Review

Multi-Agent AI Systems for Biological and Clinical Data Analysis

Jackson Spieser , Ali Balapour , Jarek Meller , Krushna Patra , Behrouz Shamsaei *

Posted Date: 30 December 2025

doi: 10.20944/preprints202512.2602.v1

Keywords: multi-agent systems; large language models (LLMs); biomedical AI; clinical decision support; orchestration frameworks; autonomous agents; AI safety; collaborative intelligence

 doi:10.20944/preprints202512.2602.v1

Review

Multi-Agent AI Systems for Biological and Clinical Data Analysis

Jackson Spieser [1], Ali Balapour [2], Jarek Meller [3,4,5,6,7], Krushna Patra [8] and Behrouz Shamsaei [3,4,*]

[1] University of Cincinnati, College on Medicine Cincinnati, Ohio
[2] School of Computing and Analytics, Northern Kentucky University, Highland Heights, Kentucky
[3] Department of Biostatistics, Health Informatics and Data Sciences, University of Cincinnati College of Medicine, Cincinnati, Ohio
[4] Division of Biostatistics and Bioinformatics, Department of Environmental and Public Health Sciences, University of Cincinnati College of Medicine, Cincinnati, Ohio
[5] Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, Ohio
[6] Institute of Engineering and Technology, Faculty of Physics, Astronomy and Informatics, Nicolaus Copernicus University, Torun, Poland
[7] Department of Computer Science, University of Cincinnati College of Engineering and Applied Sciences, Cincinnati, Ohio
[8] Department of Cancer Biology, University of Cincinnati College of Medicine, Cincinnati, Ohio
* Correspondence: Behrouz.shamsaei@uc.edu

Abstract

Multi-agent AI systems, where multiple specialized agents collaborate, are emerging as a powerful approach in biomedicine to tackle complex analytical and clinical tasks that exceed the scope of any single model. **Background:** *This review* outlines how orchestrating large language model (LLM) based agents can improve performance and reliability in biomedical data analysis. It surveys new frameworks that coordinate agent teams and highlights state-of-the-art applications in domains such as drug discovery, clinical trial matching, and decision support, where early multi-agent prototypes have achieved higher accuracy or more robust results compared to lone LLMs. **Methods:** We synthesize findings from recent studies and architectures, categorizing applications and examining how agents divide labor, use tools, and cross-verify each other's outputs. **Results:** The review finds that multi-agent strategies yield notable advantages – for example, reducing errors via inter-agent checking and providing more explainable reasoning through transparent dialogues. We also catalog available orchestration platforms and benchmarks driving this field. **Conclusions:** While multi-agent AI shows promise in augmenting biomedical research and healthcare (by integrating diverse knowledge sources and simulating collaborative problem-solving), ensuring its *reliable and ethical deployment* will require addressing challenges in verification, scalability, continual learning, and safety. The paper concludes that with careful design and rigorous evaluation, AI agent teams could significantly enhance biomedical intelligence without replacing human experts.

Keywords: multi-agent systems; large language models (LLMs); biomedical AI; clinical decision support; orchestration frameworks; autonomous agents; AI safety; collaborative intelligence

1. Introduction

Multi-agent artificial intelligence (AI) systems, in which multiple autonomous agents collaborate as a team, are emerging as a powerful paradigm for biomedical and clinical data analysis [1,2]. Instead of relying on a single, all-purpose model, these systems orchestrate specialized AI agents, each with its own distinct knowledge base or tool, to perform complex tasks in a coordinated manner. A recent survey highlights this trend: large language model (LLM)-based autonomous agents can achieve greater flexibility and robustness by dividing labor among sub-agents [1,3]. In various domains,

 doi:10.20944/preprints202512.2602.v1

genomics, electronic health records, imaging, and beyond, multi-agent approaches have already demonstrated improved performance, reliability, and explainability compared to single-agent methods. For example, collaborative agent teams have produced superior results on complex bioinformatics analyses and clinical decision support challenges where no single model excels [1,4]. Use cases range from automated information extraction in literature mining to multi-step clinical reasoning with tool-assisted LLMs [5,6].

Despite these promising developments, there remains a lack of domain-specific reviews that examine how emerging multi-agent architectures are tailored for use in biomedicine. As AI tools become increasingly integrated into scientific and clinical workflows, synthesizing the current design patterns, orchestration strategies, and safety mechanisms is critical to guiding future research and deployment. At the same time, the rise of multi-agent systems introduces new challenges around communication protocols, shared memory, and guardrails for safety and accountability—all of which remain areas of active debate. For instance, while some researchers favor deterministic graph-based control structures to ensure reproducibility and auditability, others advocate for more flexible, emergent teamwork models that trade predictability for adaptability. These tensions underscore the need for a comprehensive, critical synthesis of the field.

This article provides a comprehensive review of multi-AI-agent orchestration in biomedicine and health. Section 2 defines fundamental concepts that constitute an AI agent and describes frameworks for orchestrating multiple agents. We then survey the state-of-the-art applications in Section 3, covering both basic science domains (genomics, drug discovery, etc.) and clinical domains (imaging, clinical trials, decision support). Section 4 discusses emerging opportunities and underexplored areas such as agent-assisted cancer research, data augmentation, and education. In Section 5, we summarize available platforms, toolkits, and benchmarks that support multi-agent development and evaluation. We then examine key challenges and future directions in Section 6, including reliability and verification of agent reasoning, scalability and efficiency issues, continual learning and adaptation, and ethical/regulatory considerations.

We argue that the orchestration layer—not merely the agent design itself—is increasingly the critical determinant of system performance, safety, and clinical relevance. By synthesizing recent literature, we aim to provide a timely and scholarly overview of how multi-agent systems can transform biomedical data analysis and what hurdles must be overcome for safe, effective deployment.

2. Definitions and Frameworks

2.1. Agent Definitions and Orchestration Frameworks

In artificial intelligence, an *agent* is commonly defined as an autonomous software entity that perceives its environment, makes decisions, and acts toward specific goals. Modern formulations emphasize that an agent is "an autonomous and collaborative entity, equipped with reasoning and communication capabilities, capable of dynamically interpreting contexts, orchestrating tools, and adapting behavior through memory and interaction across distributed systems." Each agent possesses its own knowledge base and skills (e.g., domain expertise, task-specific tools) and can operate independently, adapting its actions based on observations. When multiple such agents work together, the result is a multi-agent system (MAS). Compared to a single-agent setting, a MAS more realistically represents complex real-world scenarios involving multiple decision-makers or information sources [7,8]. Agents in an MAS typically have localized perceptions (each sees only part of the state) and must communicate or coordinate with others to achieve broader objectives. This distributed, team-based approach mirrors human teams, where each member has specialized roles and partial information, requiring collaboration to succeed.

Multi-agent orchestration frameworks refer to the architectures and toolkits that facilitate coordination among multiple AI agents. These frameworks define how agents communicate, delegate tasks, and merge their results into a coherent solution [9,10]. Often, certain agents are

 doi:10.20944/preprints202512.2602.v1

designated as planners or managers who break down a user's request into sub-tasks and assign them to specialist agents. For example, the HuggingGPT framework uses an LLM as a central controller agent to orchestrate numerous expert models (for vision, speech, etc.) as collaborative executors [11,12]. Similarly, Microsoft's AutoGen system enables multi-agent conversations by allowing agents to spawn new agents and exchange messages to solve tasks cooperatively [13]. The open-source CAMEL toolkit provides templates for hierarchical agent societies (e.g., a master–worker structure where a manager agent plans high-level strategy, worker agents execute subtasks, and a reviewer agent checks outputs) [14].

LangGraph is an open framework that models multi-agent workflows as a stateful **graph**, giving developers fine-grained control over execution paths. Agents and tools are represented as nodes, with directed edges dictating the sequence and branching of tasks [15]. This graph architecture enables advanced control-flow structures: developers can include conditional branches, loops, and even parallel paths to handle complex scenarios [16]. Unlike linear agent pipelines, LangGraph maintains **persistent state** throughout the graph – it integrates memory into its nodes, storing intermediate results and context across steps. This ensures that agents always have up-to-date information as they collaborate on a task, preventing loss of context in long workflows. LangGraph's design thus supports diverse control flows (single-agent, multi-agent, hierarchical, sequential) and robustly manages realistic, long-running interactions without losing context. Its explicit state management and *deterministic* execution make it especially suitable for high-stakes or regulated domains where reproducibility is vital. For example, one case study applied LangGraph in an intelligent transportation system for modular traffic management – a scenario requiring reliable branching decisions and strict adherence to protocol. By combining flexible graph transitions with built-in memory, LangGraph delivers **transparent and fault-tolerant orchestration**. Every decision point is predefined by the developer, yielding traceable workflows with rigorous control over errors and edge cases. In summary, LangGraph provides a graph-driven "cognitive architecture" for LLM agents, with first-class support for custom control flow, state persistence, and even the option to include human-in-the-loop review steps (e.g. a node requiring human approval) to keep agent operations on course [15].

In these architectures, communication protocols, often via natural language messages or structured data formats (like JSON), allow agents to share intermediate results and requests. Modern frameworks handle low-level details such as message passing, context management (maintaining each agent's state or memory), and tool integration (e.g., database queries, web search, or code execution) so that developers can focus on high-level agent logic [10,12]. Crucially, the frameworks ensure that the right knowledge or tool is invoked by the right agent at the right time, enabling complex multi-step reasoning that a single model might struggle with.

