ARS - Coursework Guide – 24/25

Version History

1.0	29/09/24	First version.
1.1	12/11/24	Fleshed out marking criteria for task 2 report

Summary

Title: Reinforcement Learning using Gymnasium environments

Hand-in: Programs AND a written report will need to be submitted online via Moodle. Check the module's Moodle page for the precise deadline.

Late policy: The coursework deadlines (task 1 and task 2) are absolute. Late submissions are subject to a 5% deduction of the overall coursework mark per day.

Informal Description

The coursework consists of two tasks as described below. Your aim is to build several reinforcement learning agents and to design, implement and run several basic research-based experiments. You will hand-in software and a report that discusses your work on these tasks. Briefly, task 1 is about implementing some basic RL prototypes (with noise injection and basic modularity) for your chosen environment(s) and identification of key literature, gaps, and research questions, whereas task 2 is about designing, developing and running experiments based on the research questions identified in task 1.

Aims and Outcomes

- If you take the labs seriously, at the end of the semester you should be:
  - comfortable with implementing and modifying reinforcement learning agents,
  - capable of adapting your RL solutions to different kinds of robotic problems with well-defined states, actions and rewards,
  - comfortable with neural network approaches for the mapping of complex high-dimensional states to actions (if you choose to use neural network based RL solutions),
  - comfortable with setting up experiments pertaining to noise and studying and mitigating its impact,
  - comfortable with designing modular AI solutions,
  - capable of scanning the literature in order to understand modern RL techniques, and incorporating/extending these in your own solutions,
  - capable of identifying gaps, and/or weaknesses/limitations in state-of-the-art research, and using this to define research questions for guiding your research,
  - capable of studying and evaluating algorithm performance objectively,
  - capable of designing innovative algorithms and experiments, and reporting the results of these in a clear and well-structured manner.

Rough Timetable

Week	Main Lab	Main activities
1	01/10/24	Getting started. Familiarization with Gymnasium
2	08/10/24	Task 1
3	15/10/24	Task 1
4	22/10/24	Task 1
5	(28|29)/10/24	Task 1. Demos for task 1 – we may need both Mon. & Tue. slots
6	05/11/24	Task 2
7	12/11/24	Task 2.
8	19/11/24	Task 2.
9	26/11/24	Task 2
10	(02|03)/12/24	Task 2. Demos for task 2 – we may need both Mon. & Tue. slots

Laboratory notes

- You will work individually.
- We need to start working hard from the very first day to make the most of the lab sessions. In the first week you will learn the basics of Gymnasium, will experiment with several environments, and will even try some small heuristics on simple control problems (e.g. cartpole).
- Rough time estimation:
    - Total hours: 20 credits ≈ 200 hours
    - Subtract lectures (22 hours) and labs (20 hours) = 200 – 42 = 158
    - Divide the remainder by 12 weeks = 158 / 12 ≈ 13 hours per week for everything else, e.g.: studying, researching, reading, thinking, coding, testing, analyzing, writing.

Getting Started

Preliminary steps

- Check the following three main Gymnasium resources:
    - Farama's general documentation page for Gymnasium.
    - Basic usage page in the above documentation.
    - Gymnasium GitHub page – includes installation instructions.
- Install Gymnasium.
- For the purpose of the coursework it is sufficient to work with the "classic control" set of environments, however do feel free to install and use other categories of environments (e.g. MuJoCo and Atari), if you wish.
- Go through the Basic Usage page.
- You can install Gym on your own machines, or in your local directory in UNM's HPC, or you can also use Google Colaboratory. Please note that in the past there were ways to render environments properly in Colab (e.g. have a look at this tutorial) however this may change from time to time. For an example of a Jupyter notebook for the cart pole example, refer to

the module's Moodle page. I suggest not bothering with rendering, except for some debugging exercises, since performance metrics are the key concern.

- As mentioned, if you want to use any of the MuJoCo environments you can. Deep Mind recently bought MuJoCo and made it open source, which means there are no more licensing issues. You are not required to use MuJoCo, but if you really want to, you are free to install it, and get the environments setup.

- To see what environments are available use:

```
import gymnasium as gym
print(gym.envs.registry.keys())
```

- To better understand some Gymnasium environments consult this Wiki or scroll to "environments" in the Gymnasium's GitHub page, and search for your environment. For example for the cart pole environment have a look at this page.

Try to come up with some heuristic solutions for Cart Pole

- Try to come up with some simple heuristics to keep the pole up based on your understanding of the environment. You can start from and modify the (failing) heuristic example provided in the Moodle page (i.e. sol-H1-cart-pole-v0).
- Difficult? Let's see whether reinforcement learning helps.

