Xi'an Jiaotong-Liverpool University
西交利物浦大學

XJTLU Entrepreneur College (Taicang) Cover Sheet

Module code and Title	**DTS307TC Reinforcement Learning**
School Title	**School of AI and Advanced Computing**
Assignment Title	**Coursework 1**
Submission Deadline	**04/May/2026 23:59**
Final Word Count	
If you agree to let the university use your work anonymously for teaching and learning purposes, please type **"yes"** here.	

I certify that I have read and understood the University's Policy for dealing with Plagiarism, Collusion and the Fabrication of Data (available on Learning Mall Online). With reference to this policy I certify that:

- My work does not contain any instances of plagiarism and/or collusion.
  My work does not contain any fabricated data.

By uploading my assignment onto Learning Mall Online, I formally declare that all of the above information is true to the best of my knowledge and belief.

Scoring – For Tutor Use	
Student ID	

Stage of Marking	**Marker Code**	**Learning Outcomes Achieved (F/P/M/D) (please modify as appropriate)**			**Final Score**
		A	**B**	**C**	
1st Marker – red pen					
Moderation – green pen	**IM Initials**	The original mark has been accepted by the moderator (please circle as appropriate):			Y / N
		Data entry and score calculation have been checked by another tutor (please circle):			Y
2nd Marker if needed – green pen					

For Academic Office Use			**Possible Academic Infringement (please tick as appropriate)**	
Date Received	**Days late**	**Late Penalty**	☐ **Category A**	Total Academic Infringement Penalty (A,B, C, D, E, Please modify where necessary) _____
			☐ **Category B**	
			☐ **Category C**	
			☐ **Category D**	
			☐ **Category E**	

DTS307TC Reinforcement Learning

Coursework - Individual Report

Due: 04/May/2026 23:59
Weight: 40%
Maximum score: 40 marks

Overview

The purpose of this assignment is to gain experience in Python programming and the design of reinforcement leaning algorithms. You are expected to implement an RL algorithm that solves a specific environment and provide an explanation of the algorithm's methodology. You are expected to analyse your results, including challenges and your solutions.

Learning Outcomes Assessed

A: Systematically understand the fundamental concepts and principles of reinforcement learning

B: Critically analyse real-life problem situations and expertly map them as reinforcement learning tasks.

C: Mastery of Monte Carlo Methods and Temporal Difference Learning

D: Proficiency in Deep Reinforcement Learning algorithms

Late policy

5% of the total marks available for the assessment shall be deducted from the assessment mark for each working day after the submission date, up to a maximum of **five** working days

Avoid Plagiarism

- Do ***not*** submit work from other students.
- Do ***not*** share code/work with other students
- Do ***not*** use open-source code as it is or without proper reference.

Risks

- Please read the coursework instructions and requirements carefully. Not following these instructions and requirements may result in a loss of marks.
- The assignment must be submitted via Learning Mall. Only electronic submission is accepted and no hard copy submission.
- All students must download their file and check that it is viewable after submission. Documents may become corrupted during the uploading process (e.g. due to slow internet connections). However, students are responsible for submitting a functional and correct file for assessments.
- Academic Integrity Policy is strictly followed.

Individual Report (40 marks)

The primary objective of this coursework is to familiarize students with the PPO algorithm using basic deep learning libraries, enabling them to improve their capability in transferring mathematical and theoretical knowledge into Python implementation, and further their understanding of the actor-critic algorithm.

Algorithm Overview

Proximal Policy Optimization (PPO) is a state-of-the-art reinforcement learning algorithm that optimizes a stochastic policy in an on-policy manner. To ensure stable training and avoid catastrophic performance collapse, PPO utilizes a clipped surrogate objective to prevent the policy update from stepping too far from the current behavior.

