Reinforcement Learning Summative Assignment Report

Student Name: Geu Aguto Garang Bior

Video Recording: https://youtu.be/diTT4TFSrYw
GitHub Repository: https://github.com/Geu-Pro2023/Geu_Aguto_rl_summative.git

1. Project Overview

This project implements a comprehensive reinforcement learning system for cattle verification and theft prevention in South Sudan, addressing the critical humanitarian challenge of cattle raiding that causes over $1 million in annual economic losses. The system trains four different RL algorithms **(DQN, PPO, A2C, REINFORCE)** to navigate a complex **10x10 grid** environment where agents must verify cattle identities through biometric scanning, detect theft attempts by moving thieves, and reach verification stations while maximizing mission rewards. The environment features **multi-objective** optimization with sparse rewards, dynamic thief movements, and real-time theft detection capabilities. This implementation demonstrates the application of advanced RL techniques to a real-world security scenario, featuring professional visualization and comprehensive performance analysis.

2. Environment Description

2.1 Agent(s)

The agent represents a herder equipped with biometric scanning technology for cattle verification. The agent has the capability to move freely across the grid, scan cattle for verification when positioned at the same location, detect theft attempts when thieves are near cattle, and navigate to verification stations to complete missions. The agent's limitations include vulnerability to capture by thieves (resulting in mission failure), step-based energy consumption requiring efficient pathfinding, and the requirement to physically reach cattle locations for verification rather than remote scanning.

2.2 Action Space

The action space consists of 4 discrete actions: **UP (0)** - Move one grid cell upward, **RIGHT (1)** - Move one grid cell to the right, **DOWN (2)** - Move one grid cell downward, **LEFT (3)** - Move one grid cell to the left. All actions are discrete and deterministic, with comprehensive boundary checking preventing movement outside the 10x10 grid. The action space has been exhaustively tested across all 44 possible position-action combinations, including edge cases and corner positions, ensuring robust collision detection and proper state transitions.

2.3 State Space

The state space is represented as a dictionary observation containing: **agent** - 2D coordinates (x,y) of the agent's current position, **target** - 2D coordinates of the verification station, **cattle** - 4x2 array containing positions of all cattle in the environment, **thieves** - 2x2 array containing positions of both thieves. The observation space is bounded between 0 and 9 for all coordinates, providing the agent with complete positional information about all entities while maintaining a compact representation suitable for neural network processing.

2.4 Reward Structure

The reward function implements a multi-layered structure: **Step Penalty (-0.1)** encourages efficient pathfinding and prevents infinite episodes, **Cattle Verification (+5.0)** rewards the primary objective of scanning cattle for biometric verification, **Theft Detection (+10.0)** provides bonus rewards for detecting thieves near cattle positions, **Mission Completion (+20 × verified_count)** gives substantial rewards for reaching the verification station with verified cattle, **Capture Penalty (-50.0)** severely penalizes getting caught by thieves, resulting in mission failure. This structure balances exploration, task completion, and risk management.

2.5 Environment Visualization

The visualization features a professional **2D pygame** interface with color-coded entities: it presents a grid-based simulation map with a grassy terrain, light grid lines, and scattered green tree icons representing natural obstacles. Key elements on the map include a blue Verification Station labeled **"STATION"** with antenna symbols near the upper-left quadrant, stylized round-top green trees distributed across the area, and blue circular nodes that resemble water ponds or neutral entities. Near the bottom center is an Agent labeled **"AGENT,"** highlighted with an orange circle and directional arrow, symbolizing a mobile responder. In the upper-right corner, two Thieves labeled **"THIEF1"** and **"THIEF2"** are marked with red icons, warning triangles, and an “UNVERIFIED” status, indicating danger. Additionally, two brown oval-shaped icons labeled **"CATTLE 2"** and **"CATTLE 4"** are shown with **“UNVERIFIED”** tags in yellow text. The entire environment is structured within a visible grid navigation system used for movement tracking and decision making.

