Idea2Paper: What Should an End-to-End Research Agent Really Do?

Tengyue Xu*, **Zhuoyang Qian***, **Gaoge Liu***, **Zhentao Zhang***, **Li Ling***, **Biao Wu***, **Shuo Zhang**, **Ke Lu**, **Wei Shi**, **Ziqi Wang**, **Zheng Feng**, **Yan Luo**, **Shu Xu**, **Yongjin Chen**, **Zhibo Feng**, **Zhuo Chen**, **Bruce Yuan**, **Harry Wang**[†], **Kris Chen**[†]

AgentAlpha Team
[†]Corresponding author

Automated Scientific Research System, which aim to automate substantial portions of the scientific discovery process. Moving beyond isolated sub-tasks, these systems increasingly pursue end-to-end pipelines that integrate hypothesis generation, literature exploration, research planning, experimental execution, result interpretation, manuscript writing, and peer review. Despite rapid progress, the field remains fragmented, with limited consensus on system structure and evaluation. In this work, we provide a unified analysis of ASRS from the perspective of the scientific research lifecycle. We formalize ASRS as closed-loop pipelines composed of seven interdependent stages and review representative approaches for each stage, highlighting common design patterns and key technical challenges. Our analysis identifies recurring limitations, including the ideation–execution gap, trade-offs between novelty and feasibility, weaknesses in multimodal reasoning, and blind spots in automated evaluation. By organizing existing work into a lifecycle-based framework, our work clarifies the current capabilities and limitations of ASRS and highlights promising directions for future research, emphasizing deep human–AI collaboration as a practical path forward.

Date: January 27, 2026

1 Introduction

Artificial intelligence-driven scientific research is undergoing a paradigm shift, moving from systems designed for isolated sub-tasks toward more general and autonomous research entities (Wang et al., 2023; Eren and Perez, 2025). Traditionally, scientific discovery has been a costly and time-consuming process, predicated on human intuition, domain expertise, and manual experimentation. Recent breakthroughs in foundation models (Team et al., 2024; Dubey et al., 2024) and code generation capabilities (DeepSeek-AI et al., 2024; Wang et al., 2025b; Ni et al., 2025) have catalyzed this transition. Using advanced reasoning frameworks such as (Wei et al., 2022; Yao et al., 2022; Richards, 2023; Wu et al., 2023; Li et al., 2023; Hu et al., 2025), combined with self-reflective architectures (Shinn et al., 2023) and generative simulation environments (Park et al., 2023), modern systems can autonomously traverse the entire research loop: from hypothesis and execution experiments, to producing coherent research manuscripts (Boiko et al., 2023; Lu et al., 2024; Mitchener et al., 2025). This evolution has the potential to broaden access to scientific research by scaling discovery capabilities and reducing both the entry barriers and the marginal cost of conducting individual studies.

However, the current landscape of Automated Scientific Research System (ASRS) remains fragmented and conceptually unclear. Existing efforts often propose new methods or components in isolation, making it difficult for researchers to assess how improvements to a specific module contribute to the performance and reliability of the overall system (Jin et al., 2024; Sahu et al., 2025; Ajith et al., 2024; Zhang et al., 2025b, 2026). Moreover, when attempting to redesign or optimize end-to-end research agents, it is often unclear how many distinct stages are required, what roles each stage should play, and which challenges are intrinsic to individual components versus emergent from their interaction (Lu et al., 2024; Schmidgall et al., 2025; Baek et al., 2025; Hong et al., 2023; Wu et al., 2023; Lin et al., 2025; Hu et al., 2026). As a result, progress in this area is frequently incremental and difficult to compare across systems. To address these issues, we provide a structured analysis of automated research pipelines by decomposing them into a set of well-defined stages

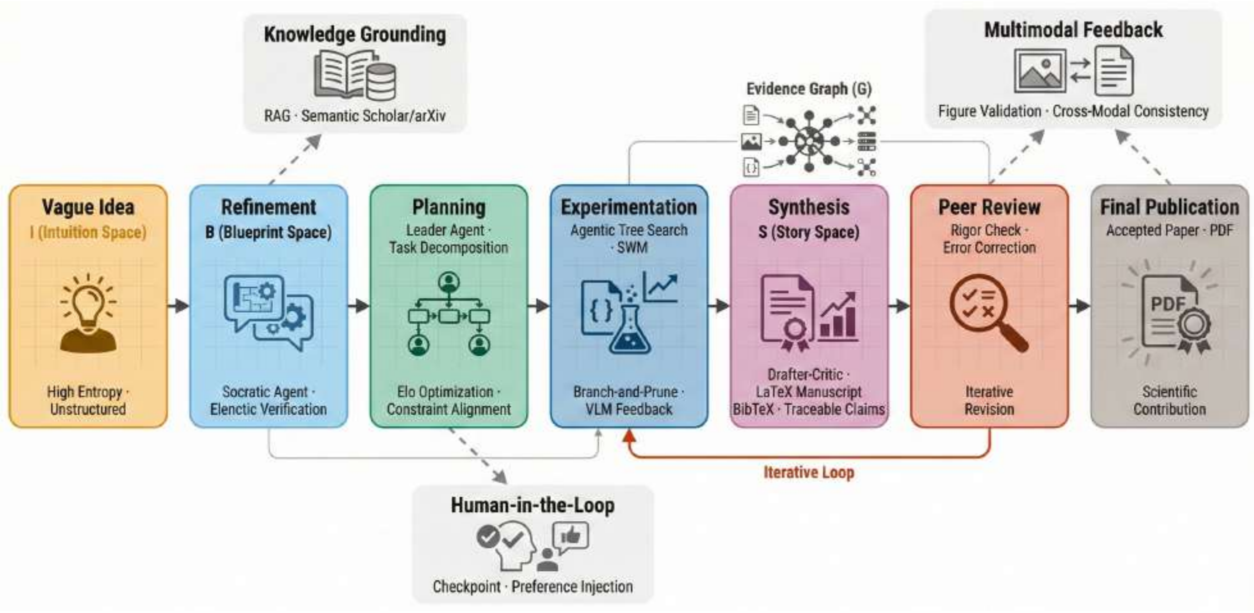

Knowledge Grounding
RAG · Semantic Scholar/arXiv
Evidence Graph (G)
Multimodal Feedback
Figure Validation · Cross-Modal Consistency
Vague Idea
I (Intuition Space)
High Entropy · Unstructured
Refinement
B (Blueprint Space)
Socratic Agent · Elenctic Verification
Planning
Leader Agent · Task Decomposition
Elo Optimization · Constraint Alignment
Experimentation
Agentic Tree Search · SWM
Branch-and-Prune · VLM Feedback
Synthesis
S (Story Space)
Drafter-Critic · LaTeX Manuscript BibTeX · Traceable Claims
Peer Review
Rigor Check · Error Correction
Iterative Revision
Final Publication
Accepted Paper · PDF
Scientific Contribution
Iterative Loop
Human-in-the-Loop
Checkpoint · Preference Injection

Figure 1 Overview of the Automated Scientific Research System lifecycle. The pipeline illustrates how an initial vague idea is progressively transformed into a reproducible scientific artifact through a sequence of stages.

and formulating a corresponding set of research questions, offering a clearer framework for understanding, evaluation, and future development.

To systematically analyze progress in this emerging area, we formalize the notion of ASRS from the perspective of the scientific research lifecycle. Specifically, we identify seven defining stages that together characterize end-to-end automated research systems:

1. *Autonomous Ideation*, which addresses the problem of generating scientifically meaningful and testable research hypotheses. In this stage, agents explore the hypothesis space by leveraging large language models in conjunction with evolutionary strategies to balance novelty and feasibility.

2. *Literature Exploration and Knowledge Grounding*, which focuses on grounding proposed ideas in existing scientific knowledge. This stage retrieves, synthesizes, and contextualizes prior work through academic indexing services such as Semantic Scholar and arXiv, together with browser-assisted frameworks, to establish a rigorous and non-redundant scientific foundation.

3. *Research Planning*, which tackles the problem of translating abstract research goals into executable experimental plans. Here, high-level objectives are decomposed into concrete experimental blueprints through multi-agent dialogue and coordination.

4. *Autonomous Experimentation*, which addresses the execution challenge of implementing and validating planned experiments. As the engineering core of the system, this stage focuses on automatic code generation, environment interaction, error diagnosis, and iterative debugging.

5. *Result Interpretation and Claim Formulation*, which focuses on transforming raw experimental outputs into reliable scientific claims. This stage analyzes empirical results, performs statistical reasoning, and formulates logically coherent and evidence-supported conclusions.