Have a look at a Q-learning solution

- Example: s1cart-pole-v0-sol1.
- Try to run the code.
- Read the code. Try to understand it as much as possible, although note, it will only fully make sense once we have done Q-Learning in the lectures.

Task Description

- **Requirements for Task 1:**
  - **Title.** Prototypes, literature, gaps, and research questions.
  - **Prototypes:**
    - **Environment selection.** Select two environments to work on throughout the whole assignment. Select **one environment** from **within** the control category (e.g. CartPole-v1) and **one environment** from **any** category (including the control one). Please recall that different environments may impose significant changes to your reinforcement learning algorithm since, for example, they may involve continual action spaces, or other representational differences. To simplify matters you might want to constrain yourself to environments with discrete action spaces.
    - **Core method required**: reinforcement learning. If you want to use other methods for other integrated modules, that is fine.

        - **Additional requirements:** (1) noise injection at the inputs and/or outputs, (2) some modularity (e.g. RL component and denoising component).
        - **Aim:** for each environment develop at least one viable proof of concept based on RL.
    - **Literature:**
        - **Steps**:
            - Explore the recent RL literature in relation to the topic of noise and or modularity.
            - Select 1-3 good papers from the date range 2022-2023 and highlight their gaps (i.e. limitations and/or open questions/problems). Note that although these 1-3 papers will be your "core/seed" papers, you should still study the literature more broadly (i.e. your report should cite other papers apart from the core papers).
            - Select your gaps for further investigation. Justify your choices.
            - Design at least 2 research questions based on your selected gaps.
        - **Aim**: clearly outline 1-3 selected papers, overall gaps, selected gaps, and research questions. Note that it is crucial for the papers, gaps and research questions to be 100% credible, i.e.: (1) the papers must be recent and good, (2) the gaps must be genuine open problems, and (3) the research questions must sit squarely in the gaps and must point in useful directions.
        - **Constraint 1**: Every student must have a different set of core papers and/or a different set of gaps and/or a different set of research questions (RQs). Once a student has defined their selected papers, gaps, and RQs, they must email them to me, in order for me to check and approve them. Please note that this process will operate on a "first come first served" basis. Please also note that if two students share the same papers, they can still be different in terms of the chosen gaps or RQs, however, it is preferable if all elements are distinct.
        - **Constraint 2**: The selected research questions must include, or focus on, (1) noise, (2) modularity, or (3) both.
- **Requirements for Task 2:**
    - **Title.** Research questions and experiments.
    - **Environment selection.** You must use the **same** two environment you selected for task 1.
    - **Core method required**: reinforcement learning. As before, if you want to use other methods for other integrated modules, that is fine.
    - **Goals**. Keywords: novel experiments and insights. The aim of this task is for you to design, develop, run, and analyze, experiments that address the research questions your listed in task 1. The mains tasks would be: (1) design experiments that address the research questions, (2) implement the experiments, (3) debug and finetune your code, (4) run the experiments and collect results, (5) analyze

the results and assess whether they answered the research questions, (6) either proceed back to step 1 with adjustments to the experiments/solutions, or proceed with additional experiments (depending on time and completion status). Document your findings.

- **Requirements for all tasks (i.e. tasks 1 and 2):**
    - **Performance**. Define one or more valid performance measures, apart from the default/compulsory one, i.e.: the average number of episodes needed before learning a problem (see below for more information).
    - **Evaluation**. Run your experiments and report your results for both of your chosen environments consistently.
    - **Four I's**. Try to maximize your work along the following dimensions: (1) informedness (i.e. it is based on a solid understanding of the literature), (2) innovativeness (i.e. novel), (3) inventiveness (i.e. not technically trivial), (4) impactfulness (e.g. generates new knowledge).
    - **Core themes**. The core themes for both tasks are: (1) reinforcement learning, (2) noise, (3) modularity. Please note that the research questions can be exclusively about noise, **or** modularity, **or** both, however, the models must always include elements of noise **and** modularity.
- **Demo.** Show and explain the performance of your solutions, and the results of your experiments.

Performance Evaluation

- Since you will be injecting noise into your sensor data and/or actions, your results are not directly comparable to solutions on external leaderboards (e.g.: https://github.com/openai/gym/wiki/Leaderboard). Your focus will be on internal comparisons (i.e. your own experimental conditions) and innovation.
- One key performance measure that you should recall is the number of episodes required before solving the problem. In other words, here you are interested in the speed of learning. Care must be taken in being explicit and consistent regarding what constitutes having solved the problem.

