CROSS-DOMAIN REINFORCEMENT LEARNING FOR LUNAR LANDER

Team: Demon Slayers

Instructor : Amam Behera

DES-646

PROBLEM DEFINITION & MOTIVATION

Context

- Autonomous landing on extraterrestrial bodies (e.g., the Moon) is a critical task in modern space missions.
- Traditional control strategies (PID, model-based controllers) require precise modeling of dynamics and may struggle under uncertain conditions.

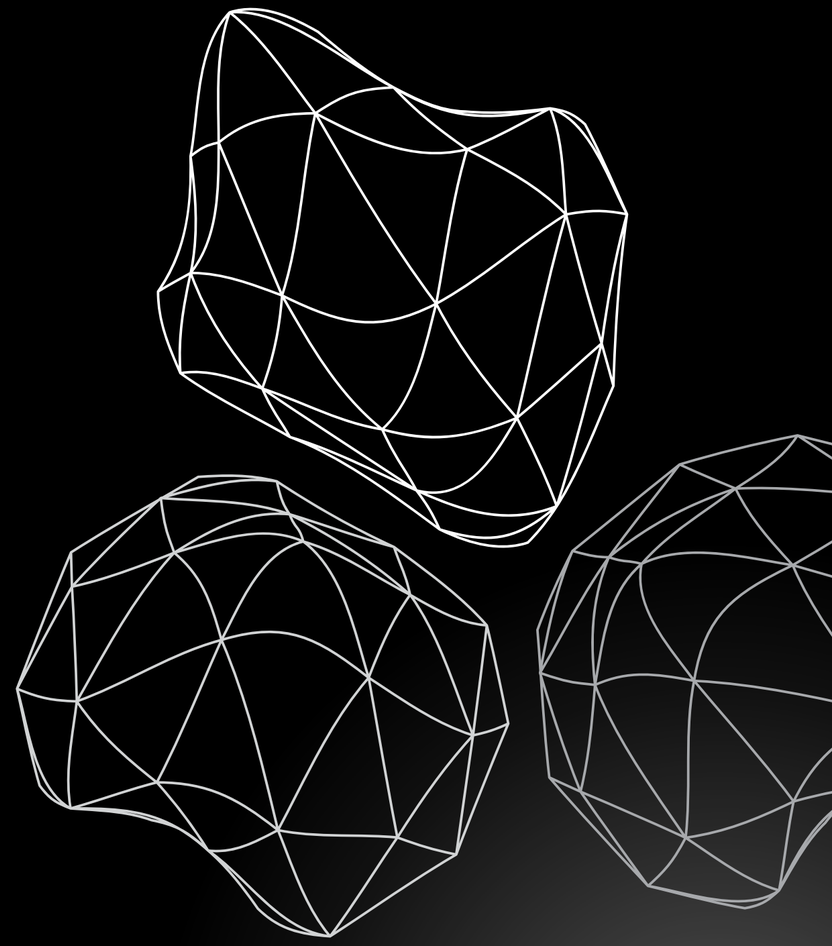
Core Problem

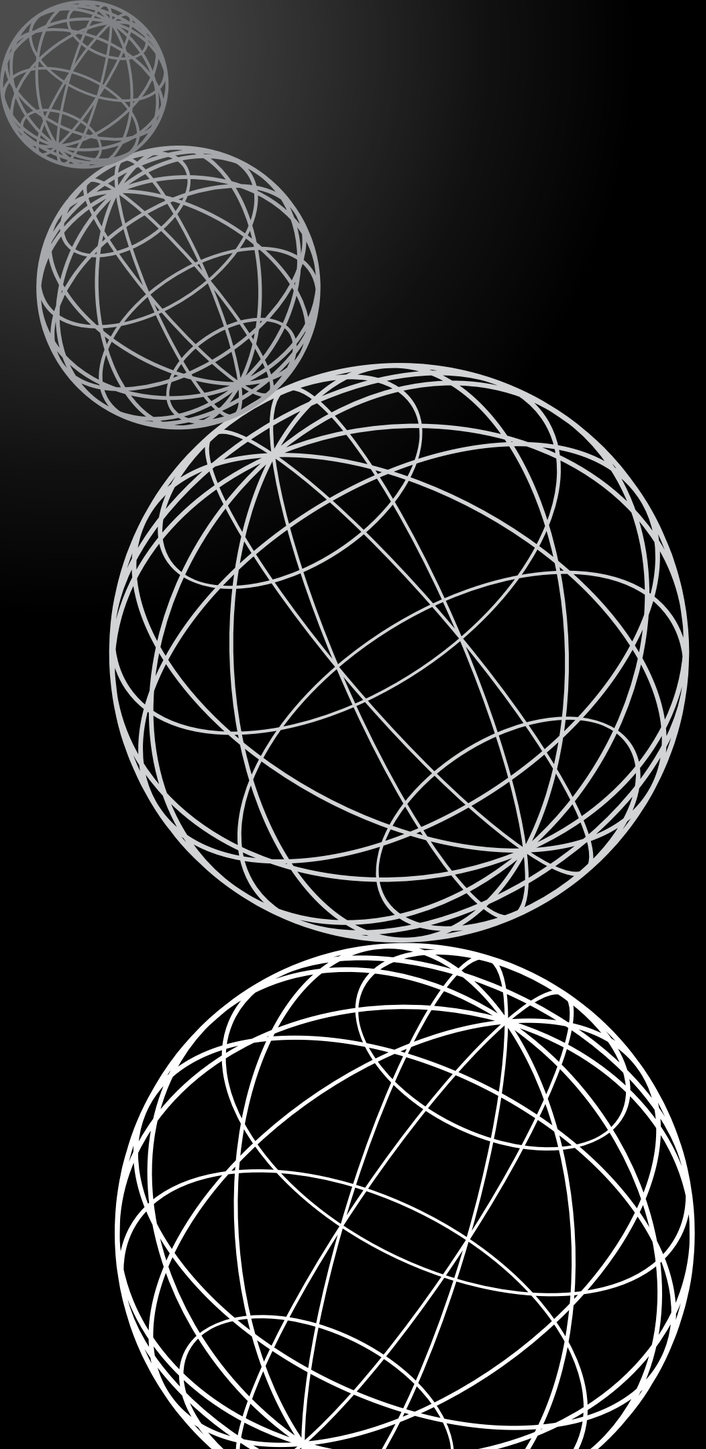
- Design a control policy that can safely and efficiently guide a lunar lander to the surface while handling variations in:

1. Gravity
2. Mass / fuel levels
3. Environmental disturbances and sensor noise

Motivation

- Reinforcement Learning (RL) can learn control policies directly from interaction with the environment.
- Cross-domain RL aims to transfer a learned policy from one environment (source domain) to another (target domain) with different dynamics, reducing retraining effort.