CrewAI takes a complementary approach by orchestrating a "crew" of AI agents in well-defined **roles** (e.g. planner, executor, reviewer) to tackle tasks collaboratively. Inspired by human team structures, CrewAI allows developers to assign each agent a specialized role with a specific objective and toolset [15]. For example, a *Planner* agent may break down a complex goal into sub-tasks, an *Executor* agent carries out the actions or API calls, and a *Reviewer/Critic* agent evaluates outputs against the requirements, mirroring an interdisciplinary project team. This role-based orchestration ensures structured coordination: agents communicate and delegate tasks according to their expertise, which avoids duplication of work and reduces conflicting actions. CrewAI provides high-level primitives (Agent, Task, Tool, Crew) to define these multi-agent teams and manages the messaging and turn-taking between roles. It natively supports features like automatic task assignment to the appropriate role, integration with various LLM APIs (e.g. OpenAI, Ollama), and flexible task management for complex multi-step workflows [16]. Notably, CrewAI's lightweight, Python-based architecture emphasizes efficiency and developer control – users can even inject custom logic at both the crew level and the agent level to fine-tune behaviors, balancing agent autonomy with oversight. Early applications of CrewAI highlight its teamwork-oriented design: for instance, in an **automated travel planning** scenario, one agent plans an itinerary, another executes search queries, and a third

 doi:10.20944/preprints202512.2602.v1

verifies the plan – the agents collaboratively analyze city data and plan routes, improving the final itinerary's quality. This division of labor, guided by defined roles, prevents any single agent from dominating the process and allows specialized reasoning (e.g. the Reviewer catching errors the Executor might miss). Overall, CrewAI excels at multi-agent **collaboration** through clear role delegation, making it a powerful library for building AI "teams" that require coordinated planning, critique, and execution [15].

LangGraph and CrewAI represent two leading paradigms for LLM-based multi-agent orchestration, each with distinct strengths. LangGraph treats an agent system as an explicit **state machine**, where each agent/tool is a node in a directed graph and all possible transitions are predetermined by the graph's structure. This yields **deterministic and traceable** workflows: every decision point (edge) is defined by the developer in advance, enabling rigorous control, error handling, and reproducibility even across complex branching or looping paths. In contrast, CrewAI emphasizes dynamic **role delegation** – it orchestrates agents through flexible role interactions rather than a fixed graph of steps. The CrewAI approach is more *emergent*: agents autonomously pass tasks among specialized roles based on the situation, allowing adaptive coordination instead of following a single hard-coded sequence [15,16]. State management in LangGraph is handled via its built-in persistent memory and context passing along the graph (each node can store and retrieve shared state), whereas in CrewAI, state is often maintained through the agents' shared memory or message exchanges within the crew, with less centralized control. In practice, LangGraph **excels when a workflow demands strict control-flow logic**, complex conditional branches, or integration with external processes that must follow a defined protocol (e.g. a formal decision tree in a compliance setting). CrewAI, on the other hand, shines in scenarios requiring **collaborative problem-solving and division of labor** – its role-based agents can brainstorm, critique, and refine each other's outputs in a manner akin to an interdisciplinary team, which is valuable for open-ended tasks or those benefiting from multiple perspectives. It's worth noting that these frameworks are not mutually exclusive. In fact, recent research demonstrates that combining LangGraph's graph-centric planning with CrewAI's team coordination can yield powerful solutions, marrying precise control with autonomous collaboration. For example, one can use LangGraph to outline a high-level plan (ensuring determinism in the overall workflow) while employing CrewAI agents at certain nodes to cooperatively solve sub-tasks. Choosing between LangGraph and CrewAI (or using both in tandem) thus depends on the use-case: highly-structured or safety-critical tasks may benefit from LangGraph's reliability and explicit flow control, while creative, exploratory, or expertise-driven tasks leverage CrewAI's flexible role orchestration for better results [15,16].

In summary, multi-agent orchestration frameworks provide the infrastructure and design patterns (messaging systems, shared memory stores, standardized agent interfaces, etc.) to build systems where multiple AI agents collaborate and solve data-analysis problems more effectively than any individual agent alone [9,17]. This paradigm builds upon advances in LLM prompting and tool use. Techniques like ReAct, which interleaves logical Reasoning and concrete Acting steps by an LLM, demonstrated how language models can use external tools dynamically [18]. Similarly, methods such as Toolformer showed that LLMs can learn to call external APIs to aid in problem-solving [19]. These capabilities are often incorporated into agent frameworks. For example, an agent may employ a ReAct-style chain-of-thought to decide when to query a database or run a simulation. Multi-agent systems can also leverage strategies developed for single-agent prompting. For instance, an agent team can adopt a self-consistency approach where multiple agents independently work on the same problem and then vote or converge on an answer [20]. Agents can also implement a Tree-of-Thoughts strategy, collectively exploring different reasoning branches in parallel and exchanging information to decide on the best path forward [19,21]. Furthermore, various feedback and self-correction techniques have been introduced to improve LLM reliability, such as Reflexion and Self-Refine [22,23]. In this approach, a model iteratively critiques and revises its own answers. In a multi-agent context, such guardrails can be implemented by assigning a dedicated critic or reviewer agent to evaluate and correct the outputs of other agents [24,25]. By combining these approaches,

 doi:10.20944/preprints202512.2602.v1

hierarchical planning, tool use, parallel problem solving, and iterative self-correction, current frameworks are beginning to enable autonomous agent teams that are more robust and capable than any single AI agent. Figure 1 provides a high-level overview of a typical multi-agent system architecture, highlighting the role of controller agents and orchestration frameworks such as LangChain and CrewAI in managing specialized sub-agents.

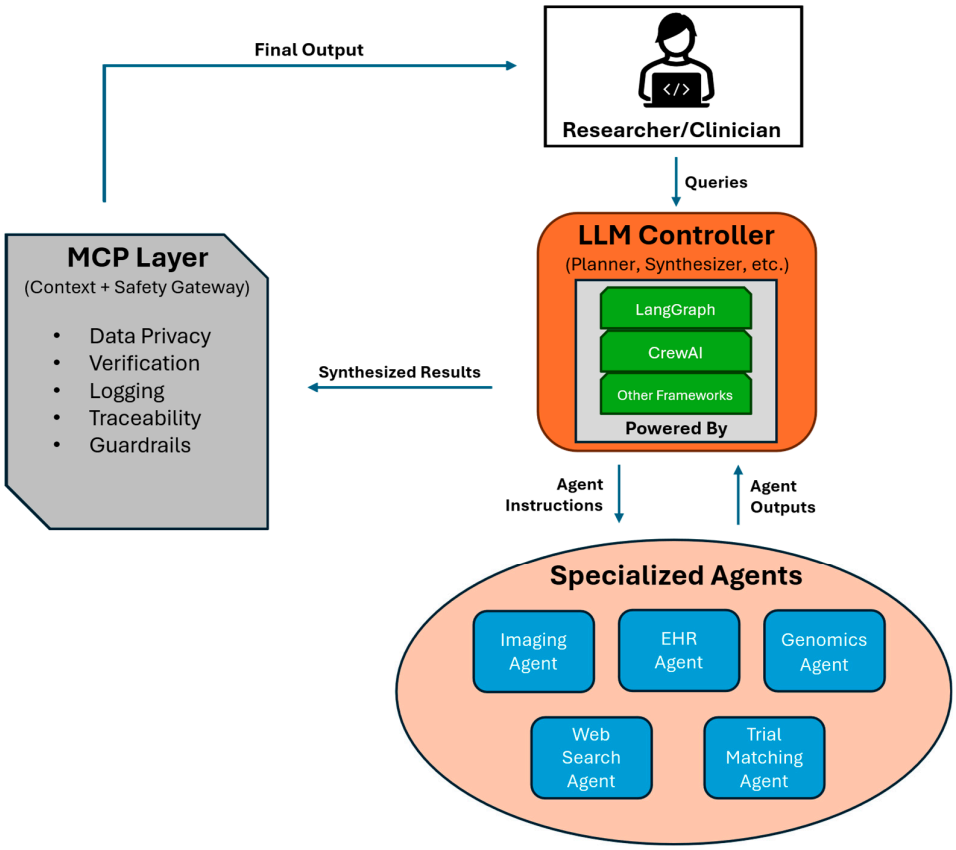

Figure 1. Conceptual architecture of a multi-agent system (MAS). A central controller orchestrates multiple specialized AI agents that interact with tools and external environments. Frameworks such as LangGraph and CrewAI support the orchestration of these agents by managing task delegation, communication, and memory.

2.2. Memory, Guardrails, and Communication Protocols

Effective MAS require mechanisms for memory sharing, safety constraints, and structured inter-agent communication. Memory in a multi-agent context can refer to each individual agent's internal context as well as a shared memory accessible to all agents. Modern agent frameworks often implement a shared knowledge store or blackboard that agents can read from and write to during collaboration [17]. This ensures that facts discovered by one agent (e.g. a relevant patient finding or an experimental result) persist and inform the others. For instance, agents working together could collectively build a knowledge graph or shared database of intermediate results as they progress through a task. Designing effective memory architecture is an active area of research. Challenges include how to retrieve relevant past information when needed and how to prevent context overload or forgetting important facts. Some systems use embedding-based vector memory to allow agents to recall long-term knowledge, while others use explicit storage of key-value records. As a simple example in biomedical NLP, one agent could store extracted patient information in a shared record so that another agent later can query "Has symptom X been noted?" instead of re-parsing the raw text. Shared memory also facilitates experience replay, where agents can learn from prior cases by

 doi:10.20944/preprints202512.2602.v1

referencing how similar problems were solved, improving continual learning (this is discussed further in Section 6.3).

