Syllabus Spring 2021

CS 253/CSYS 395D: Reinforcement Learning, 3.0 credit hours

Course Meeting Logistics

Meeting times and locations: Aiken Center, 102, WMF 9:40-11:00am (https://www.uvm.edu/map/)

Modality: This is a mixed-modality course. Monday meetings will be on MS Teams. Wednesday and Friday meetings will be synchronous, in-person. Due to Covid-19 capacity guidelines, the room has a maximum occupancy of 19. There will polls posted in the Course Meetings channel for you to signal your intent to attend, to help manage classroom capacity.

Instructor Information

Instructor: Prof. Emma Tosch – Emma.Tosch@uvm.edu

Teaching Assistant: Amanda Bertschinger – Amanda.Bertschinger@uvm.edu

Student hours (aka "office hours") are listed in Teams in the Course Schedule tab.

Course Description

In this course, we will cover the fundamentals of reinforcement learning training algorithms. Reinforcement learning is a subdomain of artificial intelligence that studies sequential decision-making. We will cover classical methods for learning the function that the sequential decision-maker uses and connect these classical methods to more recent advances in the field.

This is a project-based course. Readings and videos should be reviewed before class. Class meeting times will be student-directed.

Course Learning Objectives

The learning objectives for this course are two-pronged: there are domain-specific objectives, as well as general project/research/methodological objectives.

Content/domain-related objectives:

- Recognize when reinforcement learning (RL) is an appropriate solution to a problem
- Identify the strengths, weaknesses, and tradeoffs of different RL training algorithms

- Build intuitions of the RL framework through a semester-long project on a special topic

Methodological objectives:

- Plan a project by enumerating its constituent parts and milestones
- Execute a project plan and develop alternative approaches and hypotheses, if applicable
- Enumerate measurable satisfactory objectives on a project, and evaluate whether the project outcomes satisfy those objectives

Required platforms and software

We will primarily be using Teams to organize course materials. Quizzes, programming assignments, and project proposals and reports will all be managed on Blackboard.

Required Course Materials

All course materials will be posted in the Files tab of the General channel in Teams.

Reinforcement Learning: Theory and Algorithms, *Agarwal, Jiang, Kakade, and Sun* (**AJKS**). The pdf for the book is free online:
https://rltheorybook.github.io/rltheorybook_AJKS.pdf

Reinforcement Learning, 1st edition, *Sutton and Barto* (**SB1**)
The pdf for the book is free online:
https://www.csee.umbc.edu/courses/graduate/678/spring17/RL-3.pdf

Reinforcement Learning, 2nd edition, *Sutton and Barto* (**SB2**)
The pdf for the book is free online: http://incompleteideas.net/book/RLbook2020.pdf

A note on how to read the texts: **AJKS** *gives RL a rigorous mathematical treatment from first principles, but currently lacks narrative.* **SB1** *and the later edition* **SB2** *are more narrative, but looser in their formal treatment. Some students may prefer one text over another and may find one text more useful than the other. These texts are largely for reference and should be considered supplementary to the lecture videos.*

There will also occasionally be assigned papers. These will be posted in Teams.

Graded Evaluation

Evaluation will consist of approximate weekly online **quizzes**, periodic **programming assignments** with written analyses or reflections, and a **final project**. Therefore, this course will employ and strengthen skills in formal mathematical reasoning, programming, and writing. There will be no exams.

The objective of the **quizzes** is for students to test their own understanding of the core technical concepts. Each quiz is graded pass/fail. Students may re-take quizzes until they pass. Quizzes will be auto-graded.

The objective of the **programming assignments** is to give students hands-on experience with core RL concepts. Programming assignments may not have a single solution, and may vary significantly between students. Students will self-evaluate their programming assignments. Each assignment will include a rubric. The grading rubric may include short written reflections on choices made during implementation. Instructional staff will verify student's self-grading.

This course is heavily weighted in favor of **projects**. Students will iterate over their project proposals throughout the semester, setting individual goals and evaluation criteria over the first three weeks of the course. Once students and the instructor have determined an appropriately scoped project and evaluation criteria, if a student meets those criteria, they will earn an A on the project section.

Students taking this course for *graduate* credit should complete their projects individually. **Students taking this course for *undergraduate* credit** are strongly encouraged to work on their projects in pairs.

A grade of 70/100 will lower bound a B grade, with higher grades uniformly distributed over the remaining points.