6. *End-to-End Paper Production*, which addresses the problem of scientific communication. In this stage, validated findings and reasoning narratives are organized into structured LaTeX manuscripts that conform to academic writing conventions and formatting standards.

7. *Simulated Peer Review and Iteration*, which targets quality control and self-correction. Systems employ role-playing agents to simulate academic peer review, providing critical feedback that supports iterative refinement, error correction, and claim validation.

Building upon this lifecycle decomposition, we conduct a systematic stage-wise survey of Automated Scientific Research Systems. For each stage, we review representative existing approaches and summarize their core design choices and capabilities. We then analyze the key limitations and open challenges that hinder robust end-to-end performance, followed by a discussion of promising directions for future research. Through this structured examination, we aim to provide a unified understanding of the current landscape of automated scientific research and to highlight actionable opportunities for advancing systems.

The contributions of our work are threefold. First, we propose a comprehensive lifecycle-based taxonomy that synthesizes and contrasts existing ASRS across seven distinct stages. Second, we systematically dissect the critical technical bottlenecks that prevent the transition of these systems from experimental prototypes to reliable scientific contributors. Third, we critically discuss hybrid *scientist-in-the-loop* paradigms and standardized evaluation frameworks, such as MLE-Bench (Chan et al., 2024a), to clarify the current capability boundaries of AI agents in real-world research settings. We hope this work establishes a structured foundation for future work aiming to construct more robust, transparent, and scientifically grounded automated research systems.

2 Autonomous Ideation

Autonomous ideation constitutes the logical starting point of modern ASRS, aiming to translate vague or underspecified human research intents into scientifically meaningful and experimentally verifiable hypotheses. Unlike conventional text-generation tasks, this stage requires not only extensive domain knowledge but also a principled reasoning framework capable of bridging the substantial gap between high-level conceptual ideas and concrete experimental designs. To better understand this stage, we further examine autonomous ideation from two complementary perspectives: ideation as a closed-loop process, and the coordination paradigms that govern hypothesis generation and evaluation.

2.1 Ideation as a Closed-Loop Process

Contemporary ASRS architectures formalize scientific ideation as a three-stage closed-loop process: generation, retrieval, and validation. In this paradigm, candidate hypotheses are first produced by large language models or multi-agent systems, then evaluated against existing literature through large-scale retrieval mechanisms, and finally refined using novelty and feasibility feedback signals (Si et al., 2024; Zhang et al., 2025b; Zhuang et al., 2025; Lin et al., 2023b; Zhang et al., 2025a; Wu et al., 2025). This iterative structure allows systems to balance exploratory diversity with progressive conceptual grounding.

Early ASRS implementations adopted constrained generation strategies. For instance, Lu et al. (2024) treat research ideas as mutation operators in an evolutionary framework, iteratively perturbing predefined code templates to explore architectural or algorithmic variations. While effective within bounded domains, this template-driven approach inherently limits the search space and constrains cross-domain generalization. To overcome these limitations, Yamada et al. (2025) propose a Generalized Ideation phase that eliminates reliance on handcrafted templates, enabling free-form hypothesis construction across diverse machine learning subfields. This transition—often implemented through tree search methodologies (Browne et al., 2012)—represents a shift from local optimization toward open-ended scientific exploration, expanding the system's capacity for conceptual innovation.

Ensuring that generated hypotheses possess genuine scientific novelty remains a central challenge for autonomous discovery systems. To address this, most contemporary architectures integrate large-scale literature retrieval services (e.g., Semantic Scholar, arXiv APIs) for real-time novelty assessment (Baek et al., 2025; Lu et al., 2024; Zhang et al., 2026; Lin et al., 2023a). When substantial overlap with prior work is detected, self-reflection mechanisms reformulate or redirect the hypothesis, serving as a safeguard against superficial recombination of existing knowledge. However, recent findings suggest potential tensions in this feedback loop: Murthy et al. (2025) demonstrate that alignment training may inadvertently compress the conceptual diversity of model-generated ideas, raising questions about whether current validation mechanisms sufficiently preserve exploratory potential.

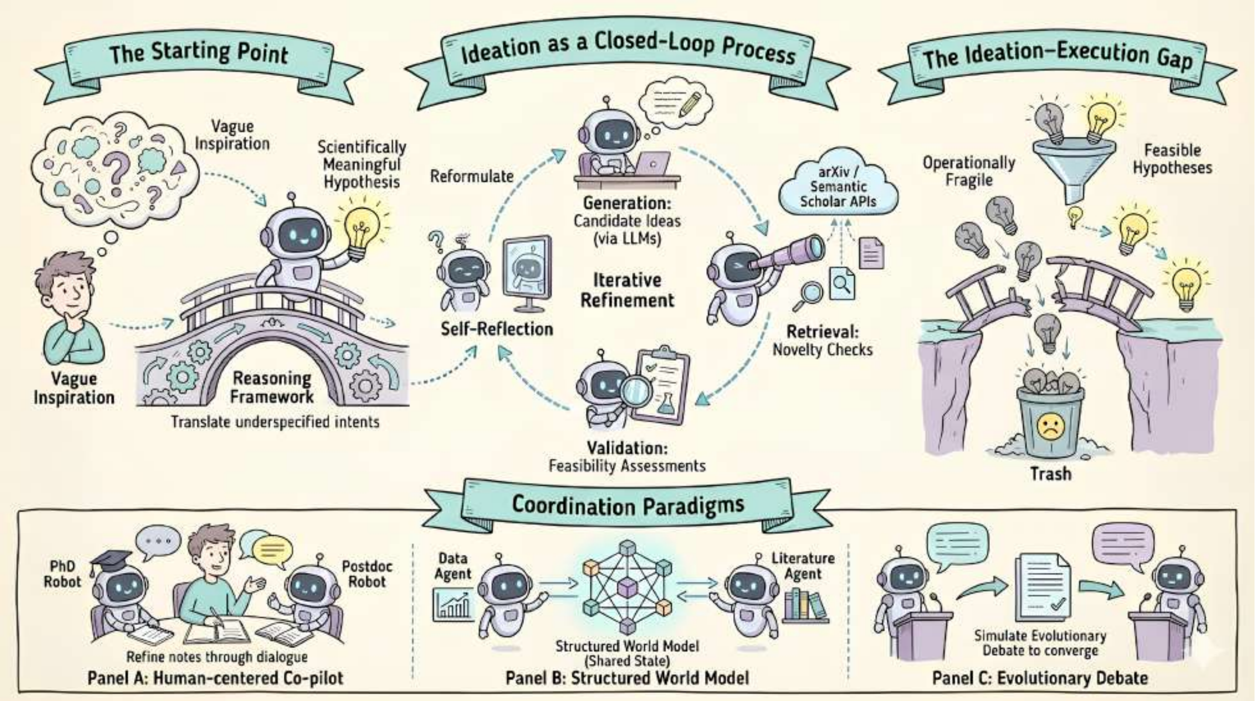

The Starting Point
Vague Inspiration
Scientifically Meaningful Hypothesis
Vague Inspiration
Reasoning Framework
Translate underspecified intents
Ideation as a Closed-Loop Process
Reformulate
Generation: Candidate Ideas (via LLMs)
Iterative Refinement
Self-Reflection
arXiv / Semantic Scholar APIs
Retrieval: Novelty Checks
Validation: Feasibility Assessments
The Ideation–Execution Gap
Operationally Fragile
Feasible Hypotheses
Trash
Coordination Paradigms
PhD Robot
Postdoc Robot
Refine notes through dialogue
Panel A: Human-centered Co-pilot
Data Agent
Literature Agent
Structured World Model (Shared State)
Panel B: Structured World Model
Simulate Evolutionary Debate to converge
Panel C: Evolutionary Debate

Figure 2 Conceptual overview of autonomous scientific ideation in ASRS. The figure illustrates the ideation pipeline as a closed-loop process that transforms vague human inspiration into candidate scientific hypotheses through iterative generation, retrieval, and validation.