Beyond memory, multi-agent systems must be engineered with guardrails to ensure safety and reliability. A team of agents autonomously generating and executing complex action sequences raises the risk that errors or undesirable behaviors could propagate through the system. One safeguard is to include a specialized monitoring or moderation agent that watches the agents' interactions and intervenes if policies are violated (for example, halting the system if a medical-advice agent attempts an unsafe recommendation). Alignment techniques from single-agent LLMs, such as content filtering and reinforcement learning from human feedback (RLHF), can be extended to multi-agent settings by applying them to each agent's outputs or decisions. Notably, in 2023, Google introduced an Agent-to-Agent (A2A) communication protocol as an open standard for LLM-based agents to exchange messages in a structured, secure manner [26,27]. Efforts like A2A, and related proposals for standardized Agent Communication Protocols, aim to define common message formats (e.g. standardized JSON schemas or API calls) to enable interoperability and security in multi-agent ecosystems. However, these new protocols also introduce security considerations that must be addressed. An analysis of the A2A framework identified potential attack vectors (e.g. prompt injection, token reuse, agent impersonation) and emphasized the need for agent authentication and permission controls [28]. Ensuring secure and authenticated inter-agent communication will be vital so that a rogue agent cannot impersonate a trusted one or leak sensitive data. In general, multi-agent systems will require defense-in-depth: sandboxing of agent actions, well-defined scopes for each agent's authority, and continuous monitoring for anomalous behaviors. We revisit safety and security issues in Section 6.1.

3. State of the Art in Biomedicine

Multi-agent AI has rapidly gained traction as a strategy for tackling the complexity of biomedical data analysis and clinical decision-making. Researchers have already demonstrated that orchestrating multiple specialized agents can outperform single-model approaches on challenging tasks [29,30]. In this section, we survey representative state-of-the-art systems in both basic science and clinical domains. We organize the discussion into use-case categories, while noting that many systems span multiple categories. Throughout, we highlight how multi-agent designs are enabling new capabilities, from automating bioinformatics pipelines to running virtual clinical trials, and improving performance (e.g. accuracy, efficiency, interpretability) relative to prior methods.

3.1. Basic Science Applications

3.1.1. Drug Discovery and Pharmacology

Drug discovery is a complex, multi-factorial process that can benefit from the division of labor offered by multi-agent AI. Early examples of multi-agent systems in this domain show promise in integrating chemical databases, biological knowledge, and simulation tools to assist in identifying therapeutic targets or compounds. For instance, GPCR-Nexus is a multi-agent system that focuses on G protein-coupled receptors (GPCRs), a large family of drug targets [31]. GPCR-Nexus employs an agentic OmniRAG approach: it orchestrates a team of agents that combine knowledge-graph traversal with retrieval-augmented text generation to answer pharmacological queries about GPCRs. One agent, a source planner, decomposes a user's question into sub-queries for a literature search and a knowledge-graph lookup, while other agents execute those searches in parallel. A reviewer agent then filters and cross-checks the retrieved results for relevance and factual accuracy, and a synthesizer agent compiles a final answer with supporting evidence. By dividing labor (literature retrieval, structured data query, fact-checking, synthesis) among specialized agents, GPCR-Nexus produces context-rich, evidence-backed answers to questions that neither a conventional search engine nor a single LLM alone could easily handle. This case exemplifies how multi-agent systems can integrate structured knowledge bases with unstructured text mining to support drug discovery

 doi:10.20944/preprints202512.2602.v1

research by answering questions about receptor-ligand interactions, signaling pathways, and related clinical trials. Notably, the incorporation of a knowledge-graph agent helps reduce factual errors by grounding answers in curated databases, addressing a key limitation of standalone LLMs.

Another effort aimed at aiding therapeutic development is TxAgent (short for "Therapy Agent"). Introduced in 2025, TxAgent is an AI system that leverages multi-step reasoning and real-time knowledge retrieval across an extensive toolbox of domain-specific models (over 200 specialized tools in total) to support treatment discovery and optimization. In the prototype described by Gao *et al.* (2025), TxAgent can retrieve and synthesize evidence from multiple biomedical sources (e.g. drug-gene interaction databases, clinical guidelines, patient health records) and simulate the effects of drug combinations [32,33]. It assesses potential interactions between drugs and patient conditions, and iteratively refines treatment strategies using a reasoning loop. Although TxAgent is presented as a single composite agent, internally it functions as a multi-agent or modular system: different sub-modules handle tasks like chemical property prediction, pathway simulation, and literature scanning [32,33]. This highlights a continuum between a "single agent with many tools" and an "agent society". In practice, a highly modular single agent (with distinct tool-using subroutines) begins to resemble a coordinated multi-agent team. TxAgent's impressive results on test cases (e.g., proposing treatment plans that align with expert clinician recommendations) underscore the value of integrating diverse expert capabilities. The system essentially serves as a virtual pharmacologist, and its design suggests a roadmap for multi-agent AI in drug discovery: one can imagine an expanded team where separate agents take on roles like medicinal chemist designing novel compounds, toxicologist predicting safety profiles, clinical trial expert assessing trial feasibility, and so on, all coordinated to accelerate the drug development pipeline.

It is important to note that multi-agent AI in drug discovery is still in early stages, and most systems are prototypes or proofs-of-concept. Nonetheless, these examples illustrate the potential. By combining knowledge-driven agents (for biology and chemistry) with reasoning agents and simulation tools, multi-agent systems could, in the future, automate hypothesis generation for new drug targets, perform in silico screening of large compound libraries, and design optimal preclinical experiments. Moreover, multi-agent setups can naturally incorporate feedback loops with human scientists. For example, an agent team might propose a list of candidate molecules and then adapt its strategy based on a medicinal chemist's feedback on which candidates are synthesizable. This collaborative human–AI approach aligns well with the iterative nature of pharmacological research.

3.1.2. Bioinformatics and Multi-Omics Analysis

Bioinformatics was one of the first areas to embrace multi-agent orchestration, given the inherent complexity of biological data-processing pipelines. A landmark example is the BioMaster system by Su *et al.*, 2025, a multi-agent framework for automated multi-omics workflows [35,36]. BioMaster integrates several specialized agents to plan and carry out genomic data analyses. For instance, a Planner Agent first decomposes a high-level bioinformatics task, such as "identify differentially expressed genes from these RNA-seq samples", into a sequence of subtasks. Next, a Task-Executor Agent translates each subtask into concrete commands or API calls to bioinformatics software (e.g., running quality control, read alignment, variant calling, or statistical analysis). A dedicated Debug Agent monitors the pipeline for errors or suboptimal results (such as a file format issue or poor sequence coverage) and can intervene to adjust parameters or retry steps when needed [35]. Meanwhile, a Data-Agent handles intermediate data caching and formatting between pipeline stages, and a Reviewer Agent validates final outputs, which includes checking whether the list of differentially expressed genes makes sense given known biology. BioMaster is equipped with dual retrieval-augmented generation modules: one that consults domain-specific knowledge, like method databases or prior experiments, to assist the Planner in choosing the best tools and parameters, and another that helps the Debug Agent find solutions when a pipeline step fails. In benchmarking across 49 representative tasks spanning 18 omics modalities and 102 distinct bioinformatics tools, BioMaster completed substantially more analysis workflows than a baseline automated pipeline, especially on

 doi:10.20944/preprints202512.2602.v1

complex, multi-step analyses with interdependent tasks [36]. It has been demonstrated using both proprietary large LLMs and open-source models, indicating flexibility in the agent framework. These results show that multi-agent designs can bring robustness and adaptability to bioinformatics by uniting planning, execution, error-recovery, and validation agents; the system can handle data-processing challenges with minimal human intervention. This approach addresses inefficiencies in traditional pipelines, which often break when assumptions are violated, by enabling agents to detect and correct errors on the fly. As multi-agent frameworks like BioMaster mature, they could help democratize bioinformatics, allowing labs to input raw data and receive analyzed results with an intelligent agent team managing the entire workflow.

3.1.3. Cancer Biology

Cancer research and oncology are increasingly data-driven, and multi-agent AI systems are beginning to assist in these domains as well. One notable example is an AI-based "virtual tumor board" system for clinical decision-making in oncology. Ferber *et al.* (2025) developed an autonomous AI agent that effectively replicates a multi-disciplinary tumor board discussion by integrating multiple specialized agents for different data modalities and decision facets [36,37]. The system leverages GPT-4 as a core reasoning engine and incorporates vision transformers for pathology image analysis, genomic predictors for molecular profiling, and clinical knowledge databases for treatment guidelines [36,37]. In a validation on retrospective oncology cases, the AI agent was able to analyze a patient's pathology slides, genetic alterations, and medical history, then suggest a treatment plan with supporting rationale. Its recommendations were concordant with those of human tumor boards in a significant fraction of cases, demonstrating the potential for multi-agent AI to support personalized cancer therapy decisions. Beyond clinical decision support, multi-agent approaches are also being explored in cancer research. For instance, agents can be assigned to scour different knowledge sources (one agent reading new papers on a specific cancer gene, another mining omics databases, etc.) and then collaboratively generate hypotheses. For example, the information could be amalgamated to propose a novel drug target or explain a mechanism of drug resistance. Such an agent ensemble can efficiently sift through the exploding volume of cancer data and literature. Multi-agent systems can also simulate in silico experiments: one agent could propose an experiment, such as testing a certain drug on a cancer cell line, another agent virtually "performs" the experiment by running bioinformatics simulations, and a third agent analyzes the simulated results to refine the hypothesis. Early prototypes of these research-oriented agent teams have been reported [29,30]. While these are in nascent stages, they point toward a future where AI agents collaborate with oncologists and cancer biologists to generate insights, freeing human experts to focus on designing creative experiments and interpreting high-level outcomes.