Quizzes	10pts	There will be 10 quizzes, each worth 1 point. Quizzes are graded pass/fail where pass == 100% correct answers. Students may retake quizzes until they pass.
Programming Assignments	20pts	There will be 4 programming assignments, each worth 5 points.
Project	70pts	All projects share a 45 point common structure and 25 points of project-specific structure:

	10pts	2pts for each of 5 mandatory attendances and any associated follow-up questionnaires
	10pts	Project Proposal
	10pts	Project milestones and goals
	10pts	Midterm milestone report
	5pts	Project presentation
	25pts	Project-specific grading criteria

All work will be graded on a rolling basis. The only hard deadlines are for project-related components. These deadlines ensure that you are on track to complete your project by the end of the semester. While we will accept late work on project milestones, we cannot guarantee that late work will be reviewed in a timely fashion, which endangers your ability to complete all of the project requirements before the end of the semester.

Quizzes and programming assignments have recommended due dates in the course schedule on Teams and in Blackboard, but can be completed on a rolling basis. This ensures that students experiencing any pandemic-related difficulties have some slack to tend to both physical and mental health. However, ***we strongly recommend you follow the course schedule for handing in work***. This will ensure that: (1) all students are working through the same material contemporaneously, (2) that each of you have sufficient time at the end of the semester to complete your projects, and (3) that course staff do not become overloaded with grading. The last possible day to turn in all work for the class is 5/10/21. Do not wait until then.

Attendance Policy and Classroom Environment Expectations

Attendance will only be graded for five class meetings; these meetings are devoted to discussing course projects. All other synchronous meetings are optional.

Students have the right to practice the religion of their choice. If you observe a religious holiday occurs that on a Monday and require accommodations, please notify me of the date *before the add/drop deadline (Feb 12)*.

Most class meeting times should be thought of as “office hours++.” Synchronous meetings will be used for student-driven discussion, and may occasionally include review of materials, based on class performance on quizzes or programming assignments. For example, if most students

appear to be struggling with a particular concept, as reflected in their quiz grades, we may devote a class meeting time to discussing that concept. Otherwise, students may use synchronous meeting times to ask about any of the assigned reading materials or videos. Students may also ask about quiz questions, programming assignments, and project ideas, problems, etc.

Students should feel free to attend in-person synchronous meetings, but use that time to read, watch videos, etc., *provided that they do not disrupt others*. ***The purpose of in-person meetings in this course is to aid students who find working in a novel environment helpful for their productivity.*** Think less "lecture space," more "study center." Students who are taking courses on campus should feel free to use the physical classroom space during virtual meeting times.

Students should follow UVM guidelines for safe in-person meetings:

> The Green and Gold Promise clearly articulates the expectations that UVM has for students, faculty, and staff to remain compliant with all COVID-19 recommendations from the federal CDC, the State of Vermont, and the City of Burlington. This include following all rules regarding facial coverings and social distancing when attending class. If you do not follow these guidelines, I will ask you to leave the class. If you forget your mask, you cannot enter the class and should go back and retrieve your mask. The Code of Student Conduct outlines policies related to violations of the Green and Gold Promise. Sanctions for violations include fines, educational sanctions, parent notification, probation, and suspension.

We do not anticipate needing to record any meetings. If this changes, the class will be notified and this document updated with the policy.

Intellectual Property Statement/Prohibition on Sharing Academic Materials

Students are prohibited from publicly sharing or selling academic materials that they did not author (for example: class syllabus, outlines or class presentations authored by the professor, practice questions, text from the textbook or other copyrighted class materials, etc.); and students are prohibited from sharing assessments (for example homework or a take-home examination). Violations will be handled under UVM's Intellectual Property policy and Code of Academic Integrity.

Please don't do it! I don't like providing free labor for the enrichment of our corporate overlords. ☹

Tips for Success

Course-specific study/preparation tips

Here are a few resources for students on remote/online learning:

- Checklist for success in https://learn.uvm.edu/about/support-for-students/checklist-online-credit-courses/
- Academic support for online courses: https://www.uvm.edu/academicsuccess/online-learning-student-resources-remote-instruction
- 30-minute webinar on online learning success (Mar 2020): https://www.youtube.com/watch?v=Xp_MYsqQyvE

Helpful resources other than the professor (e.g., Undergraduate/Graduate Writing Center, Supplemental Instruction, Learning Co-op tutors, supplemental course materials)

Student Learning Accommodations

In keeping with University policy, any student with a documented disability interested in utilizing ADA accommodations should contact Student Accessibility Services (SAS), the office of Disability Services on campus for students. SAS works with students and faculty in an interactive process to explore reasonable and appropriate accommodations, which are communicated to faculty in an accommodation letter. All students are strongly recommended to discuss with their faculty the accommodations they plan to use in each course. Faculty who receive Letters of Accommodation with Disability Related Flexible accommodations will need to fill out the Disability Related Flexibility Agreement. Any questions from faculty or students on the agreement should be directed to the SAS specialist who is indicated on the letter.