2.2 Coordination Paradigms

Beyond ideation, transforming abstract ideas into executable hypotheses requires structured reasoning and effective coordination among specialized agents, and existing ASRS adopt diverse mechanisms to achieve this goal. For example, *Agent Laboratory* (Schmidgall et al., 2025) follows a human-centered co-pilot paradigm, in which rough human notes or initial intuitions are progressively refined through multi-round dialogue among a hierarchy of agents, typically involving a PhD-level agent guided by a postdoctoral supervisor agent. Through this interaction, research objectives are decomposed, assumptions are clarified, and actionable task graphs are constructed, although the effectiveness of this approach often depends on the quality of human-provided inputs. In contrast, *Kosmos* (Mitchener et al., 2025) emphasizes persistent internal structure by introducing a *Structured World Model* that acts as a shared state across data analysis and literature retrieval agents. This design allows the system to maintain logical coherence over long execution horizons, reportedly up to 12 hours, and has demonstrated effectiveness in domains such as metabolomics and materials science, where long-range dependency tracking is essential. Complementary to these approaches, Google's *AI co-scientist* (Gottweis et al., 2025) models hypothesis refinement as an evolutionary debate process by simulating scholarly discussions among multiple expert agents. By iteratively generating, critiquing, and revising hypotheses, the system converges toward research proposals that are increasingly rigorous and defensible.

2.3 Future Work

Despite these advances, recent empirical studies have revealed a fundamental limitation of current ideation pipelines, commonly referred to as the Ideation–Execution Gap (Si et al., 2025). Large-scale analyses by (Si et al., 2024, 2025; Xu et al., 2025) show that while LLM-generated ideas often outperform human proposals in terms of initial novelty scores, their effectiveness deteriorates significantly during downstream execution, as evidenced by rigorous evaluations on machine learning experimentation benchmarks (Huang et al., 2023; Ni et al., 2025; Yang et al., 2024; Jimenez et al., 2023). This discrepancy indicates a systematic tendency of existing models to produce hypotheses that are intellectually appealing yet operationally fragile. These findings underscore the importance of incorporating feasibility-aware constraints and execution-grounded

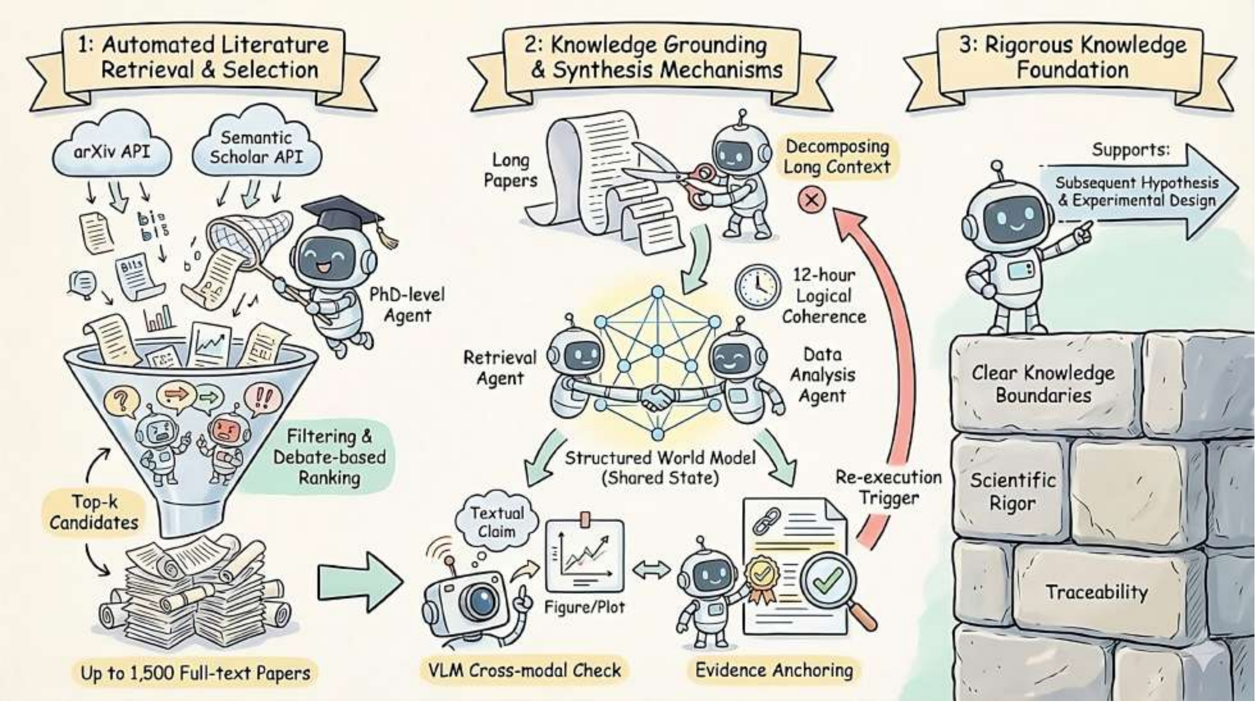

1: Automated Literature Retrieval & Selection
arXiv API
Semantic Scholar API
PhD-level Agent
Filtering & Debate-based Ranking
Top-k Candidates
Up to 1,500 Full-text Papers
2: Knowledge Grounding & Synthesis Mechanisms
Long Papers
Decomposing Long Context
12-hour Logical Coherence
Retrieval Agent
Data Analysis Agent
Structured World Model (Shared State)
Re-execution Trigger
Textual Claim
Figure/Plot
VLM Cross-modal Check
Evidence Anchoring
3: Rigorous Knowledge Foundation
Supports:
Subsequent Hypothesis & Experimental Design
Clear Knowledge Boundaries
Scientific Rigor
Traceability

Figure 3 Overview of the automated scientific discovery pipeline. The system consists of three main stages: (1) automated literature retrieval and selection using semantic search and ranking, (2) knowledge grounding and synthesis through multi-agent reasoning, long-context decomposition, and cross-modal verification, and (3) a rigorous knowledge foundation ensuring clear boundaries, scientific rigor, and traceability.

feedback earlier in the ideation loop, rather than deferring such considerations to later experimental stages.

3 Literature Exploration and Knowledge Grounding

A robust understanding of prior work is a prerequisite for any credible scientific discovery process, and this requirement becomes even more critical in automated research settings where human oversight is limited. For ASRS, failures in literature awareness or knowledge grounding can lead to redundant hypotheses, flawed experimental designs, or unsupported claims, ultimately undermining end-to-end system reliability. As a result, modern ASRS devote substantial architectural capacity to both the acquisition of relevant scientific literature and the explicit grounding of reasoning processes in verifiable evidence. In the following, as shown in Figure 6, we examine these two tightly coupled aspects in detail: first, how existing systems retrieve, filter, and synthesize large-scale scientific literature, and second, how grounding mechanisms are employed to ensure transparency, traceability, and consistency throughout long-horizon autonomous research workflows.

3.1 Automated Literature Retrieval and Selection

Literature exploration and knowledge grounding constitute a core component of ASRS, ensuring that autonomous research activities maintain scientific rigor and academic relevance. Rather than performing simple keyword-based searches, this stage focuses on autonomously reading, synthesizing, and integrating large volumes of scientific literature, thereby establishing solid knowledge boundaries and logical foundations for subsequent hypothesis generation, experimental design, and result discussion. Existing ASRS demonstrate a high degree of specialization and tool integration in literature acquisition and filtering. In the (Schmidgall et al., 2025) framework, PhD-level agents leverage the arXiv API to perform abstract retrieval, full-text extraction, and reference insertion, enabling the construction of comprehensive literature surveys within minutes through iterative querying. In contrast, (Lu et al., 2024) and its successor (Yamada et al., 2025) integrate the Semantic Scholar API to retrieve publications in real time, allowing novelty checks to be conducted during the ideation stage while automatically populating BibTeX entries to ensure citation fidelity. Beyond retrieval, specialized architectures such as (Sahu et al., 2025) incorporate (Agarwal et al., 2024), introducing debate-based ranking mechanisms to prioritize retrieved papers and select the most relevant top-k candidates. At a larger scale, (Mitchener et al., 2025) represents a qualitative leap in knowledge synthesis, with a single run reportedly reading up to 1,500 full-text papers, while empirical studies indicate that literature-focused agents such as (Skarlinski et al., 2024), as reported in (Gottweis et al., 2025) , outperform the average postdoctoral researcher on multiple benchmarks related to literature search and summarization.