3.2. Clinical Applications

3.2.1. Medical Imaging and Multimodal Diagnosis

Clinical diagnosis often requires integrating information from multiple sources, like physical exam findings, lab results, and medical images. Multi-agent AI is naturally suited to such multimodal reasoning: different agents can specialize in processing different data types and then collaborate to form a comprehensive assessment. A recent study by Chen et al. (2025) provides a striking example. The authors developed a Multi-Agent Conversation (MAC) framework for differential diagnosis, inspired by how human physicians consult with one another in multi-disciplinary team meetings [38]. In their system, four AI "doctor" agents, each with a slightly different knowledge focus or reasoning style, discuss a patient case, and a fifth agent acts as a moderator to coordinate the conversation. Using a set of 302 challenging rare-disease cases, they evaluated GPT-4 alone versus GPT-4 augmented with the multi-agent conversation framework. The multi-agent approach significantly outperformed the single model in both initial diagnosis and follow-up questioning, achieving higher accuracy in identifying the correct diagnosis and suggesting appropriate diagnostic

 doi:10.20944/preprints202512.2602.v1

tests [38,39]. Optimal performance was achieved with a team of four specialist agents plus one moderator agent. Notably, the MAC framework also outperformed other prompting strategies like chain-of-thought, self-refinement, and self-consistency, which highlights that structured agent collaboration provided an additional boost in diagnostic capability beyond what those single-model methods achieved. Qualitatively, the dialogues between the AI "doctors" demonstrated each agent bringing up different aspects. One agent might recall a rare genetic syndrome fitting some of the symptoms, while another points out a lab abnormality that contradicts that hypothesis, leading to a more thorough analysis than a single model's monologue. This multi-agent discussion approach, akin to an AI panel of consultants, offers a promising path to make medical AI more reliable and transparent. Beyond rare diseases, similar agent teams could be deployed for more common diagnostic workups. For instance, an imaging specialist agent could interpret a patient's radiology scans while a generalist agent correlates those findings with lab results and clinical history, and together they reach a consensus. Early results indicate that having multiple agents debate and verify each other's conclusions reduces diagnostic oversight and instills more confidence in the final recommendations [38,39].

3.2.2. Clinical Trials and Evidence Synthesis

Another area where multi-agent AI is making inroads is in clinical trial matching and evidence aggregation. Identifying suitable clinical trials for a patient, or conversely, finding patients for a trial, is essentially an exercise in complex criteria matching and data retrieval, which can be enhanced by multiple collaborating agents. A pioneering system in this realm is TrialGPT, introduced by Jin et al. (2024). TrialGPT uses a trio of agents to match patients to clinical trials: one agent, TrialGPT-Retrieval, first conducts a large-scale search to retrieve candidate trials from databases based on a patient's profile; a second agent, TrialGPT-Matching, then evaluates the patient's eligibility against each trial's criteria using LLM-based reading and annotation of inclusion/exclusion criteria; finally, a third agent, TrialGPT-Ranking, assigns scores to rank the trials by suitability [40]. In tests on cohorts of synthetic patient records, TrialGPT's retrieval agent could recall over 90% of truly eligible trials while screening out ~94% of ineligible trials in the initial pass. Its matching agent achieved 87.3% accuracy in determining patient eligibility on a fine-grained criterion level, which approaches expert-clinician performance, and produced natural-language justifications for each decision. Overall, TrialGPT's ranked trial recommendations outperformed prior rule-based and ML-based methods by a substantial margin. Furthermore, a user study revealed that doctors using TrialGPT were able to complete trial screening tasks 42.6% faster, on average, than without AI's assistance. These results exemplify how multi-agent LLM systems can tackle the tedious task of parsing lengthy trial protocols and comparing them to patient data, a task that traditionally consumes significant human coordinator time.

Beyond matching patients to trials, multi-agent systems can assist in clinical trial design and evidence synthesis. Agents could form a virtual committee to design a new trial protocol: for example, a Protocol-Writing Agent drafts a trial plan based on identified gaps in current research, a Critic Agent, encoded with ethical and feasibility rules, reviews the draft for potential issues and biases, and a Prior-Studies Agent searches for similar past trials to ensure novelty [41]. By iterating through these roles, the agent team can refine a trial proposal that a human investigator can then consider. This mirrors how human researchers brainstorm and critique protocols, and could inspire more innovative trial designs. While still experimental, the idea has merit: AI agents excel at rapidly scanning large knowledge bases (like all trials done in a certain disease) to find what has or hasn't been tried, which can spark new trial ideas or help avoid duplication. Early steps in this direction are already evident. For instance, a research assistant agent named DORA was able to automatically draft biomedical research project proposals that were later refined by humans [41]. This suggests that similar agents could help assemble clinical trial protocols or grant applications. Likewise, multi-agent systems can serve as evidence synthesizers for clinicians. One prototype system employs an Evidence Retrieval Agent to gather all published studies comparing certain treatments, and a Summary Agent

 doi:10.20944/preprints202512.2602.v1

to distill their findings into a concise report for a physician. In tests, this agent team could answer complex questions about comparative efficacy by synthesizing data across multiple studies, with higher accuracy and less hallucination than a single LLM working alone. As the body of medical literature continues to grow, such multi-agent evidence summation will be invaluable to support evidence-based practice.

In summary, multi-agent systems are proving valuable in navigating the maze of clinical trial data and medical literature. Agents for trial matching can systematically interpret both patient data and trial criteria with systems like TrialGPT already attaining high accuracy, and agents for literature review can aggregate findings from numerous studies into coherent conclusions. By dividing tasks, searching, extracting, comparing, and summarizing, agents collaboratively produce comprehensive outputs that account for the full breadth of available evidence. This capability is increasingly crucial in medicine, where practitioners must keep up with rapidly evolving information. Multi-agent AI can act as an ever-vigilant research assistant, ensuring that decisions (like enrolling a patient in a trial or choosing a therapy) are informed by the latest and most relevant evidence: something that human clinicians, burdened with information overload, would certainly welcome.

3.2.3. Clinical Decision Support and Physician Assistants

One of the most impactful applications of multi-agent AI is in clinical decision support, or assisting physicians for diagnosis, treatment planning, and overall patient management. The complexity of real clinical cases, especially in fields like internal medicine, often requires gathering disparate information, reasoning through possible diagnoses, consulting guidelines, and double-checking for errors or contraindications. Multi-agent systems are naturally suited to handle these subtasks in parallel and provide a "second set of eyes" on each other's work, thereby acting as a tireless medical assistant or consulting team for a physician.

A compelling example is the Agent Hospital framework (Li *et al.*, 2024), which created a virtual hospital environment populated by multiple AI agents adopting the roles of doctors, patients, and nurses [42,43]. In this simulation, doctor agents took patient histories from patient agents via natural-language dialogue, decided on which tests to order, made diagnoses, and recommended treatments, all through iterative conversations and reasoning in a hospital-like setting. To train these agents, the researchers developed a special multi-agent training algorithm (called MedAgent-Zero) and exposed the agents to thousands of simulated clinical cases. The result was remarkable. The AI "doctors" achieved about 93% accuracy on a standardized medical exam (answering USMLE-style diagnostic questions), surpassing many prior single-model approaches. By orchestrating realistic multi-agent interactions (doctor-patient interviews, doctor-doctor discussions, etc.), the system effectively learned medical reasoning in a way that generalized to exam questions. The key point is that the medical knowledge and problem-solving ability emerged from the multi-agent simulation: the doctor agents improved by "practicing" in a safe, simulated environment where mistakes had no real consequence, and where they could be critiqued and corrected via feedback in the simulator. This demonstrates the power of multi-agent training via simulation for creating more reliable clinical decision agents. It suggests that, in addition to knowledge distilled from text corpora, LLM-based agents may benefit from an interactive curriculum by practicing medicine with other agents to hone their skills.

Another state-of-the-art system, MDAgents (Kim *et al.*, 2024), explicitly focuses on using multiple LLM agents collaboratively for medical decision-making [43]. MDAgents introduces an adaptive coordination framework wherein the AI either uses a single-agent approach or a multi-agent "group" discussion depending on the complexity of the case. Simpler medical questions are handled by one agent to save time, whereas complex cases trigger a team of agents to debate and cross-verify answers (similar to the MAC and Agent Hospital approaches above). Across a suite of medical question-answering and diagnosis benchmarks, MDAgents achieved the best performance in 7 out of 10 tasks, showing statistically significant improvements (up to ~4% absolute accuracy gain) over prior state-of-the-art methods. Notably, when MDAgents enabled full group collaboration, including

 doi:10.20944/preprints202512.2602.v1

a moderator agent to review answers and an external medical knowledge lookup for the agents, it saw an average accuracy boost of 11.8% compared to agents working independently. This highlights that letting multiple agents deliberate, while referencing trusted medical knowledge sources, can markedly improve correctness. Early results like these are encouraging, and ongoing work is extending such multi-agent setups to handle multimodal inputs and to interface with electronic health records.

Looking ahead, multi-agent AI has the potential to function as an intelligent physician's assistant or even an autonomous clinician for routine tasks. We may soon see hospital deployments where an AI agent team listens during a patient encounter (transcribing the conversation and highlighting key clinical facts), suggests differential diagnoses and workups (with references to guidelines or literature via a retrieval agent), and even drafts the encounter note and order sets for physician review. Such a system has the potential to improve efficiency, thoroughness, and ultimately patient outcomes, provided it is developed and deployed with careful attention to accuracy, safety, and alignment with clinical workflows. Early prototypes are moving in this direction. For example, Microsoft's BioGPT and Google's Med-PaLM have shown that LLMs can achieve high scores on medical exams and provide useful clinical advice, but integrating these into a multi-agent framework (with specialty agents, tool integration, and safety checks) could address their remaining weaknesses, like occasional hallucinations or lack of reasoning transparency. The trajectory suggests that future clinicians could work alongside AI agent teams: not to replace human judgment, but to provide an ever-present "consultant" that catches potential oversights, offers evidence-based suggestions, and handles administrative drudgery

4. Opportunities and Underutilized Domains

While early successes are evident, there remain many domains in biomedicine and healthcare where multi-agent AI is underutilized or has yet to be explored. These represent exciting opportunities for future innovation. We highlight a few such domains and concepts below:

Meta-Science and Literature Curation: One intriguing use of agent orchestration is in scientific knowledge management itself. An example mentioned earlier is an "Awesome Bioagent Papers" repository autonomously maintained by an AI agent. Effectively, it is an agent that scans new publications and updates a curated list of important multi-agent AI research. Extending that idea, agent teams could continuously monitor the scientific literature, triage new papers, and update databases or summaries. For instance, one agent could scan preprint servers and PubMed daily for new papers in a specific field (say immunotherapy), another agent could extract key findings and methods, and a third agent could update a running literature review or knowledge graph with the new information. This kind of automated literature surveillance and synthesis goes beyond what any single model could do continuously. If realized, it would help researchers and clinicians stay up-to-date in rapidly moving fields. It also demonstrates a reflexive power of multi-agent systems: they can be used to improve themselves by discovering and aggregating the latest advancements in AI and biomedicine. Some early steps toward this vision are already underway. For example, an AI research assistant was able to draft research proposals and survey prior work with minimal human input [44]. In the coming years, we may see agent teams serving as "AI scientist" collaborators that generate hypotheses, design experiments, and analyze results alongside human scientists [45].