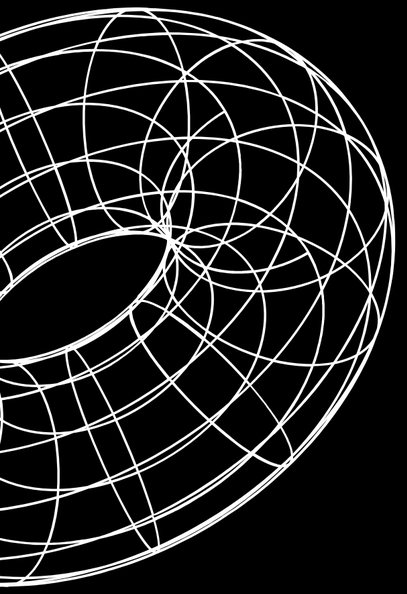
OBJECTIVES AND PROJECT SCOPE

- Primary Objective
  - Develop and evaluate a reinforcement learning–based controller for lunar descent that shows robust behavior across different domains.
- Specific Objectives
  - Model the lunar descent problem using a suitable simulation environment.
  - Train an RL agent to:
    - Minimize landing velocity and fuel consumption
    - Achieve stable, upright landings
  - Introduce domain variations (e.g., gravity, wind, mass, engine thrust) to test cross-domain generalization.
  - Compare performance within-domain vs. cross-domain:
    - Landing success rate
    - Average reward
    - Fuel usage and touchdown smoothness
- Scope
  - Work is restricted to simulation-based experiments (no hardware).
  - Focus on 2D lunar descent (similar to classic Lunar Lander setup).
  - Study limited set of RL algorithms (e.g. Algorithm name: DQN / PPO).