3.2 Knowledge Grounding Mechanisms

Knowledge grounding techniques play a critical role in ensuring the transparency and traceability of system behavior in ASRS (Luo et al., 2025; Wang et al., 2025c; Yan et al., 2025). A central innovation of (Mitchener et al., 2025) lies in its use of a Structured World Model, which enables real-time sharing of contextual information between literature retrieval agents and data analysis agents. This shared representation allows the system to maintain logical coherence throughout extended execution periods, reportedly lasting up to 12 hours. In addition to structured representations, explicit factual verification and evidence grounding are enforced in several systems. (Sahu et al., 2025) requires reviewer agents to anchor their judgments to specific textual spans or retrieved documentary evidence; failure to do so triggers an automatic re-execution mechanism. Similarly, (Yamada et al., 2025) introduces a Vision–Language Model (VLM) feedback loop, enabling cross-modal consistency checks between textual claims and figures generated from literature or experimental results. The effectiveness of literature exploration and grounding directly determines the decision quality and engineering feasibility of automated research systems. (D'Arcy et al., 2024) addresses the limitations of long-context reasoning in large language models by decomposing lengthy papers into smaller segments and distributing them across multiple collaborating agents, thereby ensuring deeper grounding in complex technical details. Moreover, findings from (Si et al., 2025) highlight that, despite strong performance in the ideation stage, LLM-generated proposals are prone to being novel yet difficult to execute when insufficiently grounded in existing experimental constraints.

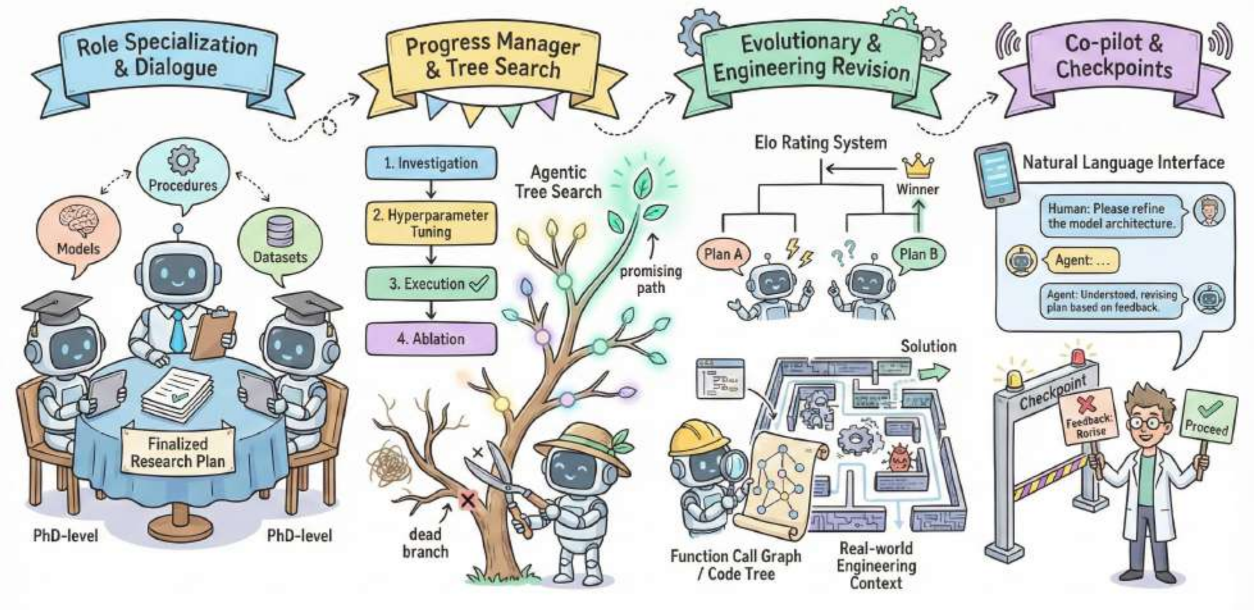
3.3 Future Work

Despite significant progress in automated literature exploration and knowledge grounding, scalability and efficiency remain major bottlenecks for current ASRS. In practice, literature retrieval and synthesis pipelines are often extremely time-consuming, with a single end-to-end run frequently lasting for many hours (Mitchener et al., 2025; Achiam et al., 2023; Team et al., 2025; OpenAI, 2025; Claude Team, 2025; Grok Team, 2025; Gemini Team, 2025). A key reason lies in the substantial redundancy across comprehension and reasoning processes: similar papers are repeatedly read, summarized, and evaluated across different stages, agents, or iterations, leading to duplicated computation and inefficient resource utilization.

Addressing this inefficiency represents an important direction for future work. Promising avenues include caching and reusing intermediate representations of papers, developing persistent knowledge memories that can be shared across agents and runs, and designing modular reasoning components that avoid repeated re-interpretation of identical content. Additionally, tighter integration between retrieval, reasoning, and planning modules may enable more selective reading strategies, allowing systems to focus computational effort on the most informative portions of the literature. Improving the efficiency of literature grounding will be critical for scaling ASRS to larger corpora, longer research horizons, and more complex scientific domains.

4 Research Planning

Research planning serves as a critical bridge between high-level scientific hypotheses and concrete engineering execution. Its primary objective is to translate abstract research ideas into actionable experimental blueprints. In modern ASRS, planning has evolved beyond simple instruction following by a single agent into a complex reasoning process involving multi-agent coordination, dynamic task decomposition, and continuous interaction between humans and AI agents.



Human: Please refine the model architecture.

Agent: ...

Agent: Understoed, revising plan based on feedback.

Figure 4 An overview of coordination-based planning and adaptive execution in ASRS. Multi-agent role specialization supports collaborative plan formulation, which is followed by structured progress management and agentic tree search for experimental exploration. Plans are iteratively refined through evolutionary revision and engineering-aware analysis, with human-in-the-loop checkpoints providing feedback and control via natural language interfaces.

4.1 Coordination-Based Planning

Multi-agent dialogue and role specialization enable structured research planning through iterative consensus-building. In (Schmidgall et al., 2025), planning unfolds through a Plan Formulation stage where PhD-level agents collaborate with postdoctoral supervisor agents to determine experimental strategies via multi-round negotiation. These agents collectively define required models, datasets, and procedural workflows before the supervising agent consolidates their discussions into a finalized plan that directs downstream execution. A complementary architecture appears in (D'Arcy et al., 2024), which employs a leader agent to draft high-level plans and coordinate task allocation across domain-specialized expert agents, thereby maintaining logical coherence when processing complex or lengthy documents.

While agent specialization enhances planning autonomy, human-in-the-loop collaboration remains critical for aligning AI-generated plans with scientific rigor, ethical standards, and practical constraints. Systems like *Agent Laboratory* integrate co-pilot modes that insert human checkpoints after each subtask, allowing researchers to review literature syntheses or experimental designs and provide corrective feedback—prompting agents to revise, repeat, or terminate specific steps as needed. Similarly, (Gottweis et al., 2025) emphasize continuous human oversight through natural language interfaces, enabling scientists to dynamically refine objectives, impose additional constraints, or directly evaluate generated hypotheses. This bidirectional interaction ensures that automated planning processes remain grounded in domain expertise and aligned with authentic research intentions.

4.2 Planning and Error Correction

To cope with the inherent uncertainty of scientific exploration and real-world execution, recent ASRS adopt planning mechanisms that combine structured stage control with adaptive revision. Prior work (Yamada et al., 2025) introduces a dedicated experiment progress manager that decomposes the research process into multiple phases, including preliminary investigation, hyperparameter tuning, research agenda execution, and ablation studies, enabling different constraints and evaluation criteria to be applied at each stage. On top of this structure, agentic tree search is employed to explore multiple experimental branches in parallel, with agents automatically expanding, evaluating, and pruning candidate experiments based on performance metrics and self-reflection signals, thereby balancing exploration efficiency against computational cost. At the same time,