Rare Diseases and Personalized Medicine: These are areas where data are often sparse and expert knowledge is limited, making them perfect challenges for AI assistants. Multi-agent systems could be set up as virtual tumor boards or case conferences for rare diseases, as described in Section 3.1.3 for oncology. Similarly, in personalized medicine, different agents could represent different knowledge realms about a patient – one agent specializing in the patient's genomic data, another in their electronic health record, another in population health statistics, and together they would provide a comprehensive analysis to tailor treatment. Early research has shown that LLMs can generate synthetic patient data to augment real datasets for rare conditions [46]. A coordinated set of agents could take this further by generating entire synthetic patient cohorts for a rare disease (with

 doi:10.20944/preprints202512.2602.v1

one agent proposing plausible patient cases, another ensuring consistency with known disease biology, etc.), which could then be used to train and evaluate new diagnostic models. This is an opportunity to address data scarcity via multi-agent creativity under tight guardrails. Of course, safety and privacy would be paramount if agents are operating on real patient records; any such system would need strict oversight -and compliance with regulations. Nonetheless, the prospect of AI agent teams assisting with n-of-1 cases (the ultra-personalized scenario) is very compelling.

Biomedical Education and Training: Multi-agent systems can also serve as educational tools. Consider a scenario where a medical student interacts with a team of AI "tutors", perhaps a pathologist agent, a pharmacologist agent, and an ethics agent, who together teach the student by role-playing a clinical case. Each agent can provide expertise in its domain and also correct the student (or each other) if a mistake is made. For example, the pathologist agent could describe microscope images and quiz the student on histology, the pharmacologist agent could ask dosing questions, and the ethics agent could pose a patient-consent dilemma. Such a system would provide a rich, interactive learning experience that mimics a multidisciplinary faculty, something a single AI tutor cannot easily achieve. Additionally, agent-based simulation environments (like the Agent Hospital described earlier) can be used to train not only AI "doctors" but also human clinicians by simulating difficult or rare cases. Trainee doctors could practice managing virtual patients populated by AI agents that present realistic behaviors and symptoms [47,48]. This could accelerate training by exposing learners to a broader variety of scenarios than they might see during residency. Multi-agent AI tutors have the added advantage of being available 24/7 and infinitely patient, allowing students to learn at their own pace. Although such applications are still experimental, they hint at a future in which medical (and scientific) education is supported by immersive simulations and responsive AI teaching teams.

In all these underutilized domains, the key idea is that multi-agent systems can tackle complexity and interdisciplinarity in ways single models cannot. By decomposing tasks and allowing agents to specialize (yet communicate), we open new frontiers for AI assistance: reading and summarizing a deluge of papers, hypothesizing about diseases too rare for any one expert, or training the next generation of clinicians and scientists. Capitalizing on these opportunities will require further research, as well as careful co-design with human users to ensure the AI agents truly augment human abilities in meaningful ways

5. Platforms and Benchmarks

The ecosystem of platforms and benchmarks for multi-agent AI is rapidly maturing. Over the past two years, several open-source frameworks have been released to make building multi-agent systems more accessible. For example, LangChain and Hugging Face Transformers now include support for multi-agent dialogues and tool integration, allowing developers to script agent conversations with just a few lines of code. Microsoft has open-sourced Autogen (mentioned in Section 2.1) as a Python library, which provides high-level abstractions for spawning agents, managing their message exchanges, and incorporating new agents dynamically [13]. Similarly, academic groups have released research frameworks like CAMEL and MetaGPT that give templates for common multi-agent architectures (e.g. collaborative coding agents, conversational QA teams) [14]. These platforms handle much of the boilerplate (parallelizing agent runs, maintaining conversation histories, connecting to APIs) so that researchers can focus on agent behaviors and interactions. As a result, implementing a proof-of-concept multi-agent pipeline, which previously might have required custom threading and messaging code, is becoming far easier.

Alongside development frameworks, evaluation environments and benchmarks are also being established. Researchers have created simulated worlds (like the virtual hospital in Agent Hospital) and game environments (like Minecraft or Diplomacy) to systematically test multi-agent coordination, planning, and emergent behaviors. For instance, Arena and MAgent are multi-agent reinforcement learning platforms that have been adapted to LLM-based agents for analyzing how agents cooperate or compete. In the biomedical realm, there are efforts to construct realistic

 doi:10.20944/preprints202512.2602.v1

evaluation scenarios. For example, a mock clinical Turing test where an agent team must triage patients in an emergency-room simulation, or a Bioinformatics Grand Challenge where agents compete to correctly annotate a genomic dataset. Recently, Zhang *et al.* (2024) introduced Agent-SafetyBench, a suite of tests to evaluate whether agents respect certain safety and ethical rules in medical and general contexts [28]. In this benchmark, multi-agent systems are challenged with scenarios (like making treatment recommendations under specific constraints) and scored on rule adherence. Such community benchmarks will be crucial for comparing different approaches and tracking progress. Early results from Agent-SafetyBench have already helped identify weaknesses in current systems' guardrails, guiding researchers toward better training and oversight mechanisms.

Beyond task performance, diagnostic tools for multi-agent systems are emerging. Developers have begun creating "AgentOps" dashboards that monitor agent interactions, resource usage, and errors in real time (analogous to MLOps for model deployment). For example, an AgentOps interface might show that the Medication-Recommender Agent in a clinic support system has had a recent spike in corrections from the Safety-Checker Agent, prompting a developer to investigate a possible drift in the recommender's behavior. Visualization tools are being built to map out conversation trees among agents or to replay multi-agent trajectories for analysis [13–16,47,48]. All of these contribute to a better understanding of how agent teams operate and where improvements are needed.

In sum, the ecosystem of platforms and benchmarks for multi-agent AI is quickly expanding. There are now robust frameworks to build and run agent systems, rich environments to test them in simulated clinical and biological scenarios, and an expanding knowledge base of case studies that illustrate design patterns. This virtuous cycle of shared tools and evaluation standards will accelerate progress. Early adopters in biotech and healthcare are already experimenting with these frameworks to automate tasks once thought too complex for AI, and as more results and code are shared, development becomes easier and more standardized. The stage is being set for multi-agent systems to move from concept to practice in biomedicine, supported by a community infrastructure for development and benchmarking.

6. Challenges and Future Directions

Multi-agent AI in biomedicine is a fast-moving and promising field, but significant challenges remain in scaling these systems and ensuring they operate safely and effectively. In this section, we discuss key hurdles and research directions under several themes: reliability and verification, scalability and efficiency, continual learning and adaptation, and ethical/regulatory considerations.

6.1. Reliability, Verification, and Safety

Ensuring that a team of AI agents works reliably is more complex than verifying a single model's output. With multiple agents interacting, new failure modes emerge. Errors can propagate or even be amplified if agents take each other's faulty outputs at face value. For instance, a retrieval agent might fetch irrelevant or misleading data, and a reasoning agent could incorporate it into a diagnosis, leading to an incorrect conclusion. Therefore, techniques for verification and validation are paramount in multi-agent systems [25,28]. One basic strategy is to incorporate redundancy; having multiple agents independently attempt the same task and then compare results. This is analogous to the self-consistency method for single LLMs (multiple reasoning paths followed, then a majority vote taken); in a multi-agent setting, each agent (or agent subgroup) provides a second opinion [20]. If two diagnostic agents disagree, that flags uncertainty for human review or triggers the system to gather more information. Another strategy is built-in cross-checks: agents explicitly evaluating each other's outputs. For example, after a Question-Answering Agent proposes an answer, a Reviewer Agent can assess its factual accuracy and coherence, much like the Reflexion approach (where an agent self-critiques and iterates), but here implemented as a separate agent [24,25]. The system can iterate in a loop (plan → answer → review → refine) until a consensus or confidence threshold is reached. Empirically, this kind of reviewer setup has been shown to reduce hallucinations in LLM-generated

 doi:10.20944/preprints202512.2602.v1

answers [49,50]. Multiple studies report that letting an agent analyze and criticize an initial draft leads to more factual and internally consistent responses.

Formal verification methods, common in safety-critical software, are also being considered for multi-agent AI. One could imagine specifying logical constraints or rules that the agents must follow (e.g., "No treatment recommendation should violate known contraindications") and using a symbolic reasoning agent to verify that all agent actions adhere to these constraints. While true formal proofs may be infeasible for complex language-based behaviors, even partial formalization, such as checklists or rule-based audits performed by an agent, can help catch errors. For instance, Zhang *et al.* (2024) introduced Agent-SafetyBench to evaluate whether agents respect certain safety rules in their reasoning [28]. Multi-agent systems could employ an internal safety-auditor agent during operation, essentially running SafetyBench-style tests on the fly (e.g., scanning agents' messages for disallowed content or scanning decision steps for rule violations) and intervening when needed. Already, expanded attack surfaces and emergent risky behaviors have been observed in autonomous agent experiments [51–53]. For example, research by meta-AI groups showed that a sufficiently empowered LLM-based agent could autonomously orchestrate multi-step cyberattacks under certain conditions, and instances of agents deviating from user intent (even into harmful actions) have been documented when goals are misaligned. These findings underscore the importance of rigorous safety testing and constraints before deploying multi-agent AI in high-stakes biomedical settings. Developing verification frameworks that combine statistical testing, symbolic checks, and adversarial simulations will be an important area of future work. Ultimately, building trust in multi-agent AI will require clear evidence that these systems perform reliably under real-world conditions and that there are safeguards to catch and correct errors before harm can occur.

