CS 4180/5180: Reinforcement Learning (Spring 2026) Chris Amato and Rob Platt

Project Proposal: Due (Midnight) February 23rd, 2026

Project Paper: Due (Midnight) April 22nd, 2026

1 Project Description

The final project should be a study of some area of RL of interest to you. For example, you might develop a game playing RL agent for version of poker, go or some other game. The game playing agent might be based on any of the methods we've discussed in class, will discuss in class or anything else that is RL. Alternatively, you might use a common video game (e.g., Atari, Doom or Starcraft) or robotics domains that have or have not been studied before in RL. It doesn't matter which algorithm or application you choose to study. The key requirement is that you think about how the methods we studied in class, and other methods that we did not study, can be applied. Some examples include:

- Recreate algorithm and results from a paper (analyzing whether it produced the same results as expected)
- Compare and analyze some algorithms we studied on a problem of your choice
- Implement an algorithm, run and analyze on a new domain (along with some baseline algorithm from the field)
- Improve an algorithm and compare with original version

You just want to make sure that the algorithms and domain are sufficiently sophisticated (e.g., you don't want to just do Q-learning on a grid-world). Specifically, you shouldn't use any algorithm that are already implementing in some homework. Conversely, some projects are ambitious or time-consuming (e.g., deep reinforcement learning and LLM-based methods often require a long time to train and debug), so make sure you schedule the work to make sure you have something to show at the end.

We encourage students to work in groups of two (or potentially three), but you may also work alone. Projects with three or more students need explicit permission before a project proposal can be submitted. In any case where there is more than one student, each student should work on a different sophisticated algorithm (that aren't simple variants of each other).

2 Timeline and Deliverables

2/23/2026 Project proposal due. Please submit a one or two-page document (as a pdf) describing a proposed problem and solution via Canvas. We will review all project proposals. Some projects will be given the go-ahead via email while we will ask to meet with other project groups. The proposal must show that you have read background material on your topic and are qualified to undertake what you propose to do. It should include full references for the papers and other sources that you have consulted and that will form the foundation for your work. If you are working in a group, each student should work on a different algorithm and what each student

will work on should be explicitly listed in the proposal. It must specify (using the same titles and questions as below):

1. Problem description: What problem are you solving? Describe the problem *formally* from a computational perspective. What are the states, actions and rewards? What simulator or domain are you using (exactly)? Why is it interesting? Does it already have an interface for RL? Have other researchers already used it?

2. Algorithms: What algorithms do you use (exactly)? What baseline methods will you compare to? Why are these algorithms appropriate? How are these algorithms typically used, and how are you using them? Have other people use similar algorithms to solve your problem before?

3. Results: What results do you expect to show? What comparisons will you do? Are there risks for not getting all the results? If so, what will you do about it?

4/22/2026 Final project due. The papers should be written using the AAAI format (for the AAAI Conference on Artificial Intelligence): `https://aaai.org/authorkit26-1/`. Of course, we don't expect these projects to be submitted to the conference (although you are certainly welcome to!), it is helpful to look at papers from previous years to get an idea how they are written. AAAI papers (and papers from several other conferences) for many years can be found at this link: `http://www.aaai.org/Library/conferences-library.php`. Some sample project reports will also be made available. Also submit the code used to produce your results as a separate link or zip. Your report can be organized differently, but the general organization is the following:

1. Abstract: A short summary of what problem you are solving, how you solved it and what the results are.

2. Introduction: A longer description motivating the problem and solution method.

3. Background: Any background information needed to understand the methods used in the project (e.g., a description of a general search problem or some simpler algorithms that you build off of).

4. Related work: What other methods could be applied to your problem, why didn't you use them and how they relate to your method.

5. Project description: What you actually did in formal detail (with algorithms, equations, etc.).

6. Experiments: A description of how you chose these experiments, how the experiments were run, what the results were and why you got these results (Under what circumstances does the algorithm solve your problem successfully? When does it fail?) Again, they should be formal, often with graphs. These results could also include analysis such as a comparison of different methods and performance on different variants of the problem.

7. Conclusion: A summary of the results and what you learned by trying to complete this project.

Project reports must be detailed and self-contained, explaining the problem, methods and results.

Note: there can not be any extensions on the final project!