The Environment: CarRacing-v3

We will be using the **Car Racing** environment from the OpenAI Gymnasium. This environment features a top-down racing track where the agent must learn to navigate through tiles based on pixel inputs. You can find more details about this environment on their website.(https://gymnasium.farama.org/environments/box2d/car_racing/)

Here's a code snippet for you to get started:

```
import gymnasium as gym
env = gym.make("CarRacing-v3", render_mode="rgb_array")
env.reset()
```

Since CarRacing-v3 is quite computationally expensive for a standard laptop (due to the pixel processing), you might want to consider using a gray-scaling or frame-stacking wrapper to speed up training. Alternatively, you can also use the lab computers, which have GPUs and have all the environment already set up.

The PPO Agent

You will implement an RL agent using PPO to play the CarRacing-v3 environment. The agent will use the standard observation and actions provided by the environment. You may edit the

environment to speed up your training, but your agent must still perform well in the standard environment. (i.e, removing the camera zoom at the beginning is allowed during training, but your agent should still be tested in the original environment.) You should record your training and evaluation process using Tensorboard. You should also record important losses and other data for your analysis later.

The Report

Upon completion of your implementation, you are required to submit a comprehensive technical report. The report should document your engineering decisions, the theoretical grounding of your code, and a critical analysis of the agent's performance.

1. **Introduction**
    - Provide a brief overview of Reinforcement Learning in the context of the *CarRacing-v3* environment.
    - Define the state space (pixels), action space (discrete commands), and the reward structure of the task.
2. **Methodology**
    - **Mathematical Foundation:** Formulate the PPO objective function. Explain the significance of the clipping parameter and the probability ratio.
    - **Advantage Estimation:** Describe your method for calculating advantages (e.g., standard advantage vs. Generalized Advantage Estimation (GAE)).
3. **Implementation Details**
    - Describe your implementation, including any challenges faced and how you addressed them.
    - Explain the structure of your policy and value networks.
    - Detail the training process and hyperparameters used.
4. **Results and Analysis**
    - Present your results (use graphs for better clarity).
    - Discuss the performance of your agent and any trends observed.
    - Briefly compare your custom implementation's stability and sample efficiency against baseline benchmarks (e.g., Stable-Baselines3).
5. **Conclusion**
    - Summarize your key findings regarding the sensitivity of PPO to hyperparameter tuning and the effectiveness of the actor-critic framework in continuous-input environments.

Note: All figures and plots must be clearly labeled with axes titles and legends. Raw code snippets should be kept to a minimum in the report; focus on high-level logic and pseudo-code where necessary.

Important Note

- Do **NOT** use Stable-baselines libraries or any other reinforcement learning specific libraries in your implementation (You may use tensorboard for recording your results).
- Do **NOT** exceed the word count limit of **3000** words for each report, reference and appendix excluded.
- Although you are allowed to use any generative AI tools to assist your work, please keep in mind that you should be using them **responsibly**. (Good use: Improve your report after writing it and always review its output to ensure that it is correct. Bad use: Copy-pasting an entire report from AI without any effort of your own.)

Submission Requirements

Please prepare and submit the following documents:

- A cover page featuring your student ID. This page should be the first page of your report.
- A zip file containing all the source codes and your trained agent model, which should be named using your full name and student ID in the following format: CW1_ID_Name.zip
- One PDF file for your report. The file should be separated from the zip file, which contains your code. The files should be named in the following format: CW1_ID_Name.pdf

Note that the quality of the code, the clarity of your writing, and the format/style of your report will be taken into consideration during the evaluation. The detailed rubric is outlined below.

Rubric

CW1 (40 makrs)	Criteria	Marks
Code Performance	Code runs without errors and performs tasks as specified.	6
Code Quality	Code is well-organized, includes meaningful comments, and uses appropriate variable names.	6
Methodology	Comprehensive coverage of topics with detailed explanations of approaches and methodologies.	6
Result analysis	Insightful analysis of results.	6
Report Quality	Report is well-structured, formatted, and free of grammatical errors.	6
Evidence of Work	All required elements are included and correct.	6
Submission	Follows all requirements for submission	4