Course Project

Code of Honor. All external resources used in the project, including research papers, open-source repositories, datasets, and any content or code generated using AI tools, e.g., ChatGPT, GitHub Copilot, Claude, Gemini, must be *clearly cited* in the final submission. The final report must also include *a clear breakdown of individual group member contributions.* Any lack of transparency in the use of external resources or in reporting group contributions will be considered academic dishonesty and will significantly impact the final evaluation.

Topic	From REINFORCE to Advantage Actor-Critic
Category	Deep RL from Scratch

OBJECTIVE Design and implement a progression of policy gradient algorithms starting from REINFORCE, extending it with a baseline, and finally implementing Advantage Actor-Critic (A2C) [3, 1, 2]. The project aims to compare the stability, sample efficiency, and learning dynamics across these algorithms in classic control environments.

MOTIVATION Policy gradient methods are foundational in Deep RL, but they suffer from high variance. By incrementally building from REINFORCE to A2C, you will gain hands-on experience with variance reduction techniques, temporal-difference baselines, and actor–critic architectures. This project provides an accessible yet research-relevant way to understand why modern policy gradient methods are effective. It also gives you the chance to develop robust implementations of what we learn in the second half of the course.

REQUIREMENTS The final submission should address the following requirements while the details can be freely decided by the group members.

1. Implementation: in this respect, you should
   - Implement REINFORCE from scratch.
   - Extend it with a baseline (value function) to reduce variance.
   - Implement Advantage Actor-Critic (A2C) using a shared neural network backbone for actor and critic.
2. Environment modification: you can use a pre-implemented Gym environment that is *required to be controlled by Deep RL*. Note that basic environments that are handled easily by tabular RL are **not accepted**. To give the implementation some level of novelty, you **must** modify standard Gym environments with **at least one** of the following modifications:
   - Adding *noise* to observations.
   - *Perturbing* actions before execution.
   - Injecting irrelevant *distractor* features into the state vector.
   - Adding *stochasticity or delays* to rewards.

3. Evaluation: the final project should report key evaluation of the implemented algorithms in the modified environment. In this respect, the results should
    - compare learning curves across algorithms,
    - quantify stability, convergence speed, and sample efficiency, and
    - report how environment modifications affect algorithm performance.
4. The results should be elaborated through
    - ablation experiments, and
    - providing discussions to explain observed results

Milestones The following milestones are to be accomplished through semester.

1. Literature Review and Setup
    - Review REINFORCE, baselines, and A2C.
    - Choose environments and finalize environment modifications.
2. Implementation
    - Implement REINFORCE and validate learning on basic environment (not-modified).
    - Add baseline and Advantage Actor-Critic (A2C).
    - Train on modified environment.
3. Evaluation and Analysis
    - Collect and plot learning curves.
    - Compare across algorithms and environment conditions.
    - Perform ablation experiments.
4. Final Report and Presentation

Submission Guidelines The main body of work is submitted through Git. In addition, each group submits a final paper and gives a presentation. In this respect, please follow these steps.

- Each group must maintain a Git repository, e.g., GitHub or GitLab, for the project. By the time of final submission, the repository should have
    - Well-documented codebase
    - Clear `README.md` with setup and usage instructions
    - A `requirements.txt` file listing all required packages or an `environment.yaml` file with a reproducible environment setup
    - Demo script or notebook showing sample input-output
    - *If applicable,* a `/doc` folder with extended documentation
- A final report (maximum *5 pages*) must be submitted in a PDF format. The report should be written in the provided formal style, including an abstract, introduction, method, experiments, results, and conclusion.
  Important: Submissions that do not use template are considered *incomplete.*
- A 5-minute presentation (maximum *5 slides including the title slide*) is given on the internal seminar on Week 14, i.e., *Dec 1 to Dec 5,* by the group. For presentation, any template can be used.

Final Notes While planning for the milestones please consider the following points.

1. You are encouraged to explore innovative approaches as long as the core objectives are met.
2. Training should remain feasible by restricting to simple environments and smaller network architectures. Trade-offs (e.g., fewer episodes, smaller networks) are expected and should be justified.
3. Teams are expected to manage their computing needs and are advised to perform early tests to estimate runtime and training feasibility. As graduate students, team members can use facilities provided by the university, e.g., ECE Facility. Teams are expected to inform themselves about the limitations of the available computing resources and design accordingly.

References

[1] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 1928–1937. PMLR, 2016.

[2] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations (section `https://stable-baselines3.readthedocs.io/en/master/modules/a2c.html`). *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[3] Richard S Sutton, Andrew G Barto, et al. *Reinforcement Learning: An Introduction*, volume 1. MIT press Cambridge, 2018.