CS 839: AI Agents for Science
Presentation Information

Instructions: The goal of the class presentation is to read a high-quality paper about AI agents and teach it to the class. The paper does not need to be about scientific agents, but your presentation should make clear how its ideas connect to the course. The suggested papers below cover a range of topics and are not an exhaustive list. Please contact me if you want to present a paper that is not listed.

Goals. We have several goals for these presentations.

- Familiarize yourself and the class with major research results on AI agents. We should understand the problem setting, the core mechanism, the evidence supporting the paper's claims, and the important limitations.
- Develop an understanding of how the paper fits into the broader research landscape. What earlier ideas does it build on? What assumptions distinguish it from related work?
- Build skill with identifying future directions: extensions of the work, open cases, failure modes, and applications to new areas.
- Practice giving a clear research talk and leading a technical discussion.

Deliverables. There are two items that must be turned in.

1. Presentation proposal: submit a short document (roughly half a page is fine) naming the primary paper, explaining why your team chose it, and describing how it fits the course. You may also name one or two supporting papers. The proposal is due October 27; I will announce the sign-up process separately.
2. Post-presentation materials: submit your slides or presentation notes. Add brief answers to questions raised in class if those answers were not already included in the presentation. The due date is TBD.

Teams and timing. Use your existing course team unless we make a different arrangement. The exact presentation length will depend on the final number of teams; I will announce the schedule and time limit after paper selection.

1 Frequently Asked Questions

- What format is appropriate? You may use slides or another format that helps you explain the work. If you use slides, do not simply copy pages from the paper. Useful talks provide a roadmap, motivate the problem, develop intuition before technical details, use concrete examples, and pause periodically for questions.
- What should the presentation cover? At minimum, explain the problem, what the authors mean by an "agent," the agent's observations and actions, the method, the main experimental evidence, and the most important failure modes. Discuss whether a simpler baseline could explain the reported gains.
- What are we being graded on? We will use the goals above and the level of effort and sophistication in the presentation. You should be able to explain the central ideas clearly, interpret the experiments rather than merely list results, and discuss limitations when asked.
- Can we present multiple papers? This can work when the papers support a focused comparison. Choose one primary paper and keep the supporting set small enough that you can still explain the primary paper in depth. Do not turn the presentation into a broad survey.
- Can we choose a paper already discussed in lecture? Possibly, but the presentation must add substantial depth or place it in a careful comparison with newer work. Check with me before selecting a paper that we covered closely in class.

- Do I have to attend the other presentations? Yes. Attendance and participation are part of the presentation grade. Ask questions and treat the presentations as discussions rather than additional lectures.
- Can the presentation connect to our project? Yes; this is often useful, but it is not required. You may present a paper that gives your team a technical starting point, evaluation method, or application domain for the project.
- What may we assume the class already knows? You may assume material from our lectures and standard machine-learning background. Include any additional background needed to understand the paper's main ideas and evidence.

2 Suggested Papers

The list below is organized by theme. Several papers fit more than one category. Links point to the paper or an official publication page.

2.1 Reasoning, planning, and memory

1. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models." ICLR, 2023. A foundational agent loop that interleaves reasoning traces with actions and observations.
2. Erdogan et al., "Plan-and-Act: Improving Planning of Agents for Long-Horizon Tasks." ICML, 2025. Separates a high-level planner from an environment-specific executor and trains both with synthetically generated planning data.
3. Zhang et al., "ReCAP: Recursive Context-Aware Reasoning and Planning for Large Language Model Agents." NeurIPS, 2025. Uses recursive decomposition, shared context, and bounded prompts to keep long-horizon plans aligned with their parent goals.
4. Xu et al., "A-Mem: Agentic Memory for LLM Agents." NeurIPS, 2025. Builds a dynamically linked memory network whose organization and representations evolve as new experiences arrive.
5. Gupta et al., "Memo: Training Memory-Efficient Embodied Agents with Reinforcement Learning." NeurIPS, 2025. Trains an embodied agent to compress long visual histories into learned memories for efficient long-horizon behavior.
6. Wang et al., "RAGEN: Understanding Self-Evolution in LLM Agents via Multi-Turn Reinforcement Learning." arXiv, 2025. Studies multi-turn reinforcement learning for agents and analyzes the stability problems that appear when behavior evolves through interaction.