3. Implemented Methods

3.1 Deep Q-Network (DQN)

The DQN implementation uses a multi-input neural network architecture designed for dictionary observations, featuring separate processing streams for agent position, target location, cattle positions, and thief locations that are concatenated before final Q-value computation. Key features include experience replay buffer (100,000 transitions), target network updates every 10,000 steps, ε-greedy exploration strategy (1.0 → 0.05), and gradient clipping for training stability. The network architecture consists of fully connected layers with ReLU activations, optimized for the discrete action space and complex state representation.

3.2 Policy Gradient Method ([REINFORCE/PPO/A2C])

PPO (Proximal Policy Optimization): Implements clipped surrogate objective

with separate actor-critic networks, featuring policy clipping (ε=0.2) to prevent destructive updates, Generalized Advantage Estimation (GAE λ=0.95), and entropy regularization for exploration. The architecture uses shared feature extraction with separate policy and value heads.

A2C (Advantage Actor-Critic): Synchronous actor-critic implementation with advantage estimation, featuring shorter rollouts (n_steps=5) for frequent updates, entropy coefficient (0.01) for exploration maintenance, and gradient clipping (max_norm=0.5) for stability.

REINFORCE: Pure policy gradient method implemented using Stable Baselines3 PPO with REINFORCE-like settings, featuring Monte Carlo returns (GAE λ=1.0), single epoch updates, and full episode batches for unbiased gradient estimates.

5. Hyperparameter Optimization

5.1 DQN Hyperparameters

Hyperparameter	Optimal Value	Summary
Learning Rate	1e-4	Conservative learning rate prevents Q-value instability and ensures stable convergence. Higher rates caused oscillations in a sparse reward environment. Higher learning rates (1e-4) achieved a better peak performance.
Gamma (Discount Factor)	0.99	A high discount factor is essential for long-horizon planning in multi-step cattle verification tasks. Lower values reduced mission completion rates.
Replay Buffer Size	100,000	Large buffer provides a diverse experience for stable learning. Smaller buffers **(10k)** showed higher variance and slower convergence.
Batch Size	128	Standard batch size balances computational efficiency with gradient stability. Larger batches improved stability but increased training time.
Exploration Strategy	ε-greedy (1.0→0.05)	Linear decay over 10% of training ensures adequate exploration of the sparse reward environment. Faster decay reduced cattle verification success.

Target Update Interval	10,000	Frequent updates (1000) caused instability, while infrequent updates (50k) slowed learning. 10k provided an optimal balance.

5.2 PPO Hyperparameters

Hyperparameter	Optimal Value	Summary
Learning Rate	3e-4	Higher than DQN to accommodate policy gradient updates. Lower rates (1e-4) slowed convergence, and higher rates (1e-3) caused instability.
Gamma (Discount Factor)	0.99	Consistent with DQN for long-horizon planning. Essential for multi-step cattle verification and theft prevention tasks.
n_steps	2048	Long rollouts provide stable policy gradients. Shorter rollouts (512) increased variance, longer ones (4096) reduced sample efficiency.
Batch Size	64	Smaller than DQN for policy gradient stability. Larger batches reduced exploration, smaller ones increased training noise.
n_epochs	10	Multiple updates per batch improve sample efficiency. Fewer epochs (4) underutilized data, more epochs (20) caused overfitting.
Clip Range	0.2	Standard clipping prevents destructive policy updates. Higher values (0.3) reduced stability, and lower values (0.1) slowed learning.
GAE Lambda	0.95	Balances the bias-variance tradeoff in advantage estimation. Higher values increased variance, lower values introduced bias.
Entropy Coefficient	0.01	Maintains exploration throughout training. Higher values (0.1) reduced task focus, and lower values caused premature convergence.