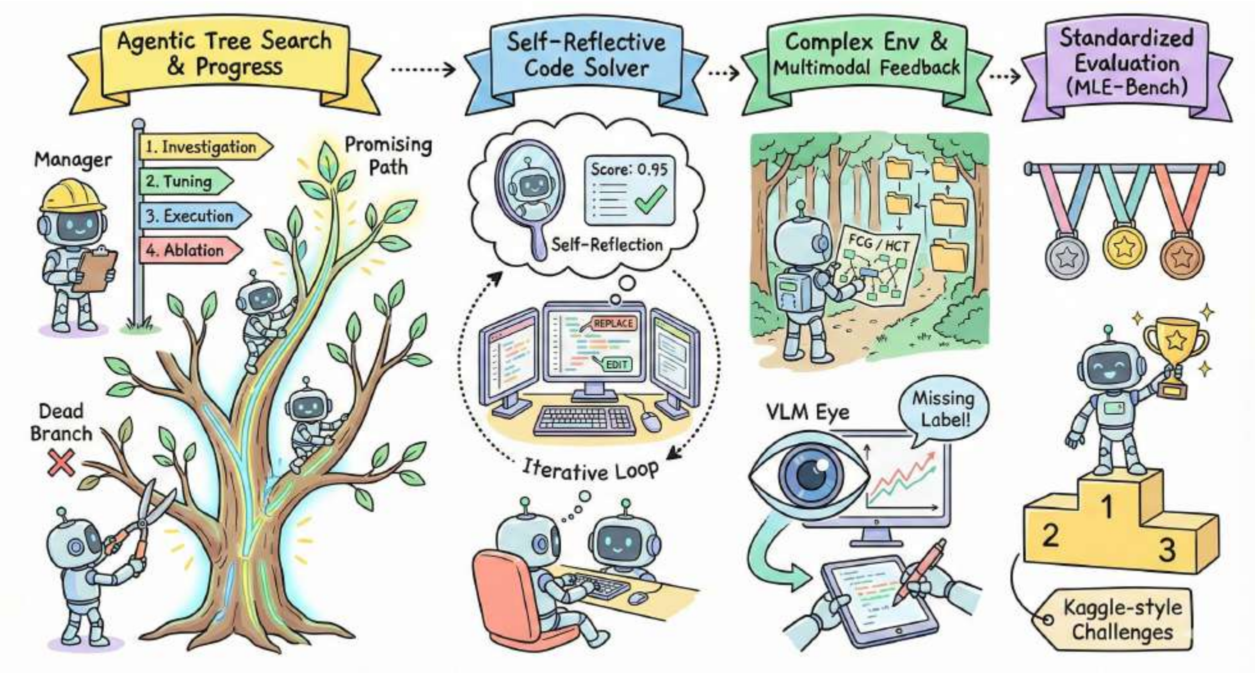


Figure 5 Execution, feedback, and evaluation in ASRS. Agentic tree search enables structured exploration of experimental paths, while self-reflective code solvers iteratively refine implementations based on execution feedback. Multimodal feedback via VLMs supports robust interaction with real-world codebases and visual outputs. Standardized benchmarks such as MLE-Bench provide objective evaluation of autonomous experimentation on Kaggle-style tasks.

complementary approaches model research planning as an evolutionary process (Gottweis et al., 2025), in which research plans are iteratively generated, debated, and refined through tournament-style optimization driven by Elo rating systems. Despite these advances, adaptive adjustment in complex engineering environments remains challenging, as agents may deviate from intended plans due to code-level errors or hallucinated reasoning during execution. To mitigate such failures, recent work (Wang et al., 2025b) introduces explore–execute loops that leverage hierarchical analyses of software repositories, such as hierarchical code trees, function call graphs, and module dependency graphs, to dynamically revise planning trajectories and improve feasibility when operating in real-world engineering contexts.

4.3 Future Work

Despite its central role in enabling robust end-to-end automated research, coordination-based planning and adaptive error correction remain among the least explored components of current ASRS. While existing studies demonstrate promising mechanisms for multi-agent coordination, structured planning, and iterative revision, systematic investigations into their scalability, reliability, and generalization across domains are still scarce. In particular, there is a notable lack of standardized experimental platforms and benchmarks that support reproducible evaluation of planning quality, failure recovery, and long-horizon coordination under realistic scientific and engineering constraints.

Future research should prioritize the development of unified testbeds and simulation environments that expose ASRS to diverse, failure-prone execution scenarios, enabling controlled analysis of planning robustness and error correction strategies. Additionally, richer abstractions for plan representation, shared memory, and inter-agent communication are needed to reduce brittleness and improve transparency in complex workflows. Finally, integrating human-in-the-loop supervision into principled planning and correction frameworks—rather than ad hoc intervention—remains an open challenge, particularly for large-scale, real-world deployments. Addressing these gaps will be critical for advancing ASRS from proof-of-concept systems toward reliable and deployable autonomous research agents.

5 Autonomous Experimentation

Autonomous experimentation forms the engineering core of ASRS. Its primary goal is to transform scientific hypotheses into objective experimental evidence through code generation, environment interaction, and iterative optimization. Compared to early approaches based on linear script execution, modern systems have evolved into complex execution frameworks that integrate structured progress management, large-scale parallel search, and multimodal feedback.

5.1 Core Paradigms of Experimental Execution

Current state-of-the-art ASRS exhibit several representative paradigms for autonomous experimental execution, differing mainly in how experiments are explored, evaluated, and refined. For example, (Yamada et al., 2025) removes the reliance on human-designed code templates used in earlier versions and introduces an Experiment Progress Manager that structures experimentation into four stages: Preliminary Investigation, Hyperparameter Tuning, Research Agenda Execution, and Ablation Studies. Within each stage, the system employs Agentic Tree Search to expand multiple experimental branches in parallel and uses scores produced by LLM-based evaluators to autonomously select the most promising paths for continued exploration. A different approach is taken by (Schmidgall et al., 2025), which emphasizes self-reflective, code-driven optimization. In this framework, a core solver iteratively modifies machine learning code through `REPLACE` and `EDIT` operations, while an automatic evaluator assigns execution outcomes numerical scores in the range $[0, 1]$ to determine whether modifications should be retained. Importantly, agents perform self-reflection after both successful and failed executions, analyzing error signals or experimental metrics to guide subsequent iterations. In contrast to these code-centric strategies, (Mitchener et al., 2025) focuses on structured discovery over large-scale real-world datasets. It employs a Structured World Model to maintain shared contextual state between data analysis agents and literature retrieval agents, enabling coherent reasoning over long execution horizons. A single run reportedly executes approximately 42,000 lines of code and generates 166 analysis trajectories, supporting autonomous discoveries in domains such as metabolomics and materials science over execution periods lasting up to 12 hours.

5.2 Execution in Real-World Engineering Environments

Autonomous experimentation often requires interacting with large and complex existing codebases, which introduces additional challenges for navigation, execution, and reuse (Lyu et al., 2023; Tang et al., 2023; Yu et al., 2024; Jimenez et al., 2023; Zhang et al., 2024). To support this setting, prior work (Wang et al., 2025b) proposes specialized exploration and execution tools that construct hierarchical representations of software repositories, including Function Call Graphs (FCG), Module Dependency Graphs (MDG), and Hierarchical Code Trees (HCT). These structures enable agents to identify critical components and efficiently reuse open-source repositories through an iterative *explore–execute loop*, even under limited context budgets. In parallel, recent systems incorporate multimodal feedback mechanisms to improve experimental robustness. For example, work on autonomous scientific discovery (Yamada et al., 2025) integrates a *Vision–Language Model (VLM)* feedback loop that allows agents to visually inspect experimental plots and figures, detect issues such as missing labels, incorrect scales, or logical inconsistencies, and automatically revise plotting or analysis code in a manner similar to human scientists.

5.3 Future Work

As autonomous experimentation systems continue to mature, the evaluation of research agents has increasingly relied on standardized benchmarks. A representative example is *MLE-Bench* (Chan et al., 2024b), which evaluates AI agents on realistic machine learning engineering tasks, including data preparation, model development, and competition submission, using Kaggle-style challenges. Recent studies (Schmidgall et al., 2025; Wang et al., 2025b) report promising results on this benchmark, with leading agents approaching or even surpassing median human performance on relatively low-complexity tasks.

