B9153 Fall 2025 — Project instructions

Dates

- Oct 28 — Project Proposal Due Date
- Dec 1 to Dec 5 — 15-min appointment-based project presentations
- Dec 7 — Final project write-up Due

Logistics

- Team size: Students can work in the sizes of 1 to 4.
- Project Proposal: 1–2 pages describing the question, foreseeable challenges, and initial attempts at solving.
- Writeup Length: 8 page limit for the main body, excluding references and appendix (if any), following ICML template:
  - `https://icml.cc/Conferences/2025/AuthorInstructions`
- Presentation: Prepare a 15 minute presentation (12 minute + 3 minute QA) during the week of Dec 1 to Dec 5 (signup the presentation slot).

Projects can be based on computation, empirics, theory, or modeling. Examples of projects can include:

Sample Project Ideas

1. **AI as Researcher:** Use an LLM to write a full research paper in your field (e.g., OR/MS, finance, marketing, etc.).
   a. See example: `https://astropilot-ai.github.io/DenarioPaperPage/`
2. **Benchmark/Environment for Your Field:** Identify an important problem in your discipline and build an environment or benchmark to evaluate and compare agents or traditional algorithms.
3. **EnvironmentArena:** Many research papers introduce new environments and report agents' performance, but such environments are often difficult to share, reproduce, or modify. The goal here is to develop an "environment-as-a-first-class-citizen" infrastructure that allows users to easily share, fork, and evaluate both LLM and non-LLM agents (e.g., bandit or queueing simulations). Ultimately, this project aims to encourage that every paper should not only post on Arxiv but also post an "Evaluation Page" on Environment Arena, facilitating benchmarks and reproducibility.

4. **Scaling Grounded Environments and Vibe Simulation:** Building grounded simulators (e.g., warehouse environments) is time-consuming. How can we scale up environment creation while maintaining realism? Can we repurpose existing simulators through modular "vibe coding"? Explore scalable methods for reproducing environments from academic papers using such compositional simulation strategies.

5. **LLM Simulation for Digital Twins:** LLM-based digital twins are gaining traction in both academia and industry. Open questions include:

   - Can we build a generalizable foundational model for human digital twins?
   - Can LLM-based user simulations replace or augment A/B testing?
   - How can we calibrate LLM behavior using real user feedback?
   - Can we collect user feedback actively so that the LLM simulation improves continuously?

6. **Optimizing Toward LLM-as-Judge:** Although LLMs are imperfect proxies for humans, they are increasingly used to make automated judgments. However, querying them is costly. Example: in a matching system with n jobs and m workers, even if the LLM can accurately score each pair (i, j), querying all entries is infeasible. Key questions include:

   - How to optimize for LLM preferences with minimal queries?
   - How to leverage a less accurate but more efficient approximator (e.g., bi-encoder embeddings)?
   - How to combine LLMs and approximators to minimize query cost?
   - How to integrate limited human-labeled data into this framework?

7. **Context Engineering:** As discussed in class, fine-tuning is costly and often lacks generalizability. Prompt/context engineering thus offers a promising alternative. Open questions include:

   - How can we build an automated, generalizable pipeline for context engineering?
   - How to optimize context dynamically when both tasks and environments evolve (e.g., in content moderation)?

8. **Multi-Agent Orchestration:** Multi-agent systems are increasingly being deployed in real-world settings. Construct a multi-agent system and optimize its performance. See AG2 (`https://ag2.ai/`), MassGen (`https://github.com/Leezekun/MassGen`), and CMBAgent (`https://github.com/CMBAgents`) for inspiration.

9. **Efficient Agentic Serving:** Agentic systems are slow and costly. How can we design better scheduling and resource allocation to reduce cost and improve throughput? This is a domain where operations research can significantly contribute to LLM systems.

10. **Beyond Transformers:** Explore new architectures such as *Mamba* ("Mamba: Linear-Time Sequence Modeling with Selective State Spaces"), which summarizes historical information via states s_t similar to MDPs.

    a. Conduct a literature review on Mamba and provide a case study.

11. **Should We Be Polite to ChatGPT?:** Users interact with ChatGPT in different tones—some polite ("please," "thank you"), others direct ("do this"). Does tone affect model output? This raises broader questions about how we should treat AI and the societal implications of anthropomorphizing them.

12. **Adding Factual Knowledge to LLMs:** If we aim to fine-tune an LLM with new knowledge (e.g., a new textbook or recent events), what's the best method?

    a. Literature review and/or case study.

    b. Related work: `https://arxiv.org/pdf/2405.05904`

13. **Text Classification with Uncertainty Quantification:** While LLMs achieve strong classification accuracy, real-world applications require probability distributions over classes. How can we best obtain and calibrate these probabilities? Compare direct LLM-generated probabilities vs. those derived from token log-probabilities, and explore how chain-of-thought reasoning affects reliability.

14. **Input-Based Prompt Engineering:** Design models that automatically optimize prompts based on input/task characteristics. Explore this direction through case studies.

15. **How Does Suno Work?:** Suno is an AI music generation platform producing surprisingly high-quality outputs. Investigate the technologies behind it and how AI models are used in music composition.

16. **LLM Gaia:** Inspired by the goddess Gaia, this project envisions an LLM that evolves by continuously integrating human knowledge. Users can submit documents; moderators approve and deduplicate data; and the LLM is fine-tuned iteratively. Explore how continuous fine-tuning affects model performance, and prototype a small-scale demo.

17. **LLMs for Agent Simulation:** Build LLM-driven agents to simulate social, economic, or behavioral systems in your field.

    a. See *Generative Agents: Interactive Simulacra of Human Behavior*.

    b. See *Automated Social Science: Language Models as Scientist and Subjects*.

18. **AI-Generated Review Summaries:** Amazon now displays AI-generated review summaries (`https://www.aboutamazon.com/news/amazon-ai/amazon-improves-customer-reviews-with-generati`
While efficient, these summaries can introduce bias or inaccuracy, affecting sales (`https://sellercentral-europe.amazon.com/seller-forums/discussions/t/8ab5cfbb-02ab-4d9d-a7bb-7d7c6`
Conduct a literature review and design methodology for effective, unbiased summarization.

19. **LLMs for Experimentation:** Inspired by *Automated Social Science*, explore how LLMs can simulate experiments.

    i. Extend the analysis from predicting mean responses to predicting full response distributions; identify and correct discrepancies.

    ii. Evaluate whether LLM-based simulated experiments can help debias treatment-effect estimation under selection bias or SUTVA violations.

References