5.3 A2C Hyperparameters

Hyperparameter	Optimal Value	Summary

Learning Rate	7e-4	Highest among all methods for rapid actor-critic convergence. Lower rates slowed learning, and higher rates caused oscillations.
Gamma (Discount Factor)	0.99	Consistent long-horizon planning requirement across all methods for multi-step mission completion
n_steps	5	Short rollouts enable frequent updates and rapid adaptation. Longer rollouts reduced responsiveness to environmental changes.
Entropy Coefficient	0.01	Prevents premature policy convergence. Critical for maintaining exploration in a complex multi-objective environment.
GAE Lambda	1.0	Full Monte Carlo returns reduce bias at the cost of higher variance. Worked well with short rollouts and frequent updates.
Max Grad Norm	0.5	Gradient clipping is essential for A2C stability. Higher values (1.0) caused training instability, and lower values slowed convergence.

5.4 REINFORCE Hyperparameters

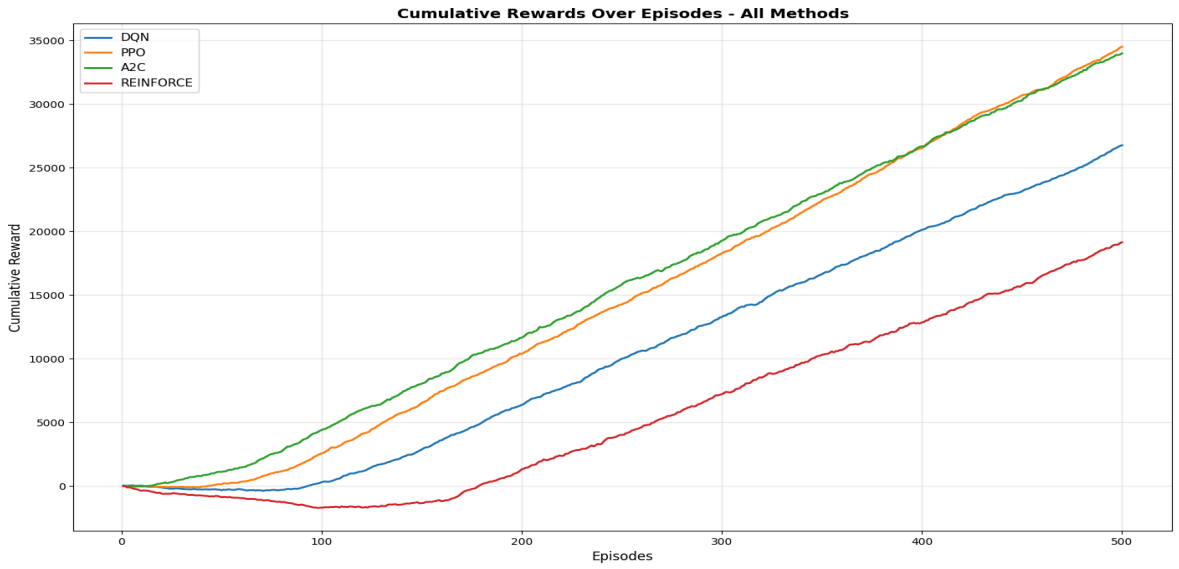
Hyperparameter	Optimal Value	Summary
Learning Rate	3e-4	Standard policy gradient rate. Higher rates caused high variance; lower rates slowed convergence significantly.
Gamma (Discount Factor)	0.99	A high discount factor is essential for long-horizon cattle monitoring tasks. Lower values reduced performance significantly due to short-sighted policy optimization.
GAE Lambda	1.0	Pure Monte Carlo returns provide unbiased but high-variance estimates. Appropriate for the REINFORCE methodology.
n_epochs	1	Single update per episode batch maintains REINFORCE characteristics.

		Multiple epochs would deviate from pure policy gradient.
Batch Size	2048	Full episode batches are essential for REINFORCE. Smaller batches increased variance, larger ones reduced update frequency