However, existing benchmarks remain limited in their ability to reflect the full complexity of autonomous experimentation. In particular, they often focus on short-horizon, well-scoped tasks and provide limited coverage of long-horizon reasoning, intricate engineering dependencies, and robust failure recovery under

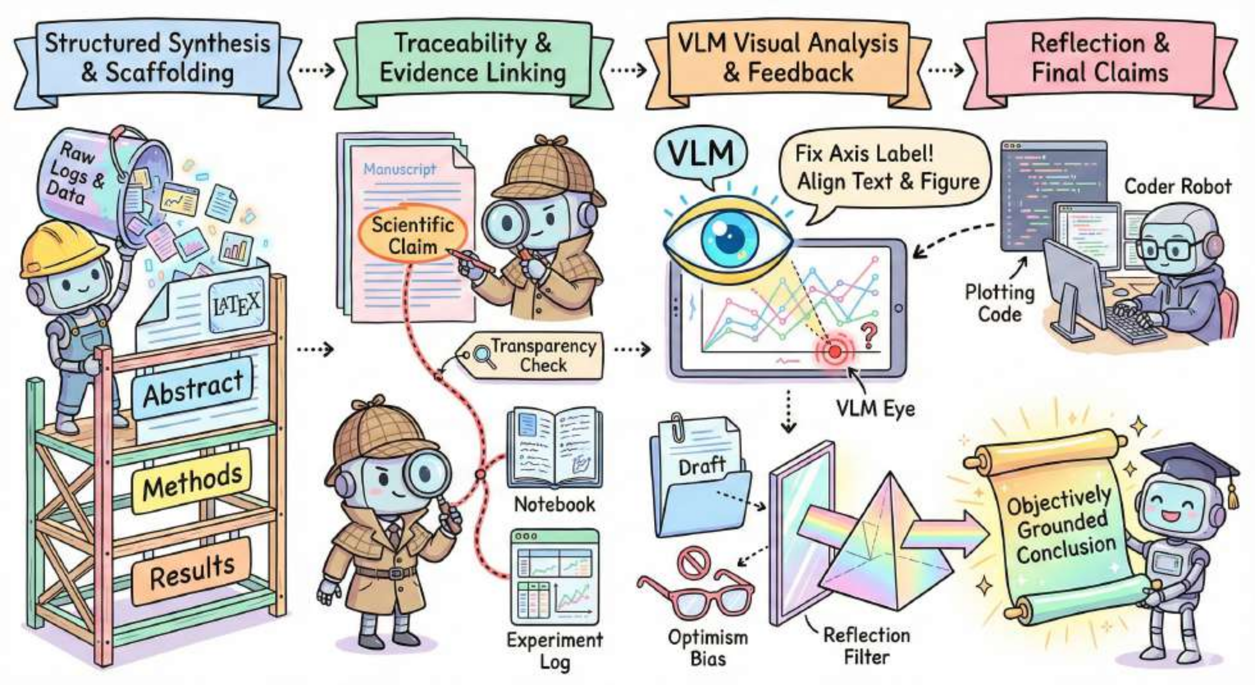

Structured Synthesis & Scaffolding
Traceability & Evidence Linking
VLM Visual Analysis & Feedback
Reflection & Final Claims
Raw Logs & Data
LATEX
Abstract
Methods
Results
Manuscript
Scientific Claim
Transparency Check
Notebook
Experiment Log
VLM
Fix Axis Label! Align Text & Figure
Coder Robot
Plotting Code
VLM Eye
Draft
Optimism Bias
Reflection Filter
Objectively Grounded Conclusion

Figure 6 Automated result synthesis and visual analysis in ASRS. Structured manuscript scaffolding and evidence-linked synthesis organize raw experimental artifacts into explicit scientific claims, while VLM-based visual analysis provides feedback on figure correctness and text–figure alignment to support transparent and reliable result interpretation.

realistic constraints. Addressing these gaps represents an important direction for future work. Developing more comprehensive evaluation frameworks that capture high-difficulty, open-ended experimental settings will be crucial for accurately assessing the capabilities and limitations of autonomous research agents.

6 Result Interpretation and Claim Formulation

Results synthesis refers to the stage in which ASRS convert low-level experimental outputs, such as execution logs, numerical results, and generated code, into structured and interpretable scientific conclusions. The primary objective of this stage is to present automated discoveries in academically acceptable formats, including research reports or manuscript drafts, while maintaining logical consistency and scientific rigor.

6.1 Results Synthesis

Most existing systems implement results synthesis through highly structured writing workflows. For example, prior work (Schmidgall et al., 2025) introduces a dedicated `paper-solver` module that automatically generates LaTeX manuscripts. The module first creates a fixed scaffold with standard sections (e.g., abstract, introduction, methods, experiments, and discussion), and then iteratively fills and refines each section based on experimental outcomes. A related approach (Gottweis et al., 2025) adopts a hierarchical synthesis strategy, where a meta-review agent periodically aggregates high-performing hypotheses from iterative evaluations into concise research summaries, which can further be formatted into domain-specific documents such as grant proposals.

Beyond document generation, several systems emphasize traceability as a core requirement of results synthesis. In particular, prior work (Mitchener et al., 2025) enforces explicit evidence linking, requiring every claim or figure in the generated report to be directly associated with its originating experiment, notebook, or literature reference. This design improves transparency and allows human researchers to independently verify each step of the automated discovery process.

6.2 Visual Analysis

Visual analysis marks an important shift in ASRS from purely text-based reasoning toward multimodal inspection of experimental evidence. Recent work integrates vision–language models (VLMs) to perform closed-loop validation of figures and plots, ensuring consistency between visual outputs and textual interpretations (Lompo and Haraoui, 2025; Masry et al., 2022; Wang et al., 2024, 2025a). In particular, prior systems (Yamada et al., 2025) employ VLM-based feedback during figure generation, where snapshots of experimental plots are automatically reviewed for basic correctness, such as the presence of axis labels, legends, and alignment between visual trends and underlying numerical execution logs. Visual analysis is further used to assess presentation quality and text–figure alignment by jointly examining figures, captions, and in-text references, enabling automatic refinement of plotting code when visual clarity or semantic consistency is insufficient. Beyond quality control, incorporating visual reasoning into the research loop also helps reduce visual hallucination and can reveal patterns overlooked by text-only analysis (Gottweis et al., 2025).

6.3 Future Work

Despite recent progress, several fundamental limitations remain in the synthesis and interpretation stages of automated scientific research systems. First, automated result synthesis is still prone to systematic bias. Empirical analyses (Lu et al., 2024) reveal a recurring optimism bias, where models tend to over-interpret noisy, inconclusive, or even negative experimental outcomes as meaningful improvements. Although subsequent work has introduced more rigorous post-generation reflection and reasoning mechanisms to improve factual accuracy and objectivity (Yamada et al., 2025), developing principled methods for uncertainty-aware reasoning, negative result analysis, and failure attribution remains an open challenge. Second, visual understanding continues to be a major bottleneck for reliable scientific reasoning. Prior studies show that earlier language models struggle to accurately interpret figures and tables, limiting the usefulness of their analytical feedback (Thakkar et al., 2025b). Consistently, existing automated research frameworks report that AI-generated figures often lag behind human-created visuals in clarity, interpretability, and semantic alignment (Schmidgall et al., 2025). These limitations highlight the need for more robust multimodal reasoning capabilities that tightly integrate visual perception with textual and numerical analysis. Addressing these challenges will be critical for transforming visual analysis from a supportive post-processing step into a fully reliable component of autonomous scientific discovery.

7 End-to-End Paper Generation

End-to-end paper production marks a critical transition of ASRS from task-oriented assistants to autonomous scientific entities (Yu et al., 2025). By integrating ideation, experimentation, and manuscript writing into a unified lifecycle, this stage enables scalable and low-cost generation of scientific outputs, fundamentally changing how research artifacts can be produced.

7.1 System Architectures for Paper Generation

Existing end-to-end ASRS primarily rely on multi-agent collaboration and modular system design to ensure the completeness and consistency of generated papers. A representative example is prior work (Liu et al., 2025; Lu et al., 2024), which introduced the first fully automated pipeline covering idea generation, code development, experimental execution, and manuscript writing. Its successor (Yamada et al., 2025) further improves autonomy by adopting *Agentic Tree Search*, enabling parallel exploration of multiple research trajectories, and by integrating VLMs to iteratively review and align generated figures with their textual descriptions. In contrast, a modular and collaboration-oriented design is adopted in (Schmidgall et al., 2025). This framework decomposes the research workflow into literature review, experimentation, and report writing stages, with a dedicated `paper-solver` module responsible for LaTeX manuscript generation. PhD-level agents and professor-level agents jointly transform experimental results into structured academic reports through iterative refinement.