Contact SAS
A170 Living/Learning Center;
802-656-7753
access@uvm.edu
www.uvm.edu/access

Important UVM Policies

Religious Holidays

Students have the right to practice the religion of their choice. If you need to miss class to observe a religious holiday, please submit the dates of your absence to me in writing by the end of the second full week of classes. You will be permitted to make up work within a mutually agreed-upon time. https://www.uvm.edu/registrar/religious-holidays

Academic Integrity

The policy addresses plagiarism, fabrication, collusion, and cheating.
https://www.uvm.edu/policies/student/acadintegrity.pdf

Grade Appeals

If you would like to contest a grade, please follow the procedures outlined in this policy:
https://www.uvm.edu/policies/student/gradeappeals.pdf

Grading

For information on grading and GPA calculation, go to https://www.uvm.edu/registrar/grades

Code of Student Conduct

http://www.uvm.edu/policies/student/studentcode.pdf

FERPA Rights Disclosure

The purpose of this policy is to communicate the rights of students regarding access to, and privacy of their student educational records as provided for in the Family Educational Rights and Privacy Act (FERPA) of 1974.
http://catalogue.uvm.edu/undergraduate/academicinfo/ferparightsdisclosure/

Promoting Health & Safety

The University of Vermont's number one priority is to support a healthy and safe community:

Center for Health and Wellbeing
https://www.uvm.edu/health

Counseling & Psychiatry Services (CAPS)
Phone: (802) 656-3340

C.A.R.E. If you are concerned about a UVM community member or are concerned about a specific event, we encourage you to contact the Dean of Students Office (802-656-3380). If you would like to remain anonymous, you can report your concerns online by visiting the Dean of Students website at https://www.uvm.edu/studentaffairs

Project Descriptions

CS 253/CSYS 395D: Reinforcement Learning, 3.0 credit hours

All students must complete a semester-long project. All projects share a core set of milestones that account for 45 of the 70 available points. The objectives of the project are to: (1) acquire a *deep understanding* of a particular topic and (2) produce a *publicly-available artifact* by the end of the semester.

Basic Structure

Students should choose a project structure from column A and a topic from column B.

Column A: Structure	Column B: Topic
• Survey Paper • Tutorial • Reproduction • Software • Other RL research project	• Inverse reinforcement learning • Safe/fair RL • Options • Deep RL • Planning • Object-oriented RL • State-space aggregation • Generalization • Meta RL • Hierarchical RL • Empirical analysis/evaluation of RL • Causality and RL • Active learning and RL • Sample complexity • Imitation learning and RL • Off-policy evaluation • Applications (medical, advertising, etc.) • *Any other special topic not listed*

Collaboration

Students enrolled for graduate credit must complete their projects individually. Undergraduate students are strongly encouraged to work in pairs. Details on which components are individual and which may be submitted as a group are listed below.

Projects do not need to be unique (e.g., two students not working together can write survey papers on the same topic). Students working on the same topic or using the same form are encouraged to talk to each other and share approaches, resources, and ideas. In some cases, sharing materials may require disclosure. Such cases are covered in the descriptions of project forms below.

Resources

You will be asked to choose your special topic and submit a draft project proposal before add/drop. This means you may be in the position of having to commit to a topic before you understand what that topic is. I do not expect anyone to have knowledge of these topics *a priori*. The first lecture video will give an overview of RL and describe what some of these topics are. I recommend picking the first that seems interesting.

The core of this course covers classical RL training algorithms. The purpose of the project is to allow students to pick one special topic that is (preferably) an active research area and for students to become an expert in that area.