2.2 Tool use, environments, and coding agents

7. Yang et al., "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering." NeurIPS, 2024. Shows how the design of an agent-computer interface changes performance on repository-level software tasks.
8. Xu et al., "TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks." NeurIPS, 2025. Places agents in a simulated software company where they must browse, code, communicate, and complete consequential workplace tasks.
9. Yang et al., "SWE-smith: Scaling Data for Software Engineering Agents." NeurIPS, 2025. Automatically synthesizes repository-level training tasks and uses them to train an open software-engineering agent.
10. Yuan et al., "OSWorld 2.0: Benchmarking Computer Use Agents on Long-Horizon Real-World Tasks." arXiv, 2026. Extends computer-use evaluation toward longer, stateful workflows with richer task setup and process-level diagnosis.
11. Lu et al., "ProjDevBench: Benchmarking AI Coding Agents on End-to-End Project Development." arXiv, 2026. Moves beyond issue resolution to evaluate whether coding agents can design and build complete software projects from specifications.
12. Lee et al., "SEC-bench: Automated Benchmarking of LLM Agents on Real-World Software Security Tasks." NeurIPS, 2025. Builds reproducible vulnerability tasks for evaluating proof-of-concept generation and security patching by coding agents.

13. Singh et al., "IssueTrojanBench: Benchmarking AI Coding Agents Against Malicious Issue Requests." arXiv, 2026. Tests whether coding agents recognize malicious repository requests or obediently implement harmful changes.

2.3 Multi-agent systems and agent training

14. Cemri et al., "Why Do Multi-Agent LLM Systems Fail?." NeurIPS, 2025. Develops a taxonomy and annotated trace dataset for failures in system design, inter-agent alignment, and verification.

15. Xi et al., "AgentGym-RL: Training LLM Agents for Long-Horizon Decision Making through Multi-Turn Reinforcement Learning." arXiv, 2025. Develops a unified reinforcement-learning framework for interactive, long-horizon agent tasks.

16. Luo et al., "Agent Lightning: Train ANY AI Agents with Reinforcement Learning." arXiv, 2025. Separates agent execution from reinforcement-learning infrastructure and tackles credit assignment across complex traces.

17. Zhu et al., "MedAgentBoard: Benchmarking Multi-Agent Collaboration with Conventional Methods for Diverse Medical Tasks." NeurIPS, 2025. Asks when multi-agent medical systems actually outperform single models and established specialized methods.

18. Kang et al., "VIKI-R: Coordinating Embodied Multi-Agent Cooperation via Reinforcement Learning." NeurIPS, 2025. Combines a visual multi-agent benchmark with reinforcement learning for heterogeneous embodied cooperation.

2.4 Evaluation, reliability, and security

19. Yao et al., "τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains." ICLR, 2025. Evaluates whether agents can interact with users, call domain APIs, and obey policy constraints consistently.

20. Debenedetti et al., "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents." NeurIPS, 2024. Jointly evaluates useful task completion and resistance to prompt injections embedded in tool outputs.

21. Lu et al., "ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities." Findings of NAACL, 2025. Tests stateful tools, implicit dependencies, missing information, and multi-turn user interaction.

22. Luo et al., "AgentAuditor: Human-Level Safety and Security Evaluation for LLM Agents." NeurIPS, 2025. Uses a memory-augmented evaluator and a new benchmark to assess risks that develop across multi-step agent traces.

23. Chen et al., "MLR-Bench: Evaluating AI Agents on Open-Ended Machine Learning Research." NeurIPS, 2025. Evaluates ideas, experiments, and papers produced by research agents—including whether their reported results are actually valid.

2.5 Agents for science

24. Schmidgall et al., "Agent Laboratory: Using LLM Agents as Research Assistants." Findings of EMNLP, 2025. Organizes literature review, experimentation, and report writing into an agent workflow with optional human feedback.

25. Toledo et al., "AI Research Agents for Machine Learning: Search, Exploration, and Generalization in MLE-bench." NeurIPS, 2025. Treats automated machine-learning research as search over candidate solutions and compares greedy, evolutionary, and tree-search strategies.

26. Yamada et al., "Towards End-to-End Automation of AI Research." Nature, 2026. Presents a substantially expanded AI Scientist pipeline and evaluates both scaffolded and template-free research automation.

27. Gottweis et al., "Accelerating Scientific Discovery with Co-Scientist." Nature, 2026. Uses specialized agents, debate, and tournament-style evolution to generate and refine biomedical hypotheses.

28. Ghareeb et al., "A Multi-Agent System for Automating Scientific Discovery." Nature, 2026. Introduces Robin, which closes a loop between literature-grounded hypotheses, human-run experiments, and agentic data analysis.

29. Gao et al., "AutoScientists: Self-Organizing Agent Teams for Long-Running Scientific Experimentation." arXiv, 2026. Studies persistent teams that reorganize their roles and coordination while running extended scientific investigations.

30. Ríos-García et al., "AI Scientists Produce Results without Reasoning Scientifically." arXiv, 2026. Provides a critical evaluation of whether successful scientific-agent outputs reflect evidence use, refutation, and belief revision.

A useful selection test. Before proposing a paper, ask whether it supports a focused presentation: Is there a concrete agent design to explain? Are the environment and evaluation clear? Does the paper contain enough evidence to assess what worked and why? Can your team identify a limitation or open question that would support class discussion? If the answer to most of these questions is yes, the paper is probably a good candidate.