Assessment – Overall

Component	Marks (100)	Description	Main Criteria
Task 1 - demo	5	Demo of work so far.	Evidence of understanding of the base code. Evidence of solid understanding of literature, gaps, questions, and innovation.
Task 1 - report	20	Report (1-2 pages) summarizing task 1	Are the core papers (1-3) well explained? Are the overall gaps well identified and explained? Are the selected gaps justified properly? Are the research questions grounded in the gaps, and are they clear, concrete, and heading in the right direction?
Task 2 - demo	5	Demo of work so far.	Evidence of understanding of the base code. Good explanation of gaps, question, experimental design, results, analyses, and conclusions. Solid argumentation vis-à-vis the 4 I's. Strong justifications and arguments. Clear communication.
Task 2 - paper	50	Mini-conference paper (4 pages) summarizing all of the work done on both tasks.	Are the structure, grammar and argumentation of the paper/report good? Are the introduction, background, methods, results and analyses, clear, comprehensive and insightful? Does the paper show critical and creative thinking?
Task 2 - software	20	Multiple files organized with a clear structure.	Is the code complete? Is the code well-designed, clean, elegant, and well commented? Is the code complex/challenging enough?

Assessment Criteria for the Report (task 1) and Paper (task 2)

- 1st an excellent, well-written report/paper demonstrating extensive understanding and good insight.
- 2:1 a comprehensive, well-written report/paper demonstrating thorough understanding and some insight.
- 2:2 a competent report/paper demonstrating good understanding of the implementation.
- 3rd an adequate report/paper covering all specified topics at a basic level of understanding.
- F an inadequate report/paper failing to cover the specified topics.

Report guide (task 1)

- The report for task 1 has no fixed format, as long as it is well structured and well organized. The only constraint is that it should be 1-2 pages long. No appendices are allowed, and to be fair to all, no material on page 3 onwards (if you exceed 2 pages) will be included in the assessment. The font size of the main text should not be smaller than 11.
- This report will exclusively focus on: (1) a very brief summary of your prototypes, (2) brief summaries of your selected core papers, and why they were chosen, (3) lengthier

explanations on the weaknesses/gaps of the papers, (4) an explanation and justification of your selected gaps, and (5) an explanation and justification of your research questions, and how they are grounded in the gaps.

Paper Guide (task 2)

You should design your final report as a conference paper. The paper should contain:

- [8 marks] **Introduction** (about 1 page). Brief explanation of the motivation and main concepts, a problem statement, an extremely brief overview of the key papers and their gaps, the research questions, and a brief summary of your main contributions. **Key marking criteria:** (1) Structure and grammar, (2) Clarity, (3) Comprehensiveness, (4) Argumentation, (5) Insightfulness, (6) Critical and creative thinking.

- [8 marks] **Background** (about 0.5 pages). Brief overview of the field and the key papers closely related to your work (this will include the core 1-3 papers and other relevant papers). The core selected papers with their gaps, and why there were chosen selected, must be clearly explained. **Key marking criteria:** (1) Structure and grammar, (2) Clarity, (3) Comprehensiveness, (4) Argumentation, (5) Insightfulness, (6) Critical and creative thinking.

- [8 marks] **Methods** (about 1 page). A detailed and concise description of how you implemented task 2 (e.g. algorithms and experimental design). **Key marking criteria:** (1) Structure and grammar, (2) Clarity, (3) Comprehensiveness, (4) Argumentation.

- [10 marks] **Results** (about 1 page). An overview of your key results encompassing performance measures and other results leading to insights about the problem and/or your solutions. **Key marking criteria:** (1) Structure and grammar, (2) Clarity, (3) Comprehensiveness, (4) Argumentation, (5) Insightfulness.

- [10 marks] **Discussion** (about 0.5 pages). Your interpretation of the results, your conclusions, and proposed future work. **Key marking criteria:** (1) Structure and grammar, (2) Clarity, (3) Comprehensiveness, (4) Argumentation, (5) Insightfulness, (6) Critical and creative thinking.

- [6 marks] **References & Appendices** (not included in the word count). **Key marking criteria:** (1) Consistency of references, (2) Comprehensiveness of references, (3) Structure and clarity of appendices, (4) Insightfulness of appendices.

Note: Writing a concise report/paper is a core part of the assignment. The total number of pages for your paper (i.e. main sections, excluding references and Appendices) cannot exceed 4 pages (with a minimum page margin of 2.5cm on each side), using single line spacing, a two-column format, and a minimum font size of 11).