When choosing a topic, I recommend you look into the work of well-known active RL researchers such as:

- Emma Brunskill
- Charles Isbell
- Michael Littman
- Joelle Pineau
- Doina Precup
- David Silver
- Phil Thomas

If you find someone whose work you like, follow them on Google scholar to get email updates on related work. I also recommend **creating a Google scholar alert for your special topic**, especially if you are writing a survey paper or doing some other RL research project.

Reputable venues

Most of the projects involve reading one or more academic research papers. In computer science, most research findings appear in conference proceedings. If you search on Google Scholar, you will be able to find the proceedings of most of these venues, free of charge. Note however that not all venues should be considered equally reputable. For example, anyone can submit work to ArXiv, which is not peer-reviewed (it is a preprint server), and is indexed by Google Scholar. To guide your research, I recommend initially restricting your search to reputable venues, including:

- International Conference on Machine Learning (ICML)
- International Conference on Learning Representations (ICLR)
- Journal of Machine Learning Research (JMLR)
- Neural Information Processing Systems (NeurIPS)
    - Any NeurIPS co-located workshop
- AAAI Conference on Artificial Intelligence (AAAI)
- International Joint Conference on Artificial Intelligence (IJCAI)
- Conference on Learning Theory (COLT)
- Multi-disciplinary Conference on Reinforcement Learning and Decision Making (RLDM)
- International Conference on Artificial Intelligence and Statistics (AISTATS)
- Uncertainty in Artificial Intelligence (UAI)

This list is not exhaustive. Papers in AI-related fields are often posted to ArXiV long before they are accepted for publication. If you are unsure of whether a paper was peer-reviewed and accepted, just search for the paper name. Sometimes RL papers show up in non-RL/AI/ML venues. If you are not sure about a paper, just ask!

Core Milestones

Initial Project Proposal (Monday 8 Feb)

The minimum requirements are a one-paragraph (4-6 sentence) description of your project, including your selections from columns A and B (with as precise a description of the special topic as you can). You may be as detailed in your proposal as you like.

Proposals are due by end of day on Monday. However, do not wait to think about your proposal until then; we will be holding a “carousel networking” event during class on Monday, February 8, during which time you will pitch your proposed idea to other students. Each student will meet with ~7 other students (exact number to be determined by the total enrollment, plus

waitlist) for ~5 minutes, to pitch and workshop their ideas. You should use this time to refine your ideas and build connections with other students. You will earn 2 points for attending and participating in carousel networking.

You will receive feedback on your initial project proposals that will help guide your revisions. The more you write at this stage, the easier the revision will be.

Revised Project Proposal (Monday 15 Feb)

We will again run a "carousel networking" project on Monday, February 15. The final form of the project proposal (~750 words) will be due that evening. The project proposal should be appropriately scoped (i.e., should take 60-80 hours of work) with clearly defined goals. Unsatisfactory project proposals should be revised until they are graded as satisfactory. Satisfactory project proposals earn the student 10 points. Participation in carousel networking will earn 2 points.

Moving forward from this session, all in-class project workshopping will be with three students, for 15 minutes each, worth 2 points each. After this session, you will be asked to give a list of the top 5-7 people with whom you'd be interested in workshopping your ideas. You will meet with the same three people during future project workshopping sessions.

Project Milestones and Goals (Monday 22 Feb)

The project proposal requires you enumerate clearly defined goals. You should spend the week between the project proposal and this deadline breaking down your project into milestones. You may find along the way that there are other goals you wish to accomplish, or goals that seem more suitable to your strengths and interests. At the end of day on Monday, February 22, you will be asked to submit another document containing milestones and revised goals. You should have at least three milestones listed. You will earn 10 points for satisfactory completion of this document. This document will be used to grade the 25 points of project-specific work.

At this point in the semester, students who are working in pairs may begin working together. Paired students may or may not be in each other's project workshopping groups. Paired students should have the same set of milestones but may have different goals. Students may have different goals in cases where they split the work on a project. For example, two students wish to study the effects of variations in RL agent implementation across environments. Student A may be a strong programmer or wish to strengthen their programming abilities. That student's goal might be to implement three deep RL training algorithms. Student B may be more interested in the evaluation. That student's goal might be to do a rigorous evaluation and learn how to use a GPU cluster. In that case, the report should break out the individual goals.

Students who work in pairs where one student's goals or output depends on another should also include a contingency plan for the event that the other student does not complete their part of the project. In the above example, a contingency goal would be for Student B to train existing out-of-the-box agents, in the event that something holds up Student A's work.

