Final Project

Teams

- Teams of 1, no.
- **Teams of 2-6, yes!**
- Teams >6, no.

Topics

- Should be related to something we've talked about in class:
  - Sequential decision making
  - Learning from evaluative feedback
  - Learning from human feedback
  - Bandits, MDPs, Planning, RL, IL, BC, LfD, IRL, RLHF, TAMER, reward design, LLMs, VLMs, etc.

LinUCB

- Basically anything listed here would be fine: https://rl-conference.cc/callforpapers.html

Examples of bad projects

- Training an image classifier to recognize street signs.
- Solving a regression problem to predict house prices in SLC.
- Solving a generic optimization/supervised learning problem that doesn’t involve sequential decision making or evaluative feedback.

Examples of good projects from last year

- RL for Container Warming Optimization in Serverless Computing
- Comparative Analysis of Multi-Agent RL Algorithms
- Applying RL to a favorite board or card game (e.g., Gin Rummy, Settlers of Catan, BS/Cheat)
- Contextual bandits to help find effective visualizations of scientific data.
- RL for Cryptocurrency Trading
- RL-Based Electric Vehicle Charging Control
- Combining IRL and RL
- Model Distillation for BC
- Diffusion Policy for a Car Racing Simulator
- Human-Guided Exploration in RL

Double dipping is encouraged

- Combine with work you're already doing, e.g., ugrad thesis, MS thesis, dissertation research, hobby project.

- I'm open to you double dipping between classes, but you need to get permission from me and the other instructor.

Project Ideas

- **Reproducibility challenge.**
- **Add an idea from a paper we read in class to a research project you are already working on.**
- **Improve an existing approach.**
- **Apply an existing algorithm to a different domain.**
- **Stress test, analyze, and compare and contrast existing approaches.**
- **Formalize and solve a new problem.**

Goal

- Write a report that could be submitted as a workshop paper to an AI venue.

Some Ideas

- Deep TAMER ++ (RL, RLHF, demos, diverse exploration, continuous action spaces, combine with LLMs)
- Interpretable behavioral cloning
- Assisting users in designing good reward functions through multiple feedback modalities
- Inverse RL to learn intrinsic reward functions (curiosity, exploration)
- Multi-Agent RLHF, TAMER, Eureka, etc.
- Optimizing loss functions for RLHF (is Bradley Terry the right model?)

More Ideas

A > B

- Reward hacking of human oversight (human raters are the faulty reward)
- Expectation alignment of AI systems

Capability

- Human-AI collaboration
- AI for assisting and empowering human decision making
- Constraint Learning
- Combining a sparse reward with IRL or RLHF
- Imitation-Bootstrapped RL

More Ideas

- Bandit - NeuralUCB
- Using foundation models to assist in reward, policy, or constraint learning
- Representing uncertainty in LLMs and VLMs
- RLHF with non-stationary/changing preferences
- Adversarial attacks and defenses on RL, IRL, BC, RLHF
- Based on “Perils and Pitfalls of Reward Design” propose and evaluate a “better” way to evaluate and benchmark RL algorithms.
- Helping humans give better feedback to AI
- Are humans good at knowing how many demos to give an AI?

IRL, TAMER

Poisoning

More Ideas

- Are people better or worse at inferring rewards than Inverse RL algos?
- What types of feedback (demos, reward design, clicker, prefs) work best in different tasks? What do people prefer?
- Countering Inverse RL (BC). Can you act in a way such that an observer can’t recover your reward (policy)?
- How to do practical imitation learning for teams of robots or for human-AI teaming?
- RL/policy unit testing. Can we design a small “driver’s test” for an AI system that indicates overall performance.