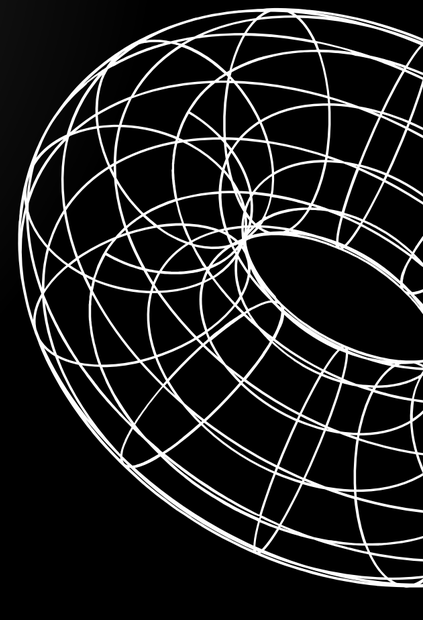
BACKGROUND: REINFORCEMENT LEARNING AND CROSS-DOMAIN TRANSFER

- Reinforcement Learning Basics
  1. RL formalized as a Markov Decision Process (MDP):
     - State sts_tst – describes lander position, velocity, angle, etc.
     - Action ata_tat – engine thrust commands, orientation controls.
     - Reward rtr_trt – feedback signal encouraging safe, efficient landings.
     - Policy π(a ∣ s)\pi(a|s)π(a ∣ s) – mapping from states to actions.
- Goal of RL Agent
  1. Maximize expected cumulative reward
  2. Learn via trial-and-error interactions with the environment.
- Cross-Domain RL
  1. Source Domain: Original environment used for training.
  2. Target Domain: New environment with modified parameters (gravity, mass, noise, etc.).
  3. Aim: Transfer knowledge/policy from source to target with:
     - Minimal additional training
     - Limited performance degradation
- Relevance to Lunar Descent
  1. Different lunar regions or mission configurations may lead to different dynamics.
  2. Cross-domain RL may allow a single robust controller instead of many task-specific controllers.

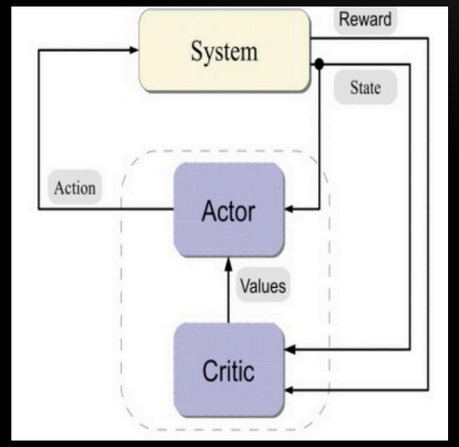
LUNAR DESCENT ENVIRONMENT AND STATE–ACTION DESIGN



- Environment Representation
  1. A 2D simulation representing the lander approaching lunar surface.
  2. The environment provides:
     - Lander’s position (x, y)
     - Horizontal and vertical velocities
     - Lander rotation angle and angular velocity
     - Contact flags for landing legs
- Action Space
  1. Discrete / continuous actions (depending on your setup):
     - Main engine thrust: ON/OFF or continuous thrust level
     - Side thrusters: left/right to control horizontal motion and orientation
- Reward Function (Example)
  1. Large positive reward for successful, stable landing within safe velocity limits.
  2. Penalty for:
     - Crashes or going out of bounds
     - Excessive fuel usage (frequent thrusting)
     - Large tilt at touchdown
- Episode Termination Conditions
  1. Lander touches the surface (safe landing or crash).
  2. Lander moves outside allowed region.
  3. Maximum number of time steps reached.



METHODOLOGY: LEARNING ALGORITHM AND TRAINING SETUP



The methodology adopted in this project is based on a deep reinforcement learning approach, where a neural network is used to approximate either the value function or the policy of the agent. In particular, we employ a modern deep RL algorithm (for example, Proximal Policy Optimization (PPO) or a similar method) which is well-suited for continuous control tasks. The neural network receives the state vector of the lander as input and outputs either action probabilities or continuous control values, depending on the specific formulation.

