



Selected Topics in AI Project

Project's Goal: Apply Reinforcement Learning for Autonomous Decision-Making in Simulated or Real-World Scenarios

Students will design and implement a reinforcement learning (RL) agent to solve a decision-making problem in a simulated or real-world environment. The project will involve selecting an environment, implementing an RL algorithm (e.g., Q-learning, Deep Q-Networks, or Policy Gradients), and evaluating the agent's performance. Teams will also present their work, including a live demo of the trained agent and a documentation of their work.

- **This is a group project for graduation project members.**

Phase 1: Problem Definition and Environment Setup

- Choose a simulation environment (e.g., OpenAI Gym).
- Define the problem (e.g., autonomous navigation, game playing, robotic control).
- Set up the environment and ensure it is compatible with the chosen RL framework (e.g., TensorFlow, PyTorch).

Phase 2: Algorithm Selection and Implementation

- Select an RL algorithm (e.g., Q-learning, DQN, Policy Gradients) and justify the choice.
- Implement the algorithm and integrate it with the environment.
- Begin training the agent and collect initial performance metrics, demonstrate your results with diagrams.

Phase 3: Optimization and Evaluation

- Tune hyperparameters (e.g., learning rate, discount factor, exploration rate).
- Experiment with advanced techniques (e.g., reward shaping, experience replay, neural network architectures).



- Evaluate the agent's performance using metrics such as cumulative reward, convergence time, and robustness

Phase 4: Reporting and Presentation

- Prepare a final report detailing the problem, methodology, results, and conclusions.
- Create a presentation showcasing the project, including a live demo of the trained agent.
- Submit all code, data, and documentation

Evaluation Criteria

- **Technical Implementation (40%)**: Quality of the RL algorithm, environment integration, and optimization.
- **Performance Metrics (20%)**: Effectiveness of the trained agent (e.g., cumulative reward, convergence).
- **Teamwork and Collaboration (20%)**: Evidence of effective teamwork and role fulfillment.
- **Documentation and Presentation (20%)**: Clarity and completeness of the report and presentation.

Delay penalty(20%): note that any failure to meet any deadlines can result in a penalty, any excesses must be reported beforehand to be accepted, otherwise none will be accepted.

GitHub Repository Requirements

- Mandatory use of **GitHub** with **frequent commits** (at least 2-3 commits per member).
- **Commit history must reflect individual contributions** (e.g., code, documentation, experiments).
- Zero marks if the repository is forked/copied from online sources without significant modification.



Unique Project Approval

- Teams must propose unique project ideas (approved by the instructor in Phase 1).
- Identical projects across teams will be rejected unless justified (e.g., different algorithms/environments).