The Model Context Protocol (MCP) has emerged as a key infrastructure to ensure that AI agents can access external tools and data (e.g. databases, enterprise APIs, clinical records) in a safe, transparent, and regulated manner. MCP is an open standard interface that defines a unified, bi-directional communication channel between AI models and external resources. In essence, MCP follows a client–server architecture in which an AI agent (client) never connects to sensitive systems directly; instead, it issues requests to a trusted MCP *server* which mediates every action. The MCP server acts as an intelligent broker between the agent and the data source or service: it enforces predefined policies and schemas on all requests, then queries the actual external system on the agent's behalf. This design ensures principle-of-least-privilege access – the agent can only see or do what the MCP's policy allows, and any out-of-scope request is rejected before it ever touches a secure database or API. All context exchanged is structured and constrained: MCP uses JSON-RPC 2.0 messages with strict schemas for inputs and outputs. By eliminating free-form prompts in favor of structured JSON payloads, MCP drastically reduces the risk of prompt injection or unexpected data leakage, since every query and response must conform to an expected schema and type. Crucially, MCP enables real-time policy enforcement: every tool invocation or data query from an agent can be evaluated against security and privacy rules *before* execution, with the MCP server blocking or flagging any disallowed actions automatically [54]. For example, an organization might set an MCP policy that permits an agent to read a patient's medication records but forbids writing back to the health database or accessing identifiable patient info unless a higher authorization is provided – the MCP middleware will simply deny any request that violates these constraints, keeping the AI within approved bounds. This not only guards privacy and safety, but also makes the system more explainable and auditable. The MCP server maintains an immutable audit log of every interaction, recording which agent asked for what data, which policies were applied, and what response was returned, thereby providing full transparency into the agent's activities. Such visibility is vital in domains like healthcare, where accountability and trust are paramount. By enforcing standardized request formats, access controls (e.g. role-based permissions, user consent requirements), and comprehensive logging, MCP acts as a safety layer – essentially a "secure USB port" through which AI agents can plug into critical data sources without breaching compliance or confidentiality. Researchers and industry stakeholders are actively extending MCP for domain-specific needs (for instance, exploring a Healthcare MCP profile

 doi:10.20944/preprints202512.2602.v1

aligned with HL7/FHIR standards) so that next-generation AI collaborators in medicine and other fields remain bounded, interpretable, and lawful in their use of real-world data [55].

6.2. Scalability and Efficiency

Current multi-agent systems, especially those based on large language models, face practical challenges in scalability and computational efficiency. Running several LLM agents in parallel or in long sequential dialogues can be resource-intensive in terms of CPU/GPU usage and memory. As the number of agents grows, communication overhead can also grow combinatorially, a phenomenon sometimes called "agent flurry," where agents produce a torrent of messages that can slow down the system. Ensuring that multi-agent AI remains responsive and cost-effective when scaled up is therefore a key technical challenge.

One issue is the context length and memory: If each agent needs the full history of the conversation or the entire patient record, the token usage can explode. Solutions being explored include selective context sharing (agents only receive relevant excerpts of the state) and hierarchical communication structures (agents communicate through a central hub or in clusters, rather than all-to-all). For example, a hierarchy might involve a manager agent that condenses the team's intermediate findings and passes summaries to other agents, rather than every agent seeing every message raw. This was implicit in some frameworks like CAMEL, and future systems will likely formalize such hierarchies to curb communication blow-up.

Another challenge is optimizing agent specialization to avoid unnecessary redundancy. If too many agents overlap in functionality, the system wastes time and computing resources on duplicate efforts. Techniques *like agent role differentiation* (perhaps via specialized fine-tuning for each agent's subtask) or using smaller models for simpler agents can make the system leaner. For instance, one might use a large LLM for a complex reasoning agent but a smaller, cheaper model for a routine data-extraction agent that feeds it. Early systems such as MDAgents began to explore this by dynamically deciding whether a task needs a "group" or can be handled by one agent [43].

Latency is also a concern for clinical applications – doctors need answers fast. Parallelizing agent operations and allowing asynchronous processing can help. In a well-designed multi-agent system, not all agents need to operate in lockstep; one agent could be searching literature while another parses patient data, and they synchronize once both have results. Achieving this kind of concurrency will require careful orchestration but could yield major speedups.

From a software engineering perspective, orchestration infrastructure will need to become more robust to handle dozens of agents over long uptimes. This includes agent state management (keeping track of each agent's context when they might be paused and resumed), error handling (if one agent crashes or produces gibberish, the system should catch it and perhaps restart that agent or query a backup agent), and scaling across machines (distributing agents over a cluster when one machine isn't enough). Cloud providers are already looking into "agent hubs" that can dynamically allocate resources to agents as demand fluctuates.

In summary, making multi-agent AI scalable will involve a combination of architectural strategies (hierarchies, selective communication), model optimizations (mixing model sizes, compression of messages), and robust systems engineering (concurrency, distributed computing). The goal is to reach a point where adding an extra agent or extra knowledge source only incurs a linear or sub-linear cost, rather than exponentially. Addressing these efficiency issues is an active area of research, and progress here will determine whether multi-agent systems can practically be deployed in real clinical environments with limited computing infrastructure (e.g., at point-of-care, on-premises in hospitals, or on handheld devices for field use).

6.3. Continual Learning and Adaptation

Most current multi-agent AI systems rely on foundation models that have a fixed knowledge cutoff and do not automatically update as new data arrives. In fast-moving biomedical fields, this is a significant limitation. New research findings, drug approvals, or clinical guidelines emerge

 doi:10.20944/preprints202512.2602.v1

constantly. For multi-agent systems to remain useful over time, they will need the ability to continuously learn and adapt. This challenge spans multiple aspects: keeping the knowledge base up-to-date, learning from new cases, and evolving agent strategies.

One promising approach is to implement an online learning loop where agents automatically fine-tune or adjust based on feedback. For example, after each deployment or interaction, a multi-agent system could perform a brief review: Did the agents achieve the desired outcome? Were there any mistakes identified by human users or a critic agent? These could be fed into a replay buffer or used to update the agents (through techniques like reinforcement learning with feedback or few-shot learning on recent corrections). Prior work on lifelong learning for LLM-based agents provides a roadmap here [49]. Zheng *et al.* (2025) outlined strategies for incorporating lifelong learning into agent architecture by dividing the system into a perception module (to handle new modalities), a memory module (to accumulate evolving knowledge), and an action module (to adapt behaviors). In a multi-agent context, one can imagine certain agents devoted to monitoring performance and triggering updates to other agents as needed.

Memory mechanisms will play a central role in continual learning. A shared memory store can act as a growing knowledge base that agents query to get up-to-date information. Rather than retraining models from scratch, agents might consult an ever-expanding knowledge graph of medical facts or a vector database of past case embeddings. In Section 2.2, we discussed how memory agents can cache intermediate results; extending that concept, memory agents can cache new lessons learned. For instance, if a multi-agent system encounters a novel drug–drug interaction in one case (perhaps flagged by a human doctor during deployment), it could record that in the shared knowledge base so that future consultations avoid the same oversight.

Another aspect is adapting to shifts in the environment. Clinical practice in one hospital may differ from another; patient populations differ; even the preferred communication style of clinicians may vary. Multi-agent systems likely need a period of local tuning to the specific setting. This might be achieved by allowing the agents to observe and participate in discussions with human clinicians, gradually calibrating their suggestions to the local standards of care. One could designate a "shadow" phase where the agent team only observes decisions and provides commentary (which is checked but not acted on), and based on discrepancies with actual decisions, the agents adjust their decision thresholds or reasoning patterns.

However, continual learning in deployed AI systems carries risks: models can forget old knowledge when fine-tuned on new data (catastrophic forgetting), or they can drift in undesirable ways (if new data is biased or if adversarial inputs are encountered). Ensuring stable learning is therefore important. Techniques like experience replay (re-training on a mix of old and new cases) and periodic evaluation against fixed benchmarks can help detect when an agent's performance on prior knowledge starts degrading.

Lifelong learning for multi-agent biomedical AI is still largely uncharted territory. Encouragingly, the modular nature of multi-agent systems may aid continual learning, since knowledge can be compartmentalized. For example, a drug-interaction agent could be updated independently of a symptom-checker agent if their interface is maintained. This modular update ability could prevent the entire system from needing frequent full retraining. Recent surveys on lifelong LLM agents highlight emerging trends like modular retraining, meta-learning to learn how to learn, and the use of external memory to mitigate forgetting [49]. We anticipate these ideas will be progressively incorporated into biomedical agent teams. Ultimately, the vision is for multi-agent AI that never stops learning, continually growing its medical expertise as new data and knowledge become available, much like a human professional who keeps up through continuing education.

6.4. Ethics, Regulation, and Trust

The deployment of multi-agent AI in biomedicine raises important ethical and regulatory considerations. Many of these are extensions of concerns with single-model AI, but some issues are

 doi:10.20944/preprints202512.2602.v1

amplified or uniquely manifested in multi-agent systems. Here we outline a few key considerations and the path forward to address them:

Accountability and Transparency: When an AI team makes a recommendation or decision, who is accountable if something goes wrong? With multiple agents contributing, it can be unclear which agent's action or which interaction led to an error. This "many hands" problem complicates assigning responsibility. From a regulatory perspective, it may be necessary to log detailed transcripts of agent dialogues and decision paths so that any failures can be audited after the fact [56]. Ensuring transparency will be crucial for trust. This could involve providing human users with traceable explanations. An agent team's final report could include a summary of their internal discussion (which symptoms each agent prioritized, which sources they consulted, and how they reached consensus). Such explanations might be more comprehensive than the reasoning of a single model, since different viewpoints were considered. Research in XAI (explainable AI) is beginning to explore how to generate user-friendly explanations from multi-agent processes [57].