Midterm Milestones Report (Monday 29 March)

You will submit a report on your progress on your milestones and goals. You should begin with a paragraph or two on the status of the project and include any findings, interesting leads, or experimental results. You should then directly address milestones. For milestones met, you should include a report of what you learned in the process. For unmet milestones, you should report any difficulties you are having, and any plans for addressing them. You will earn 10 points for satisfactory completion of the milestones report. Students working together may submit a single milestone report. However, that report should include the progress made by both students. If only one student submits a milestone report, and that report only includes progress made by the submitting student, then only the submitting student will receive credit.

Draft Presentations Due (Monday 12 April)

On Monday, April 12 we will have an in-class project workshopping session where students can practice explaining their project goals and findings to their workshop groups. Draft presentations should be uploaded to Blackboard by end of day. Note that presentations need not be in final form, but should have a complete outline and placeholder data. Graduate students' presentations should be close to their final form.

Project Presentations (Wednesday 14 April – Friday 7 May)

The remaining three weeks of the course will consist of project workshopping and presentations. Students taking the course for graduate credit will present first. Final project reports are due on the last day of classes, and it is acceptable to work on the project until then. Project presentations can be for incomplete work, so long as students describe the remaining work, along with a roadmap for its completion.

Project Forms

There may be existing work similar to your form and content combination. If you come across this, be sure to cite and acknowledge. All submitted artifacts should be your original work.

Survey Paper

Survey papers strive for *breadth;* a survey paper reviews the existing work on a particular topic. Good survey papers tell a story about trends and connect the topic to the broader subject area. They summarize existing work, but more importantly they highlight the major contributions of

existing work and explain how pieces connect. Your objective is to write something you'd be happy to upload to ArXiV.

This type of project will involve no coding, no data analysis, and no mathematical formalisms. The majority of the work will be reading research papers and writing.

Examples

Peer-reviewed survey papers in RL, and in AI more broadly, can be hard to find, since (1) there is little incentive to publish them, and (2) textbooks are not common in the field. Some good examples of surveys can be found in the introduction, related work, or background chapters of PhD theses. Note that technical writing in computer science gets easier to read the longer the document. Conference papers are very dense with notation and often require background knowledge. Journal articles are longer and tend to have more narrative and contextualizing information. **PhD theses tend to be the easiest to read because they are meant to be self-contained and thus are fairly accessible to outside readers.**

Final Artifact Requirements

The final artifact for this project is a paper, to be submitted by end of day on the last day of classes with the following requirements:

- 8-12 pages single spaced
- Written in LaTeX (e.g., on overleaf.com)
- 1 in margins (use the `fullpage` or `geometry` package)
- Single spaced
- Content: should cover the background and history of that topic, recent developments, and connections to other areas of computer science
- Should be substantially different from any existing surveys on the topic
- Should use citations and quotations when appropriate

Suggested Milestones

You are free to define your own milestones, but to manage your time properly, you may want to consider writing an annotated bibliography and an outline.

Presentation

I recommend modeling your presentation as a short lecture or TED talk.

Tutorial

Tutorials strive for depth; a tutorial ought to convey the details of a technical problem. Where survey papers strive Tutorials can be either text-only or Jupyter notebooks. Text-only tutorials are only suitable for purely mathematical tutorials. Jupyter notebooks are suitable for coding or data analysis. Note that the architecture of Jupyter notebooks can make loading from local packages or multiple files difficult, due to its path resolution semantics. Therefore, if you intend to write much code, you may want to opt to make your project a software artifact instead.

This type of project will involve a fair amount of writing, and may involve some mathematical derivations, data analysis, or a small amount of coding.

Examples

There are copious RL tutorials on Medium.com that can serve as a reference. There are some well-known text-based tutorials from other machine learning disciplines that can serve as models, Conditional Random Fields: An Introduction, or Bayesian Inference with Tears. Note that the latter would be well-suited to a Jupyter notebook were it written today.

Final Artifact Requirements

The final artifact for this project is a paper or a Jupyter notebook, to be submitted by end of day on the last day of classes with the following requirements:

- 18-12 pages single spaced, or notebook equivalent.
- Should define basic terminology and enumerate assumptions.
- Should include three exercises/problems that illustrate major challenges, along with their solutions.

Presentation

I recommend modeling your presentation as an interactive workshop. If your final artifact is a Jupyter notebook, plan to distribute your Jupyter notebook and have students work through the sections. If your final artifact is a text document, plan to work through problems during your presentation, like it is a lecture.

