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	Using RL for Traffic Signal Control
Category	Applications of RL

Objective Design and implement an RL agent to optimize traffic signal control at one or more intersections. The agent should minimize congestion and vehicle waiting times by dynamically adjusting traffic light phases. The project aims to compare RL-based policies with simple heuristics such as fixed-time or round-robin controllers.

Motivation Traffic congestion is a major urban challenge, leading to increased travel times, fuel consumption, and emissions. Similar to other control problems, RL offers a promising framework for adaptive traffic signal control, where decisions are made based on real-time traffic conditions. Several works have shown that RL agents can significantly outperform fixed-timing strategies in simulations [2, 1]. This project provides hands-on experience in applying RL to smart city applications, highlighting how sequential decision-making can improve urban infrastructure.

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
   - use a lightweight traffic environment such as `Highway-env` or a simplified simulation in Python,
   - implement an RL agent (e.g., DQN, PPO) to control signal phases, and
   - implement heuristic baselines (e.g., fixed-timing, round-robin) for comparison.
2. Environment modification: you can use a pre-implemented traffic environment. To give the implementation some level of novelty, you **must** modify the environment with **at least one** of the following modifications:
   - varying vehicle arrival rates,
   - adding sensor noise to vehicle counts,
   - introducing delays in signal switching,
   - adding distractor features to the state representation.
3. Evaluation: the final project should report key evaluation of the implemented algorithms in the modified environment. In this respect, the results should

- compare average waiting times, queue lengths, and throughput across methods,
- study the effect of environment modifications on agent performance, and
- evaluate stability and convergence speed of training.

4. The results should be elaborated through
   - performing ablation studies, e.g., without noise, with different reward functions, and
   - providing discussion on trade-offs between complexity, generalization, and performance.

MILESTONES The following milestones are to be accomplished through semester.

1. Literature Review and Setup
   - Review RL for traffic signal control.
   - Select and configure simulation environment.
2. Implementation
   - Implement heuristic baselines.
   - Implement RL agent and validate on a simple traffic setting.
   - Extend to modified environments.
3. Evaluation and Analysis
   - Collect and plot traffic performance metrics.
   - Compare RL agent vs baselines under different 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. Students are encouraged to explore different reward functions (e.g., minimizing max queue length vs minimizing total waiting time).
2. Training should remain feasible by limiting to small-scale traffic simulations (single or two intersections).
3. Creativity in environment design and reward shaping is encouraged as long as the core objectives are met.
4. 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] LA Prashanth and Shalabh Bhatnagar. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1640–1645. IEEE, 2011.

[2] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1913–1922, 2019.