DA 346: Deep Reinforcement Learning

Final Project Report Guidelines

Overview

This is a **group project**. Only the **Team Lead (Team Member 1)** will submit the final report.

Each team must complete **two connected projects**:

- **Final Project – Part 1: Comparative Study of Deep RL Algorithms**
- **Final Project – Part 2: Design of a Custom Environment and Training of an RL Agent**

Both parts must be included in a **single consolidated report**. The report must be written in LaTeX and submitted as a compiled PDF.

The emphasis is on:

- conceptual clarity,
- mathematical correctness,
- experimental rigor,
- and thoughtful analysis.

Running code is not research. Understanding why it works—or fails—is.

Project Structure

Part 1: Comparative Study of Deep Reinforcement Learning Algorithms

1. Problem Selection

- Choose a **non-trivial RL task**, preferably from a standard benchmark suite:
  - Classic control (CartPole variants, MountainCar, Acrobot)
  - Atari-style environments
  - Continuous control (MuJoCo, PyBullet, Brax)

- The task must be sufficiently complex to expose differences in:
  - sample efficiency,
  - stability,
  - convergence behavior,
  - and policy quality.

Avoid trivial environments where every algorithm "solves" the task in a few thousand steps.

2. Algorithms Studied

- Implement and compare **at least three Deep RL algorithms**, chosen from:
  - DQN / Double DQN / Dueling DQN
  - Policy Gradient (REINFORCE)
  - Actor–Critic methods
  - PPO, A2C, A3C
  - SAC / TD3 (for continuous control)
- You may use standard libraries (e.g., Stable-Baselines3), but blind usage without understanding will be penalized.

3. Formal RL Formulation

Provide a precise formulation of the task as a Markov Decision Process:

$$\mathcal{M} = (S, A, T, R, \gamma)$$

Clearly define:

- State space S (including dimensionality),
- Action space A (discrete or continuous),
- Reward function $R(s, a, s')$,
- Discount factor γ,
- Episode termination conditions.

If approximations or shaping are used, justify them explicitly.

4. Experimental Setup

- Network architectures (policy and value networks),
- Optimizers and learning rates,

- Batch sizes, replay buffers, entropy bonuses, clipping parameters, etc.
- Training horizon (number of steps / episodes),
- Hardware used.

Hyperparameters are part of the experiment, not magic numbers.

5. Results and Analysis

- Learning curves with mean and variance across seeds,
- Final policy performance comparisons,
- Sample efficiency analysis,
- Stability and convergence behavior.

Plots without interpretation are decoration. Explain what they mean.

6. Critical Discussion

- Why do certain algorithms perform better or worse?
- Where do they fail?
- How sensitive are results to hyperparameters?
- What does theory predict—and does practice agree?

Part 2: Custom Environment Design and Agent Training

7. Environment Design

- Design a **custom RL environment** (Gym-compatible preferred).
- The task must be:
  - dynamic,
  - decision-driven,
  - and genuinely interesting.
- Examples include:
  - multi-agent coordination or competition,
  - resource management under uncertainty,
  - navigation with partial observability,
  - strategic games with delayed rewards.

Note: If the optimal policy is obvious to a human in 30 seconds, redesign the task.

8. MDP Specification

Formally define:

- Observation space (fully or partially observable),
- Action space,
- Reward design (and shaping, if any),
- Transition dynamics,
- Episode structure.

Discuss design trade-offs. **Reward design deserves special attention—this is where most RL projects fail.**

9. Algorithm Selection and Training

- Choose one or more suitable Deep RL algorithms.
- Justify your choice based on:
  - environment properties,
  - action space,
  - stability considerations.
- Train the agent and report learning behavior.

Show us both success and failure. Failure analyzed well scores higher than success explained poorly.

10. Results and Evaluation

- Learning curves,
- Policy rollouts or visualizations,
- Quantitative performance metrics,
- Ablation or sensitivity studies (if applicable).

Conclusion

- Summarize insights from both parts.
- Reflect on what Deep RL can and cannot do.
- Discuss limitations, open problems, and possible extensions.

References

Use a consistent academic citation style. Cite all papers, environments, and software libraries used.

Appendix (Recommended)

- Key algorithmic pseudocode,
- Important architectural details,
- Additional plots or diagnostics.

Additional Instructions

- **Code Submission:** Provide a GitHub repository link (submitted separately).
- **Originality:** Custom environments must be original.
- **Deadline:** 5 January 2026.
- **Evaluation Criteria:** Depth of understanding, experimental rigor, originality, clarity, and technical correctness.