoRR* **2024**.
481. Guan, J.; Mao, X.; Fan, C.; Liu, Z.; Ding, W.; Huang, M. Long text generation by modeling sentence-level and discourse-level coherence. *arXiv preprint arXiv:2105.08963* **2021**.
482. Franceschelli, G.; Musolesi, M. On the creativity of large language models. *AI & society* **2025**, *40*.
483. Lin, E.; Peng, Z.; Fang, Y. Evaluating and enhancing large language models for novelty assessment in scholarly publications. In Proceedings of the Proceedings of the 1st Workshop on AI and Scientific Discovery: Directions and Opportunities, 2025.
484. Jiang, X.; Tian, Y.; Hua, F.; Xu, C.; Wang, Y.; Guo, J. A survey on large language model hallucination via a creativity perspective. *arXiv preprint arXiv:2402.06647* **2024**.
485. Zhang, Y.; Diddee, H.; Holm, S.; Liu, H.; Liu, X.; Samuel, V.; Wang, B.; Ippolito, D. NoveltyBench: Evaluating Language Models for Humanlike Diversity. *arXiv preprint arXiv:2504.05228* **2025**.
486. Azerbayev, Z.; Piotrowski, B.; Schoelkopf, H.; Ayers, E.W.; Radev, D.; Avigad, J. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433* **2023**.
487. Yu, Z.; Peng, R.; Ding, K.; Li, Y.; Peng, Z.; Liu, M.; Zhang, Y.; Yuan, Z.; Xin, H.; Huang, W.; et al. Formalmath: Benchmarking formal mathematical reasoning of large language models. *arXiv preprint arXiv:2505.02735* **2025**.
488. Ren, Z.; Shao, Z.; Song, J.; Xin, H.; Wang, H.; Zhao, W.; Zhang, L.; Fu, Z.; Zhu, Q.; Yang, D.; et al. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801* **2025**.
489. Zheng, K.; Han, J.M.; Polu, S. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. *arXiv preprint arXiv:2109.00110* **2021**.
490. Tsoukalas, G.; Lee, J.; Jennings, J.; Xin, J.; Ding, M.; Jennings, M.; Thakur, A.; Chaudhuri, S. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition. *Advances in Neural Information Processing Systems* **2024**.
491. Ming, Y.; Purushwalkam, S.; Pandit, S.; Ke, Z.; Nguyen, X.P.; Xiong, C.; Joty, S. FaithEval: Can Your Language Model Stay Faithful to Context, Even If" The Moon is Made of Marshmallows". *arXiv preprint arXiv:2410.03727* **2024**.
492. Liu, Y.; Wei, X.; Shi, L.; Li, X.; Zhang, B.; Dhillon, P.; Mei, Q. ExAnte: A Benchmark for Ex-Ante Inference in Large Language Models. *arXiv preprint arXiv:2505.19533* **2025**.
493. Karger, E.; Bastani, H.; Yueh-Han, C.; Jacobs, Z.; Halawi, D.; Zhang, F.; Tetlock, P.E. Forecastbench: A dynamic benchmark of ai forecasting capabilities. *arXiv preprint arXiv:2409.19839* **2024**.
494. Zhu, L.; Wang, X.; Wang, X. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631* **2023**.
495. Sahoo, A.; Karnuthala, J.K.; Budhwani, T.P.; Agarwal, P.; Vaidyanathan, S.; Siu, A.; Dernoncourt, F.; Healey, J.; Lipka, N.; Rossi, R.; et al. Quantitative LLM Judges. *arXiv preprint arXiv:2506.02945* **2025**.
496. Zhen, C.; Zheng, E.; Kuang, J.; Tso, G.J. Enhancing llm-as-a-judge through active-sampling-based prompt optimization. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, 2025.
497. Lu, L.C. A Critical Analysis of Existing Creativity Evaluations, 2025, [arXiv:cs.CL/2508.05470]. https://doi.org/10.48550/arXiv.2508.05470.
498. Yang, Q.; Shi, Q.; Wang, T.; Ye, M. Uncertain multimodal intention and emotion understanding in the wild. In Proceedings of the Proceedings of the Computer Vision and Pattern Recognition Conference, 2025.
499. Yang, Q.; Ye, M.; Du, B. Emollm: Multimodal emotional understanding meets large language models. *arXiv preprint arXiv:2406.16442* **2024**.
500. Wang, X.; Wang, H.; Zhang, Y.; Yuan, X.; Xu, R.; Huang, J.t.; Yuan, S.; Guo, H.; Chen, J.; Zhou, S.; et al. CoSER: Coordinating LLM-Based Persona Simulation of Established Roles. In Proceedings of the The Forty-second International Conference on Machine Learning, 2025.
501. Wan, Y.; Wang, W.; Yang, Y.; Yuan, Y.; Huang, J.t.; He, P.; Jiao, W.; Lyu, M. LogicAsker: Evaluating and Improving the Logical Reasoning Ability of Large Language Models. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024.

 doi:10.20944/preprints202511.2077.v1

502. Sprague, Z.; Yin, F.; Rodriguez, J.D.; Jiang, D.; Wadhwa, M.; Singhal, P.; Zhao, X.; Ye, X.; Mahowald, K.; Durrett, G. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183* **2024**.
503. Wu, C.H. Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction. In Proceedings of the International Conference on Machine Learning, 2025.
504. Ferrara, E. Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. *Sci* **2024**.
505. Kheya, T.A. The Pursuit of Fairness in Artificial Intelligence Models. *arXiv preprint arXiv:2403.17333* **2024**.
506. Tang, X.; Li, J.; Hu, K.; Nan, D.; Li, X.; Zhang, X.; Sun, W.; Xie, S. CogniBench: A Legal-inspired Framework and Dataset for Assessing Cognitive Faithfulness of Large Language Models. *arXiv preprint arXiv:2505.20767* **2025**.
507. Chen, X.; He, B.; Lin, H.; Han, X.; Wang, T.; Cao, B.; Sun, L.; Sun, Y. Spiral of Silence: How is Large Language Model Killing Information Retrieval? – A Case Study on Open Domain Question Answering, 2024, [arXiv:cs.IR/2404.10496]. https://doi.org/10.48550/arXiv.2404.10496.