Beyond these mainstream approaches, alternative end-to-end pathways have also been explored. For example, *data-to-paper* combines multiple language models with rule-based agents to produce verifiable research papers,

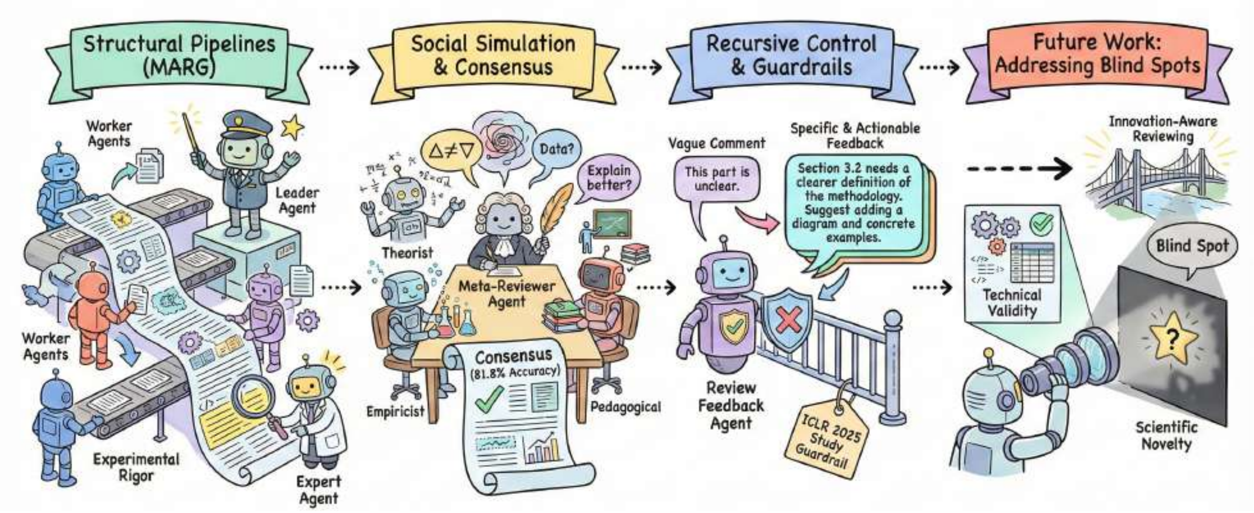

Structural Pipelines (MARG)
Worker Agents
Leader Agent
Worker Agents
Experimental Rigor
Expert Agent
Social Simulation & Consensus
Δ≠∇
Data?
Explain better?
Theorist
Meta-Reviewer Agent
Empiricist
Pedagogical
Consensus (81.8% Accuracy)
Recursive Control & Guardrails
Vague Comment
This part is unclear.
Specific & Actionable Feedback
Section 3.2 needs a clearer definition of the methodology. Suggest adding a diagram and concrete examples.
Review Feedback Agent
ICLR 2025 Study Guardrail
Future Work: Addressing Blind Spots
Innovation-Aware Reviewing
Technical Validity
Blind Spot
Scientific Novelty

Figure 7 Simulated peer review and recursive feedback control in ASRS. The pipeline decomposes scientific evaluation into structural review workflows, social simulation with consensus formation, and recursive feedback with guardrails, enabling systematic critique, refinement, and quality control of autonomous research outputs.

while (Weng et al., 2024) focuses on the trajectory from idea generation to draft writing and deliberately omits experimental execution to reduce system complexity.

7.2 Writing Mechanisms and Academic Standardization

At the technical level, end-to-end systems aim to mimic human writing workflows through structured scaffolding and iterative refinement. A common strategy is to first construct a fixed LaTeX scaffold with standard sections, such as abstract, introduction, methods, experiments, results, and discussion, and then gradually refine the content through targeted editing operations. For example, prior work (Schmidgall et al., 2025) employs a dedicated `paper-solver` module that performs line-level edits to incrementally improve each section. Similarly, the AI Scientist series (Lu et al., 2024; Yamada et al., 2025) adopts Aider-based streaming writing to maintain close alignment between experimental outputs and the evolving manuscript text. To ensure adherence to academic writing norms, several systems further incorporate *style normalization* mechanisms. As explored in prior studies (Si et al., 2024), generated content is rewritten into a consistent scholarly tone, reducing stylistic variance and improving fairness under anonymous peer review. In addition, citation handling and document compilation are largely automated. Systems retrieve relevant references through large-scale literature APIs such as Semantic Scholar and automatically generate BibTeX entries to reduce hallucinated citations. LaTeX compilation errors are detected and fed back to the agents for automatic correction, ensuring that generated manuscripts can be directly compiled into valid PDF documents without manual intervention.

8 Simulated Peer Review & Iteration

Simulated peer review serves as a critical control mechanism in ASRS, aligning autonomous research outputs with established scientific norms through iterative critique and refinement. Large-scale empirical analyses confirm the validity of this approach: LLM-generated feedback can significantly overlap with human reviewer comments (up to 39.23% for ICLR papers) and is often perceived by researchers as helpful or even more beneficial than human feedback (Liang et al., 2024). Current frameworks have evolved from simple generation to complex multi-agent ecosystems that enforce quality through structural hierarchy, social simulation, and recursive feedback loops.

8.1 Structural Pipelines

To manage the complexity of scientific evaluation, recent frameworks decompose the review process into specialized modular workflows. The MARG (D'Arcy et al., 2024) addresses the context limitations of LLMs by employing a hierarchical structure where "Leader" agents coordinate "Worker" agents to process long documents, while specialized "Expert" agents evaluate specific dimensions like experimental rigor. Moving beyond functional decomposition to social simulation, (Sahu et al., 2025) models peer review as a socio-technical process. It instantiates diverse reviewer personas (e.g., "Theorists," "Empiricists," "Pedagogical") and integrates a "Meta-Reviewer" agent that synthesizes consensus from these often conflicting perspectives. Empirical results show that such ensemble-based meta-reviewing significantly outperforms single-agent approaches, achieving decision accuracy comparable to human reviewers (81.8% vs. 83.9%).

8.2 Recursive Feedback and Meta-Control

A crucial aspect of control is ensuring the actionable quality of the feedback itself. Pipelines now incorporate self-correction mechanisms; for instance, MARG includes a dedicated "refinement" stage where initial comments are pruned or revised for clarity and validity before presentation (Sahu et al., 2025). Furthermore, Liang et al. (2024) demonstrate that LLMs can function as effective "meta-reviewers" via a Review Feedback Agent. In a large-scale randomized study at ICLR 2025, this system acted as a guardrail by detecting vague or unprofessional comments and prompting reviewers to be more specific. This recursive control mechanism significantly increased the length and constructiveness of reviews, proving that AI can regulate the quality of the evaluation process itself, not just the scientific content (Thakkar et al., 2025a).

8.3 Future Work

The effectiveness of iterative refinement in ASRS depends critically on the quality and focus of the feedback provided. Recent work shows that LLMs can not only generate reviews but also critique and refine them, acting as meta-reviewers that detect vague or unprofessional comments and encourage more specific, actionable feedback, thereby improving the depth and usefulness of the review process (Thakkar et al., 2025b). However, existing systems exhibit a systematic imbalance in feedback focus. While LLMs are generally effective at evaluating technical validity, such as methodological correctness and experimental details, they show a notable blind spot in assessing scientific novelty. Compared to human reviewers, off-the-shelf LLMs tend to under-emphasize novelty and over-focus on validity (Shin et al., 2025). As a result, although feedback-driven iteration is effective for improving clarity and correcting technical errors, future ASRS must explicitly address this limitation to ensure that simulated peer review supports both rigor and genuine scientific innovation.

9 Conclusion

Our work has presented a unified overview of ASRS from the perspective of the scientific research lifecycle. By decomposing end-to-end automated research into seven interrelated stages, we provide a structured framework for understanding existing system designs, capabilities, and limitations, helping to organize a fragmented body of prior work. Our analysis shows that, despite notable progress, current ASRS remain far from fully autonomous scientific discovery. Persistent challenges—such as the ideation–execution gap, brittle experimental reasoning under real-world constraints, limitations in multimodal understanding, and blind spots in evaluating scientific novelty—indicate that advances in individual modules do not automatically translate into robust end-to-end performance. Looking ahead, we argue that the most practical path forward lies in deep human–AI collaboration rather than full automation. By combining AI-driven large-scale execution and analysis with human judgment and domain expertise, future ASRS can more effectively support scientific discovery. We hope this paper provides a useful foundation for developing more robust, transparent, and scientifically grounded automated research systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H Laradji, Krishnamurthy Dj Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. Litllms, llms for literature review: Are we there yet? *arXiv preprint arXiv:2412.15249*, 2024.

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. Litsearch: A retrieval benchmark for scientific literature search. *arXiv preprint arXiv:2407.18940*, 2024.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. ResearchAgent: Iterative research idea generation over scientific literature with large language models. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6709–6738, 2025.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating machine learning agents on machine learning engineering, 2024a.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024b.

Claude Team. Claude research, 2025. https://www.anthropic.com/news/research.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*, 2024.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Albin Madappallil, Agam Dua, Abhay Gupta, Adam Gaier, Ahmed Aggour, Ahmed Audhkhasi, Adrien Wherry, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Maksim E. Eren and Dorianis M. Perez. Rethinking science in the age of artificial intelligence, 2025.