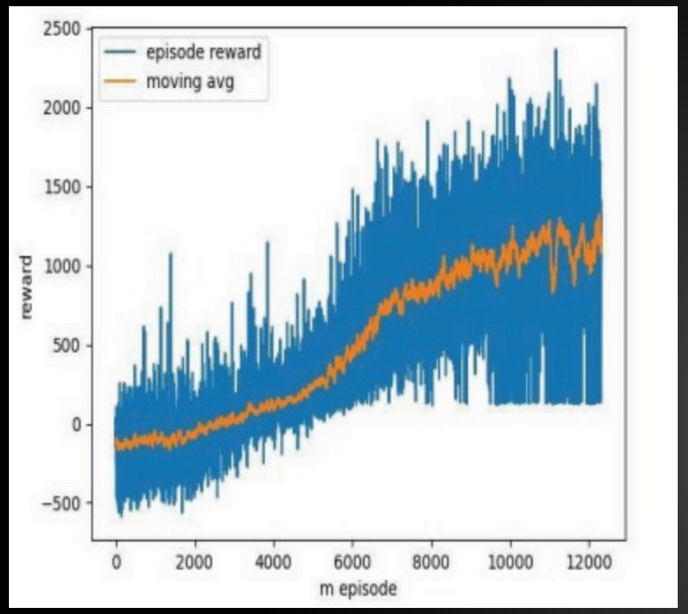
The training procedure involves repeated simulation of episodes in the source environment. During training, the agent starts with a largely uninformative policy and gradually improves it by updating the network parameters in the direction that increases expected cumulative reward. Standard hyperparameters such as learning rate, discount factor, batch size and exploration strategy are selected based on preliminary experiments. Techniques like normalization of input features, reward scaling and early stopping are also considered to stabilize training and avoid divergence. Over a large number of episodes, the agent learns a policy that yields a high success rate of safe landings in the source domain.

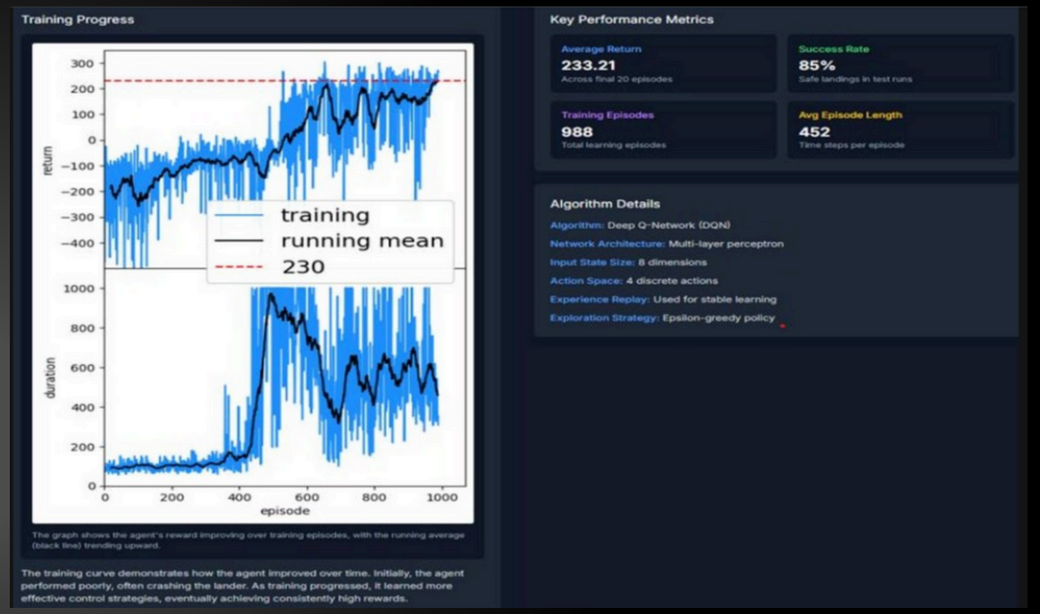
CROSS-DOMAIN EXPERIMENTAL SETUP

To investigate cross-domain behaviour, a clear distinction is made between the source domain and several target domains. The source domain is defined as the nominal lunar environment with a reference gravity value, a fixed lander mass, a specified engine thrust profile and moderate levels of sensor noise. The agent is trained exclusively in this domain until its performance, in terms of landing success and reward, becomes stable.