Reproduction

The machine learning community has recently been faced with a "replication crisis," where results from major papers don't hold up, even under identical inputs. This style of project involves selecting a peer-reviewed research paper (i.e., one that has appeared at one of the approved conferences, or in one of the approved journals) and reproducing its results.

In some cases you will be able to find working code online and may be able to simple re-run the entire experimental pipeline on your local machine, without training new models or configuring

anything. This is a *replication*, which differs from a *reproduction*. At a minimum, a reproduction should vary input, which might mean different data sets, training new agents with different random seeds, or training new families of algorithms. The parameters that need to vary for a reproduction depend on the claims of the paper whose findings you are reproducing.

This type of project will involve writing and reading one academic paper in depth. You may need to read another researcher's source code, and will likely need to read through software package documentation. You will need to reproduce the data analysis. You may find that some analyses were incorrectly applied (e.g., a researcher may improperly applied a statistical test).

Examples

Many major conferences have begun running replication or reproduction workshops or similar events, for example:

- ICLR 2019 Workshop: Reproducibility in Machine Learning
- NeurIPS 2019 Workshop: Reproducibility Challenge
- ICML 2018 Workshop: Reproducibility in Machine Learning

One of the most well-known papers highlighting problems with reproducibility and a paucity of scientific rigor in deep reinforcement learning is the 2018 AAAI paper *Deep Reinforcement Learning that Matters*. All of these resources will be useful for guiding your reproducibility study.

Final Artifact Requirements

The final artifact for this project is a report, to be submitted by end of day on the last day of classes with the following requirements:

- 4-8 pages, NeurIPS formatting
- A 1-2-paragraph summary of the original paper
- A 1-pargraph description of the findings or claims of the original paper
- A clear statement of the pieces you are replicating
- A clear description of the methodology you use to replicate
- A list of the elements you varied, with explanations of why you chose to vary them
- A conclusion paragraph on whether you feel you were able to reproduce the results

Presentation

Your presentation should include a sufficiently detailed summary of the original paper, an explanation of the claims, and a report on how you tested each claim.

Software

Fundamentally, the agent's policy is always encoded as a program. The agent software may reside in an embodied agent (e.g., robot), or it may be part of a pipeline in a larger software system, such as an RL-based optimizer for SQL database queries. Agents may interact with a physical environment where state is the (potentially buggy) output of sensors, or they may interact with simulations of a physical environment. Deep agents, which use neural network architectures for function approximation, further complicate the verifiability of the software stack. As a result, researchers have developed software artifacts to aid in various aspects of RL research, including the development of new training algorithms, environments, tasks, and evaluation methodologies.

Examples

- Autonomous Learning Library: high quality baseline implementations of object oriented RL agents (source code)
- Stable Baselines: high quality baseline implementations of deep RL agents (documentation (source code linked))
- Dopamine: end-to-end workbench for developing novel RL training algorithms (paper and source code)
- Toybox: intervenable and modifiable RL environments for explaining RL agents (workshop paper, longer paper, source code)
- simple_rl: end-to-end workbench for reproducible RL research (workshop paper and source code)
- keras-rl: software shim to connect RL agent implementations with environments (source code)

Final Artifact Requirements

Students may contribute to any of the above open source software. For agent libraries such as the Autonomous Learning Library, Stable Baselines, or keras-rl, students should focus on implementing a missing training algorithm. Students may instead choose to implement software, e.g., a new game for Toybox. For new agent implementations, the artifact should include an analysis of the agent's performance on at least three environments. For new environment implementations, the artifact should include an analysis of three agents' performances on the new environment. The final report should describe the work, implementation choices, and agent-environment analysis.

Presentation

Students should include videos of the agent(s) performance on the environment(s) in their presentation to the class.

Other RL research project

Students may propose their own RL research project. Note that in order for this project to qualify as research, it must be novel and it must address a research question. Ideally, the final artifact of this RL research project should be a research paper suitable for submission to NeuIPS or an equivalent venue. The final artifact should be a NeurIPS paper and the presentation should follow the structure of a 20-minute conference presentation. For graduate students from non-CS fields, you may choose a conference format/structure that comports with your field, but you will need to clear it with me first. For undergraduates interested in research who are looking for a starter project, please contact me directly. I will be offering a starter project to any student who has interest, but all interested students will be working on the same starter project.