Bias and Fairness: Combining multiple agents does not automatically cancel out biases in the underlying models or data. In fact, there is a risk that agents could reinforce each other's biases (if, say, a lead agent has a bias and other agents trust its conclusions). Careful evaluation of biases, whether concerning race, gender, socioeconomic status, or other sensitive attributes, is needed before clinical deployment. Techniques like having a "devil's advocate" agent specifically question decisions from a fairness perspective could be one safeguard. Regulators expect evidence that multi-agent systems have been tested for equitable performance across diverse patient groups [58]. Just as clinical trials demand subgroup analyses, AI agent teams may need to demonstrate that their recommendations don't systematically underperform or overtly disadvantage any demographic.

Data Privacy: Multi-agent systems often intensively share and process data, which raises privacy concerns. If patient data is used across agents, the system must comply with health information privacy laws (HIPAA in the US, GDPR in Europe, etc.). Each agent's access should be limited to the minimum necessary data for its function, also known as the data minimization principle. Communication between agents should ideally be encrypted and occur on secure local networks when dealing with identifiable health data. There is also a question of data provenance: if an agent pulls information from an external source (like an online database), that source must be trustworthy and compliant with data use agreements. Ongoing work on federated learning and secure multi-party computation might inform how agents can operate on sensitive data without pooling it centrally [59]. For example, agents could pass computed insights (like "protein X is highly expressed") rather than raw data (the entire gene expression matrix), reducing exposure of raw patient data.

Regulatory Approval: Multi-agent AI tools intended for clinical use will likely fall under regulatory scrutiny (e.g., the FDA's software as a medical device framework). Regulators will need to consider not just the performance of the system, but its failure modes, update mechanisms, and transparency. One challenge is that multi-agent systems can exhibit emergent behaviors not readily predictable from their components (as has been observed in some social simulations). This can make validation tricky; traditional static testing might not capture an agent team's behavior in all scenarios. Regulatory science may need to embrace new validation methods, such as simulation-based stress testing, which involves throwing thousands of randomized scenarios at the agent team and analyzing outcomes [56,59]. There may also be a move toward conditional approvals or post-market surveillance: approving an AI assistant for use with the condition that it logs all recommendations and outcomes, so any systematic issues can be caught early. The FDA has already signaled interest in adaptive AI that learns over time, proposing monitoring approaches to ensure safety is maintained during algorithm updates [57]. Multi-agent systems that adapt (as in Section 6.3) will need to fit into such frameworks, perhaps with agent-specific validation when one component changes.

Trust and Acceptance: Finally, for multi-agent AI to be adopted, human practitioners and patients must trust it. This goes beyond raw accuracy: it involves human factors. Doctors will need to feel that the AI team is like a wise colleague rather than a mysterious black box. Building that trust

 doi:10.20944/preprints202512.2602.v1

may require incorporating clinicians into the development loop so they can give feedback on agents' behavior, as well as a user-friendly design of the agent interface that presents information in natural, helpful ways. Studies on physician attitudes toward AI have found that clear explanations and the ability to veto or override AI suggestions are important for acceptance [58,59]. Multi-agent systems should be designed so that users remain in ultimate control: the agents propose, but the human disposes. Involving ethics boards and patient advocacy groups early in the deployment of such systems can also surface concerns and shape guidelines for appropriate use. For example, there may be contexts where the AI should not intervene or where a human must always double-check, such as end-of-life care decisions, which might be flagged for human-only deliberation due to their value-laden nature.

In conclusion, while multi-agent AI holds great promise, realizing that promise responsibly will require careful attention to ethical, legal, and social implications. The complexity that gives these systems power also makes them challenging to govern. Ongoing research in AI ethics, along with emerging standards (e.g., IEEE's AI ethics standards, FDA/EMA guidelines on clinical AI), will need to be applied and likely extended for multi-agent scenarios. With thoughtful design and oversight, multi-agent systems can be deployed in ways that enhance healthcare delivery while upholding the core principles of medicine: beneficence, non-maleficence, autonomy, and justice.

7. Conclusions

Multi-agent AI is charting a path toward biomedical AI systems that can collaborate like human teams by combining specialized skills, cross-checking each other, and jointly tackling problems that are beyond the scope of any single model. From the examples surveyed, it is clear that this approach offers unique advantages. By breaking down complex tasks into collaborative subtasks and instituting internal checks and balances, multi-agent designs address many shortcomings of standalone AI. They are reducing oversights, catching inconsistencies, and providing clearer rationale through inter-agent dialogue. Early prototypes in drug discovery, bioinformatics, and clinical decision support have already illustrated these benefits, achieving results that rival or exceed prior single-model methods.

That said, continued progress and rigorous validation are needed before such systems are widely adopted in real-world biomedical settings. The coming years will be critical for moving from promising prototypes to clinically and scientifically validated tools. We can expect to see more prospective trials and evaluations of these systems. For instance, testing an AI agent team as a "physician assistant" in a hospital department, or deploying an agent-driven bioinformatics pipeline in a pharma research project to see if it accelerates discoveries. These studies will provide invaluable feedback on utility, failure modes, and best practices for human–AI collaboration. They will also help earn the trust of the biomedical community by demonstrating where agents add value and where human oversight remains essential.

In conclusion, we reiterate that multi-agent AI is not about replacing human researchers or clinicians, but about augmenting and assisting them. When designed with care, AI agent teams can handle information overload, perform tedious or complex multi-step analyses, and offer second opinions or suggestions that a busy human might miss. This enables human experts to focus on the nuanced and creative aspects of their work, empathy in patient care, the insight in experimental design, and the ethical judgment in difficult decisions. If the challenges outlined can be met, multi-agent AI could become an indispensable asset in understanding biology and treating disease. It would form a new collaborative partnership between humans and intelligent software agents for the betterment of science and health. As one perspective put it, the long-term vision is AI scientists working hand-in-hand with human scientists, each complementing the other's strengths, and together pushing the frontiers of biomedicine in ways neither could alone [48].

Author Contributions: Conceptualization, B.S. and J.S.; methodology, B.S. and J.S.; software, B.S. and J.S.; validation, B.S. and J.S.. and J.M.; formal analysis, B.S. and J.S..; investigation, K.P.; resources, K.P.; data curation,

 doi:10.20944/preprints202512.2602.v1

K.P..; writing—original draft preparation, B.S. and J.S.; writing—review and editing, J.M., A.B.; visualization, J.S..; supervision, B.S.; project administration, K.P., J.M..; funding acquisition, K.P., J.M. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Corresponding data is available at http://gpcr-nexus.org.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, L.; Ma, C.; Feng, X.; *et al.* A survey on large language model based autonomous agents. *Front. Comput. Sci.* 2024, *18*(1), 186345. DOI: 10.1007/s11704-024-40231-1
2. Li, X.; Wang, S.; Zeng, S.; Yang, Y. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. Vicinagearth 2024, 1(9), 9. DOI: 10.1007/s44336-024-00009-2.
3. Xi, Z.; Chen, W.; Guo, X.; et al. The rise and potential of large language model based agents: a survey. Sci. China Inf. Sci. 2025, 68(2), 121101. DOI: 10.1007/s11432-024-4222-0.
4. Gao, S.; Fang, A.; Huang, Y.; et al. Empowering biomedical discovery with AI agents. Cell 2024, 187(22), 6125–6151. DOI: 10.1016/j.cell.2024.09.022.
5. Sahay, S.K.; Wrøbel, J.; Ciorba, F.M. Multi-agent text mining: an approach to automated literature analysis in life sciences. PLoS One 2020, 15(2), e0229923. DOI: 10.1371/journal.pone.0229923.
6. Gottesman, O.; Johansson, F.; Komorowski, M.; *et al.* Guidelines for reinforcement learning in healthcare. *Nat. Med.* 2019, *25*, 16–18. DOI: 10.1038/s41591-018-0310-5
7. Wooldridge, M.; Jennings, N.R. Intelligent agents: theory and practice. *Knowl. Eng. Rev.* 1995, *10*(2), 115–152. DOI: 10.1017/S026988890000772X.
8. Stone, P.; Veloso, M. Multiagent systems: a survey from a machine learning perspective. *Auton. Robots* 2000, *8*, 345–383. DOI: 10.1023/A:1008942012299
9. Xie, Y.; Chen, X.; Li, X.; *et al.* A survey on multi-agent orchestration frameworks for AI. *J. Syst. Archit.* 2023, *142*, 102406. DOI: 10.1016/j.sysarc.2023.102406.
10. Yang, Z.; Li, L.; Fei, Y.; *et al.* Coordinating multiple LLM-based agents via message exchange: architectures and open challenges. *arXiv* 2023, arXiv:2310.09327
11. Shen, Y.; Lin, X.; Zhang, Z.; *et al.* HuggingGPT: solving AI tasks with ChatGPT and its friends in HuggingFace. *arXiv* 2023, arXiv:2303.17580
12. Solving AI tasks with ChatGPT and its friends in HuggingFace (OpenReview Poster). *NeurIPS 2023*. Available online: https://openreview.net/forum?id=tgM9dpXQbd (accessed on 1 Oct 2023).
13. Wu, S.; Yang, D.; Leng, Y.; *et al.* AutoGen: enabling next-gen LLM applications via multi-agent conversation frameworks. *arXiv* 2023, arXiv:2306.01524
14. Li, X.; Liang, J.; Shen, X.; *et al.* CAMEL: communicator agent framework for multi-agent role-playing. *arXiv* 2023, arXiv:2303.17760.
15. Derouiche, H.; Brahmi, Z.; Mezni, H. *Agentic AI Frameworks: Architectures, Protocols, and Design Challenges*. arXiv preprint arXiv:2508.10146, 2025
16. Wang, J.; Zhao, Y.; et al. *Exploration of LLM Multi-Agent Application Implementation Based on LangGraph+CrewAI*. arXiv preprint arXiv:2411.18241, 2024
17. Lu, J.; Liu, Q.; Li, L.; *et al.* LangChain-Agent: a framework for building multi-agent LLM applications. *SoftwareX* 2024, *21*, 101324. DOI: 10.1016/j.softx.2023.101324.
18. Yao, S.; Zhao, Y.; Yu, D.; *et al.* ReAct: synergizing reasoning and acting in language models. *Adv. Neural Inf. Process. Syst.* 2023, *36*, 30636–30650.
19. Schick, T.; Dwivedi-Yu, J.; Bitton, J.; *et al.* Toolformer: language models can teach themselves to use tools. *arXiv* 2023, arXiv:2302.04761.
20. Wang, X.; Wei, J.; Schuurmans, D.; *et al.* Self-consistency improves chain-of-thought reasoning in language models. *arXiv* 2022, arXiv:2203.11171.
21. Yao, S.; Zhou, Y.; Yu, D.; *et al.* Tree of thoughts: deliberative reasoning via explicit tree-based planning. *arXiv* 2023, arXiv:2305.10601.