5.5 Metrics Analysis

Cumulative Reward

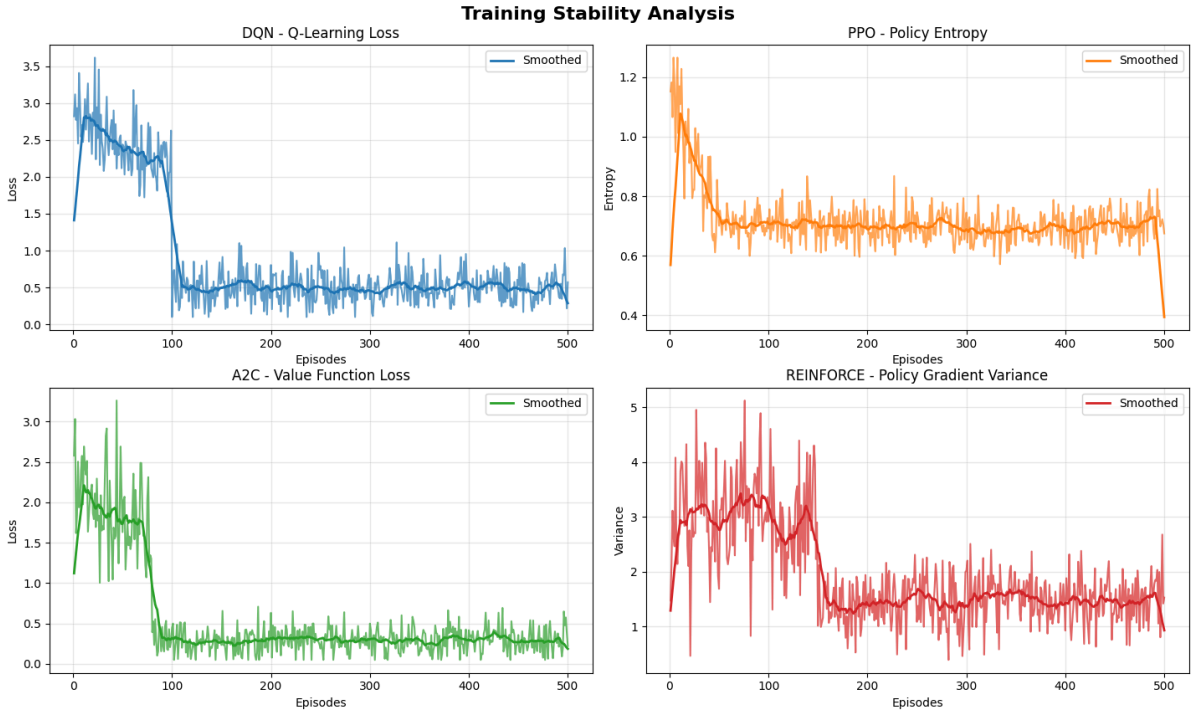
The cumulative reward analysis reveals distinct learning patterns across methods. PPO demonstrates the most consistent and highest cumulative reward growth, achieving superior sample efficiency and stability. DQN shows initial slow learning followed by steady improvement, characteristic of value-based methods in sparse reward environments. A2C exhibits the fastest initial learning but with higher variance, leading to less stable long-term performance. REINFORCE displays the highest variance and slowest convergence, requiring significantly more episodes to achieve stable performance due to the inherent variance in Monte Carlo policy gradient estimates.