The target domains are constructed by systematically modifying one or more parameters of the environment. For example, one target domain may have increased gravity, another may involve a heavier lander or reduced maximum thrust, and a third may introduce higher sensor noise or random lateral disturbances. Two transfer strategies are examined: zero-shot transfer, in which the source policy is directly deployed in the target domain without further training, and fine-tuning, in which the source policy is used as an initialization and briefly retrained in the target domain. Performance in each case is evaluated using the same metrics, such as success rate and average cumulative reward.

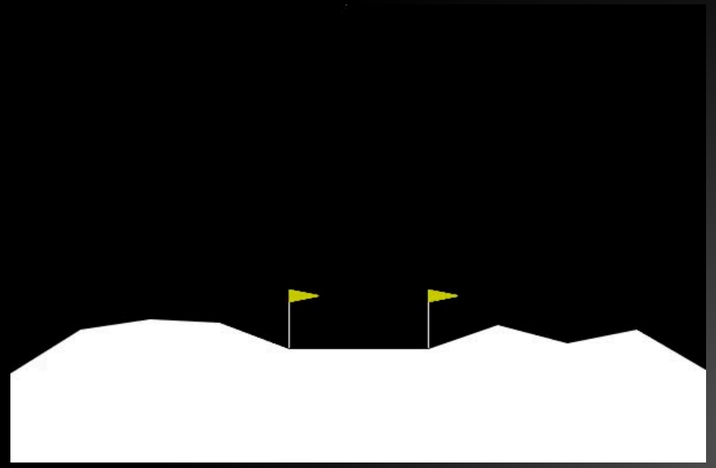
PERFORMANCE ANALYSIS



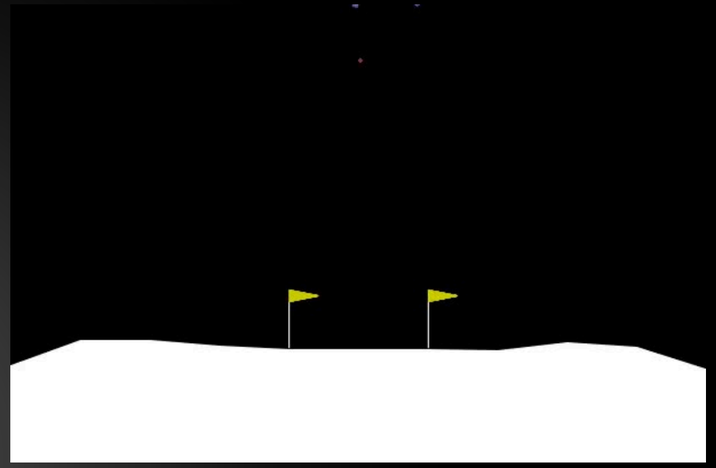



```
Episode 100     Average Score: -171.00
Episode 200     Average Score: -120.86
Episode 300     Average Score: -49.91
Episode 400     Average Score: -42.87
Episode 500     Average Score: -10.98
Episode 600     Average Score: 8.52
Episode 700     Average Score: 185.20
Episode 715     Average Score: 200.11
Environment solved in 615 episodes!     Average Score: 200.11
```