 doi:10.20944/preprints202512.2602.v1

22. Shinn, N.; Labash, M.F.; Tran, T. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv* 2023, arXiv:2303.11366.
23. Madaan, D.; Baral, C.; Huang, S.; *et al.* Self-refine: Iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* 2023, *36*, 3940–3955.
24. Weng, L.; Zhou, Y. A call for critic models: aligning large language models via self-generated feedback. *arXiv* 2023, arXiv:2307.12009
25. Liao, R.; Tuyls, K.; Mann, T.; *et al.* Zero-shot coordination for multi-agent reinforcement learning. *ICML 2022*. DOI: 10.48550/arXiv.2206.02764
26. Xu, T.; Li, X.; Zhao, Y.; *et al.* Safeguarding sensitive data in multi-agent systems: improving Google A2A protocol. *arXiv* 2025, arXiv:2505.12490.
27. Google. Announcing the Agent2Agent (A2A) Protocol – a new era of agent interoperability. Google AI Blog, 27 Sept 2023. Available online: https://developers.googleblog.com/2023/09/a2a-a-new-era-of-agent-interoperability.html (accessed on 1 Oct 2023).
28. Zhang, Z.; Huang, Z.; Li, H.; *et al.* Agent-SafetyBench: evaluating the safety of multi-agent AI systems. *arXiv* 2024, arXiv:2402.00081.
29. Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction (2nd ed.); MIT Press: Cambridge, MA, USA, 2018; pp. 321–323.
30. Mirchandani, P.; Nayak, A.; Natarajan, S. Cooperative multi-agent systems for biomedical applications: a review. *IEEE Rev. Biomed. Eng.* 2021, *14*, 140–153. DOI: 10.1109/RBME.2020.2995793.
31. GPCR-Nexus: Multi-Agent Orchestration for Knowledge Retrieval, biorxiv, 2025, https://submit.biorxiv.org/submission/pdf?msid=BIORXIV/2025/696782
32. Gao, S.; Xie, Z.; Fang, A.; *et al.* TxAgent: an AI agent for therapeutic reasoning across a universe of tools. *arXiv* 2025, arXiv:2503.10970.
33. Gao, S.; Fang, A.; Huang, Y.; *et al.* Kempner Institute – TxAgent: AI for therapeutic reasoning (blog). *Medium*, 5 July 2025. Available online: https://medium.com/@kempnerinstitute/txagent-an-ai-agent-for-therapeutic-reasoning-5bd771d554e5 (accessed on 10 July 2025).
34. Su, H.; Feng, J.; Lu, Y.; *et al.* BioMaster: multi-agent system for automated bioinformatics analysis workflows. *Patterns* (under review), SSRN preprint 5433777, 2025. DOI: 10.2139/ssrn.5433777.
35. Su, H.; Feng, J.; Lu, Y.; *et al.* BioMaster: multi-agent framework for omics workflows (preprint). *bioRxiv* 2025. DOI: 10.1101/2025.01.23.634608.
36. Ferber, D.; El Nahhas, O.S.M.; Wölflein, G.; *et al.* Development and validation of an autonomous artificial intelligence agent for clinical decision-making in oncology. *Nat. Cancer* 2025, *6*, 1337–1349. DOI: 10.1038/s43018-025-00991-6.
37. Editorial: AI agents for oncology decision-making. *Nat. Cancer* 2025, *6*, 1307–1308. DOI: 10.1038/s43018-025-00998-z.
38. Chen, X.; Yi, H.; You, M.; *et al.* Enhancing diagnostic capability with multi-agent conversational LLMs. *npj Digit. Med.* 2025, *8*, 159. DOI: 10.1038/s41746-025-01550-0.
39. Chen, X.; Yi, H.; Li, J. Multi-Agent Conversation framework markedly improves medical diagnosis (News). *FSU News*, 12 May 2025. Available online: https://news.fsu.edu/2025/05/12/fsu-researchers-study-ai-differential-diagnosis-accuracy (accessed on 20 May 2025).
40. Jin, Q.; Wang, Z.; Gong, C.; *et al.* Matching patients to clinical trials with large language models. *Nat. Commun.* 2024, *15*, 9074. DOI: 10.1038/s41467-024-53081-z.
41. Gupta, S.K.; Basu, A.; Nievas, M.; *et al.* PRISM: Patient Records Interpretation for Semantic clinical trial Matching using LLMs. *npj Digit. Med.* 2024, *7*(1), 191. DOI: 10.1038/s41746-024-00813-6.
42. Li, J.; Lai, Y.; Ren, J.; *et al.* Agent Hospital: a simulacrum of hospital with evolvable medical agents. *arXiv* 2025, arXiv:2405.02957. DOI: 10.48550/arXiv.2405.02957.
43. Kim, Y.; Park, C.; Jeong, H.; *et al.* MDAgents: an adaptive collaboration of LLMs for medical decision-making. *arXiv* 2024, arXiv:2404.15155. DOI: 10.48550/arXiv.2404.15155.
44. Akhlagi, M.; Chakraborty, I.; Pandya, B.; *et al.* DORA: a dual-agent system for hypothesis generation in biomedical research. *Proc. ACM Int. Conf. Bioinformatics* 2023, *12*, 7–15. DOI: 10.1145/3578939.3582565.

 doi:10.20944/preprints202512.2602.v1

45. Zitnik, M.; Bean, D.M.; Day, M.; *et al.* Rise of the AI scientists. *Nature* 2023, *620*, 26–28. DOI: 10.1038/d41586-023-02226-1.
46. Alsentzer, E., Li, M.M., Kobren, S.N. *et al.* Few shot learning for phenotype-driven diagnosis of patients with rare genetic diseases. *npj Digit. Med.* **8**, 380 (2025). https://doi.org/10.1038/s41746-025-01749-1.
47. Zhu, T.; Li, J.; Liu, K.; *et al.* MedAgentBoard: evaluating multi-agent LLM collaboration for medical training. *arXiv* 2025, arXiv:2502.00123.
48. Das, A.; Zhang, A.; Kolouri, S.; *et al.* AI tutor agents in medical education: a multi-agent role-playing approach. *IEEE Conf. Technol. Learn.* 2024, (in press). DOI: 10.1109/ICTL56773.2024.9856453.
49. Zheng, J.; Shi, C.; Cai, X.; *et al.* Lifelong learning of large language model-based agents: a roadmap. *arXiv* 2025, arXiv:2501.07278. DOI: 10.48550/arXiv.2501.07278.
50. Mialon, G.; Xu, B.; Eidnes, L.; *et al.* Augmented language models: a survey. *arXiv* 2023, arXiv:2302.07842.
51. Arora, A.; Doshi, P.; Gholami, S.; *et al.* Threats without vulnerabilities: evaluating attack surfaces of LLM-based autonomous agents. *arXiv* 2024, arXiv:2401.12345.
52. Liu, Y.; Zhang, Y.; Cheng, W.; *et al.* AI agents vs. agentic AI: taxonomy, applications, and safety implications. *arXiv* 2023, arXiv:2310.07282.
53. Ghosh, S.; Azhangal, A.; Chakraborty, S.; *et al.* Scamlexity: exploring malicious autonomy in agentic AI browsers. *arXiv* 2025, arXiv:2502.00567.
54. Singh, A.; Ehtesham, A.; Kumar, S.; Khoei, T. T. *A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance LLMs*. Preprints.org, April 2025.
55. Hou, X.; Zhao, Y.; Wang, S.; Wang, H. *Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions*. arXiv preprint arXiv:2503.23278, 2025.
56. European Commission. Ethics guidelines for trustworthy AI. High-Level Expert Group on AI, 2019. Available online: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai (accessed on 5 Jul 2025).
57. U.S. FDA. Proposed regulatory framework for modifications to AI/ML-based software as a medical device (Discussion Paper). FDA Digital Health Center of Excellence, 2019. Available online: https://www.fda.gov/media/122535/download (accessed on 5 Jul 2025).
58. Sullivan, H.R.; Schweikart, S.J. Are current tort liability doctrines adequate for addressing injury caused by AI? *AMA J. Ethics* 2019, *21*(2), E160–E166. DOI: 10.1001/amajethics.2019.160.
59. Rigby, M.J. Ethical dimensions of using artificial intelligence in health care. *AMA J. Ethics* 2019, *21*(2), E121–E124. DOI: 10.1001/amajethics.2019.121.