Gemini Team. Gemini deep research, 2025. https://gemini.google/overview/deep-research/.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an AI co-scientist, February 2025. arXiv:2502.18864 [cs].

Grok Team. Grok-3 deeper search, 2025. https://x.ai/news/grok-3.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. MetaGPT: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*, 2025.

Tu Hu, Ronghao Chen, Shuo Zhang, Jianghao Yin, Mou Xiao Feng, Jingping Liu, Shaolei Zhang, Wenqi Jiang, Yuqi Fang, Sen Hu, Huacan Wang, and Yi Xu. Controlled self-evolution for algorithmic code optimization. *arXiv preprint arXiv:2601.07348*, 2026.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2023.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. AgentReview: Exploring peer review dynamics with LLM agents. *arXiv preprint arXiv:2406.12708*, 2024.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36: 51991–52008, 2023.

Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Yi Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Scott Smith, Yian Yin, et al. Can large language models provide useful feedback on research papers? a large-scale empirical analysis. *NEJM AI*, 1(8):AIoa2400196, 2024.

Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. Automated scholarly paper review: Concepts, technologies, and challenges. *Information fusion*, 98:101830, 2023a.

Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. Moprd: A multidisciplinary open peer review dataset. *Neural Computing and Applications*, 35(34):24191–24206, 2023b.

Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Hongzhang Liu, Ronghao Chen, Yangfan He, Daxin Jiang, Binxing Jiao, Chen Hu, and Huacan Wang. SE-Agent: Self-evolution trajectory optimization in multi-step reasoning with LLM-based agents. *arXiv preprint arXiv:2508.02085*, 2025.

Xiaochuan Liu, Ruihua Song, Xiting Wang, and Xu Chen. Select, read, and write: A multi-agent framework of full-text-based related work generation. *arXiv preprint arXiv:2505.19647*, 2025.

Boammani Aser Lompo and Marc Haraoui. Visual-tableqa: Open-domain benchmark for reasoning over table images. *arXiv preprint arXiv:2509.07966*, 2025.

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

Ziming Luo, Zonglin Yang, Zexin Xu, Wei Yang, and Xinya Du. Llm4sr: A survey on large language models for scientific research. *arXiv preprint arXiv:2501.04306*, 2025.

Bohan Lyu, Xin Cong, Heyang Yu, Pan Yang, Yujia Qin, Yining Ye, Yaxi Lu, Zhong Zhang, Yukun Yan, Yankai Lin, et al. Gitagent: Facilitating autonomous agent with github by tool extension. *arXiv preprint arXiv:2312.17294*, 2023.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279, 2022.

Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, Arvis Sulovari, Eric C. Landsness, Daniel L. Barabasi, Siddharth Narayanan, Nicky Evans, Shriya Reddy, et al. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Sonia Krishna Murthy, Tomer Ullman, and Jennifer Hu. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, page 11241–11258. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.561. http://dx.doi.org/10.18653/v1/2025.naacl-long.561.

Ziyi Ni, Huacan Wang, Shuo Zhang, Shuo Lu, Ziyang He, Wang You, Zhenheng Tang, Yuntao Du, Bill Sun, Hongzhang Liu, Sen Hu, Ronghao Chen, Bo Li, Xin Li, Chen Hu, Binxing Jiao, Daxin Jiang, and Pin Lyu. GitTaskBench: A benchmark for code agents solving real-world tasks through code repository leveraging. *arXiv preprint arXiv:2508.18993*, 2025.

OpenAI. Deep research system card, 2025. https://cdn.openai.com/deep-research-system-card.pdf.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

Toran Bruce Richards. Autogpt. https://github.com/Significant-Gravitas/AutoGPT, 2023. GitHub repository.

Gaurav Sahu, Hugo Larochelle, Laurent Charlin, and Christopher Pal. ReviewerToo: Should AI join the program committee? *arXiv preprint arXiv:2510.08867*, 2025.

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

Hyungyu Shin, Jihoon Kim, Hwaran Lee, Kyohoon Jin, and Seung won Hwang. Mind the blind spots: A focus-level evaluation framework for LLM reviews. *arXiv preprint arXiv:2502.17086*, 2025.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*, 2024.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The ideation-execution gap: Execution outcomes of llm-generated versus human research ideas. *arXiv preprint arXiv:2506.20803*, 2025.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge, September 2024. arXiv:2409.13740 [cs].

Xiangru Tang, Yuliang Liu, Zefan Cai, Yanjun Shao, Junjie Lu, Yichi Zhang, Zexuan Deng, Helan Hu, Kaikai An, Ruijun Huang, et al. Ml-bench: Evaluating large language models and agents for machine learning tasks on repository-level code. *arXiv preprint arXiv:2311.09835*, 2023.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

Naitian Thakkar, Yilun Xu, Shikhar Varma, Ke Wu, Zhaofeng Wang, Dawn Song, Huazhe Xu, Trevor Darrell, Shanghang Wang, and Joseph E Gonzalez. Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025. *arXiv preprint arXiv:2504.09737*, 2025a.

Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025. *arXiv preprint arXiv:2504.09737*, 2025b.

Baode Wang, Biao Wu, Weizhen Li, Meng Fang, Zuming Huang, Jun Huang, Haozhe Wang, Yanjie Liang, Ling Chen, Wei Chu, et al. Infinity parser: Layout aware reinforcement learning for scanned document parsing. *arXiv preprint arXiv:2506.03197*, 2025a.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972): 47–60, 2023.

Huacan Wang, Ziyi Ni, Shuo Zhang, Shuo Lu, Sen Hu, Ziyang He, Chen Hu, Jiaye Lin, Yifu Guo, Ronghao Chen, et al. Repomaster: Autonomous exploration and understanding of github repositories for complex task solving. *arXiv preprint arXiv:2505.21577*, 2025b.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

Penghao Wang, Yuhao Zhou, Mengxuan Wu, Ziheng Qin, Bangyuan Zhu, Shengbin Huang, Xuanlei Zhao, Panpan Zhang, Xiaojiang Peng, Yuzhang Shang, et al. Researchgpt: Benchmarking and training llms for end-to-end computer science research workflows. *arXiv preprint arXiv:2510.20279*, 2025c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

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

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Wenqing Wu, Chengzhi Zhang, and Yi Zhao. Automated novelty evaluation of academic paper: A collaborative approach integrating human and large language model knowledge. *Journal of the Association for Information Science and Technology*, 76(11):1452–1469, 2025.

Zhijian Xu, Yilun Zhao, Manasi Patwardhan, Lovekesh Vig, and Arman Cohan. Can llms identify critical limitations within scientific research? a systematic evaluation on ai research papers. *arXiv preprint arXiv:2507.02694*, 2025.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search, 2025.

Shuo Yan, Ruochen Li, Ziming Luo, Zimu Wang, Daoyang Li, Liqiang Jing, Kaiyu He, Peilin Wu, George Michalopoulos, Yue Zhang, et al. Lmr-bench: Evaluating llm agent's ability on reproducing language modeling research. *arXiv preprint arXiv:2506.17335*, 2025.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2022. https://arxiv.org/abs/2210.03629.

Tian Yu, Ken Shi, Zixin Zhao, and Gerald Penn. Multi-agent based character simulation for story writing. In *Proceedings of the Fourth Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2025)*, pages 87–108, 2025.

Xiao Yu, Lei Liu, Xing Hu, Jacky Wai Keung, Jin Liu, and Xin Xia. Where are large language models for code generation on github? *arXiv preprint arXiv:2406.19544*, 2024.

Haoxuan Zhang, Ruochi Li, Yang Zhang, Ting Xiao, Jiangping Chen, Junhua Ding, and Haihua Chen. The evolving role of large language models in scientific innovation: Evaluator, collaborator, and scientist. *arXiv preprint arXiv:2507.11810*, 2025a.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. CodeAgent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. *arXiv preprint arXiv:2401.07339*, 2024.

Ming Zhang, Kexin Tan, Yueyuan Huang, Yujiong Shen, Chunchun Ma, Li Ju, Xinran Zhang, Yuhui Wang, Wenqing Jing, Jingyi Deng, et al. Opennovelty: An llm-powered agentic system for verifiable scholarly novelty assessment. *arXiv preprint arXiv:2601.01576*, 2026.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating creativity and diversity in language models. *arXiv preprint arXiv:2504.05228*, 2025b.

Zhenzhen Zhuang, Jiandong Chen, Hongfeng Xu, Yuwen Jiang, and Jialiang Lin. Large language models for automated scholarly paper review: A survey. *Information Fusion*, page 103332, 2025.

Appendix