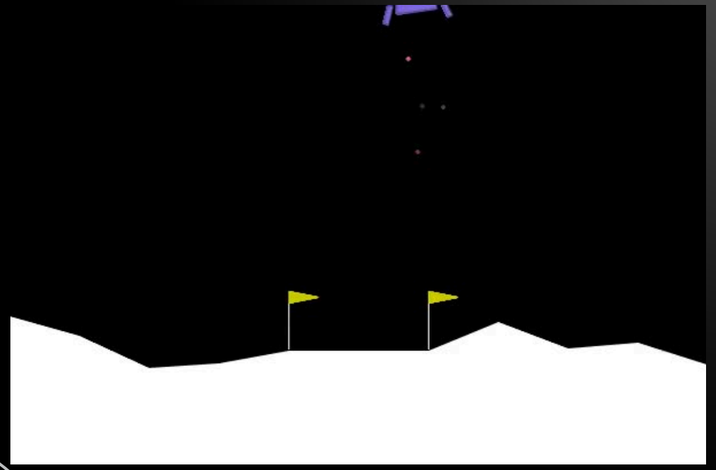
RESULTS



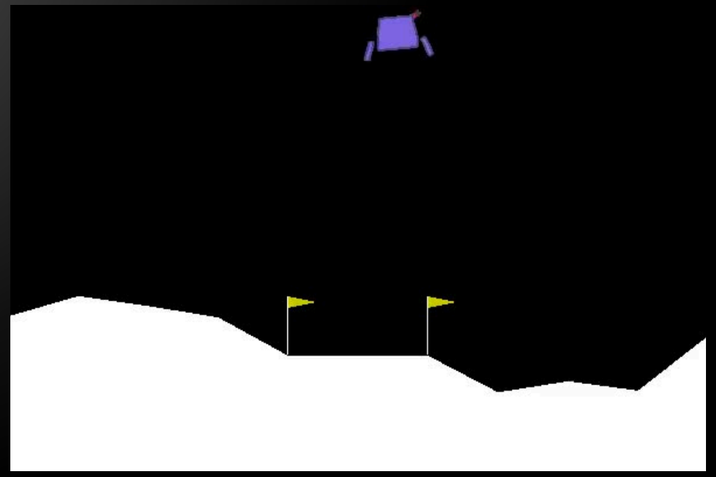
Training



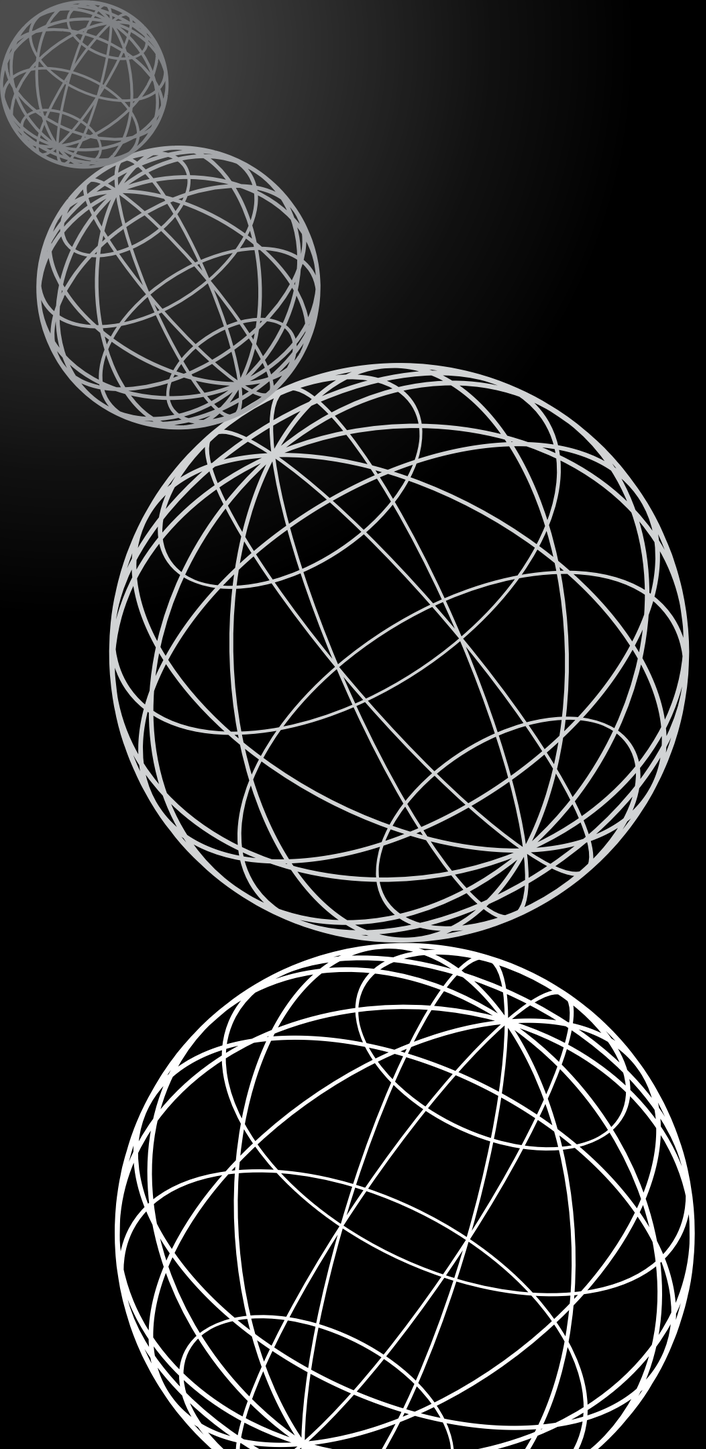
Testing in different gravities



Trained in Earth’s gravity
Failed in Moon’s gravity



Success in Moon’s gravity



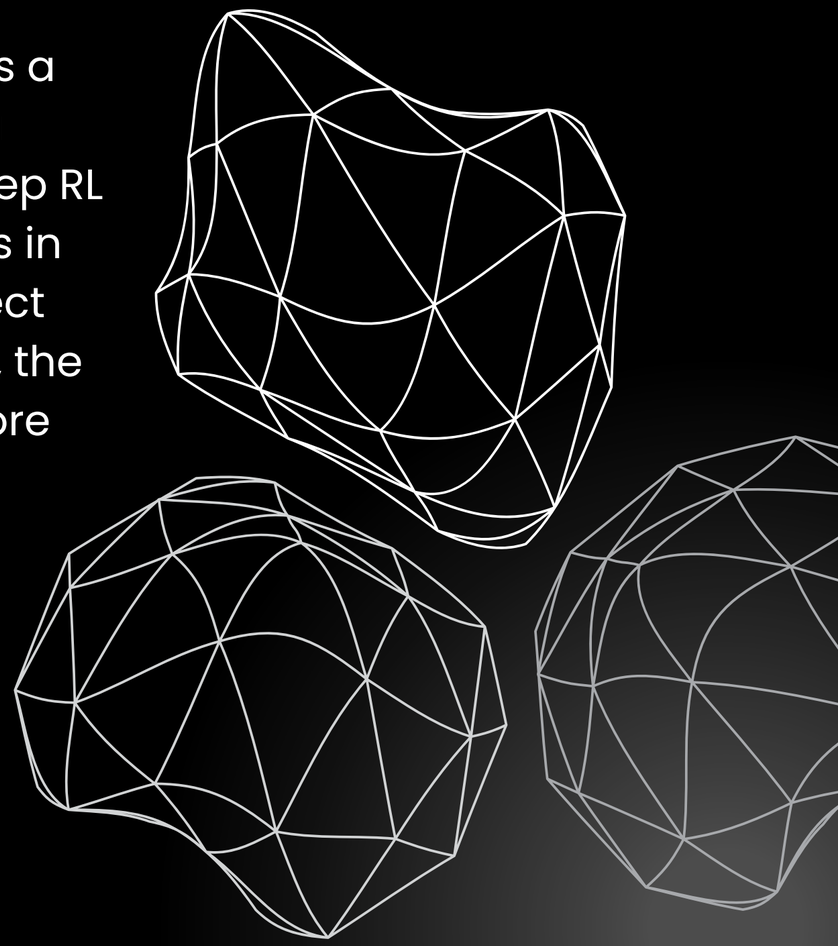
LIMITATIONS AND DISCUSSIONS

- The present study has several limitations that should be acknowledged. First, the lunar lander model and environment are simplified to a two-dimensional representation, which does not fully capture the complexities of three-dimensional motion, irregular terrain and real sensor behaviour. Second, the set of domains considered for cross-domain experiments is restricted to variations in a few physical parameters such as gravity, mass and noise level. Real missions may involve many more uncertainties and constraints, including communication delays and hardware failures.

- In future work, the framework can be extended to a three-dimensional lander model with more realistic terrain and obstacle features. Advanced reinforcement learning techniques, such as safe RL or constrained policy optimization, can be explored to incorporate explicit safety requirements. Furthermore, more sophisticated transfer learning strategies, including meta-learning and multi-task learning, can be employed to enhance generalization across a larger family of domains. Finally, hybrid approaches that combine reinforcement learning with classical control methods may offer a practical balance between robustness, interpretability and performance.

CONCLUSION

- In conclusion, this project demonstrates that reinforcement learning provides a viable and flexible approach for controlling the descent of a lunar lander in a simulated environment. By formulating the task as an MDP and training a deep RL agent, it is possible to obtain a policy that ensures safe and efficient landings in the nominal source domain. Cross-domain experiments show that while direct transfer to altered environments leads to some degradation in performance, the use of fine-tuning and appropriate training strategies can substantially restore and improve robustness in the target domains.



TEAM

SAURABH
220988

VISHESH
221209

HIMANSHU
220455

ADITYA
220069

THANK YOU