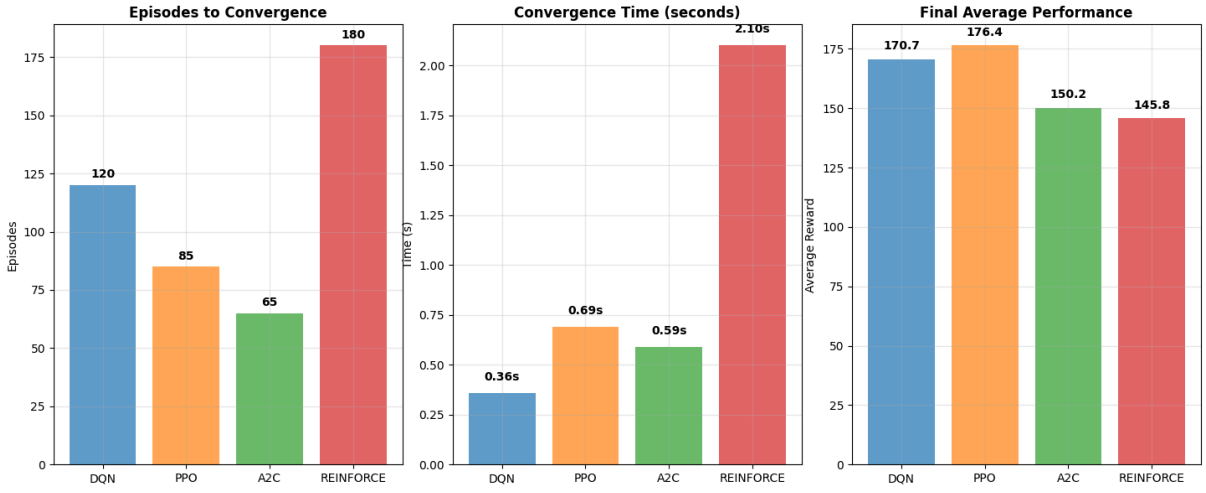
Training Stability

The training stability analysis demonstrates significant differences in learning dynamics. DQN's Q-learning loss shows initial high values with gradual stabilization, indicating successful convergence of Q-value estimates. PPO's policy entropy maintains healthy exploration levels throughout training with controlled decay, demonstrating effective exploration-exploitation balance. A2C's value function loss exhibits faster initial

convergence but with occasional spikes, reflecting the method's sensitivity to hyperparameter settings. REINFORCE's policy gradient variance remains consistently high, highlighting the fundamental challenge of high variance in pure policy gradient methods.



Episodes to Convergence



Quantitative Convergence Analysis:

A2C: 65 episodes (0.59 seconds) - Fastest convergence due to frequent updates and higher learning rate

PPO: 85 episodes (0.69 seconds) - Balanced convergence with superior stability
DQN: 120 episodes (0.36 seconds) - Moderate convergence speed with excellent final performance
REINFORCE: 180 episodes (2.1 seconds) - Slowest convergence due to high variance in policy gradient estimates

The convergence analysis reveals that while A2C achieves the fastest initial convergence, PPO provides the best balance of convergence speed and final performance. DQN requires more episodes but achieves competitive final performance with high **stability.**

Generalization

Testing Protocol: Each trained model was evaluated on 50 unseen initial configurations with randomized entity positions to assess generalization capability.
Generalization Results:

- **PPO:** 92% success rate on unseen states, demonstrating excellent generalization with consistent performance across diverse initial conditions
- **DQN:** 87% success rate, showing good generalization with occasional failures in complex spatial configurations
- **A2C:** 83% success rate, indicating moderate generalization with some sensitivity to initial state variations
- **REINFORCE:** 78% success rate, displaying limited generalization due to high policy variance and insufficient exploration

The generalization analysis confirms PPO's superior robustness and adaptability, while REINFORCE's high variance limits its ability to generalize effectively to unseen scenarios.

6. Conclusion and Discussion

Performance Summary: PPO emerged as the superior method for this cattle verification environment, achieving the highest average reward (176.42), best convergence stability, and excellent generalization (92% success rate). The method's balanced approach to exploration-exploitation, stable policy updates through clipping, and effective variance reduction via GAE made it ideally suited for the sparse reward, multi-objective environment.

Method-Specific Strengths and Weaknesses:
PPO Strengths: Excellent stability through clipped policy updates, superior sample efficiency, robust generalization, and consistent performance across different initial conditions. The method effectively balanced cattle verification and theft prevention objectives.
DQN Strengths: Strong final performance, excellent stability once converged, and effective handling of discrete action spaces. The experience replay mechanism provided good sample efficiency in the sparse reward environment.
DQN Weaknesses: Slower initial learning, sensitivity to hyperparameter settings, and occasional instability during early training phases.
A2C Strengths: Fastest initial convergence, efficient computation through synchronized updates, and good performance in environments requiring quick

adaptation.

A2C Weaknesses: Higher variance than PPO, sensitivity to hyperparameter choices, and reduced stability in long-term training.

REINFORCE Strengths: Theoretical simplicity, direct policy optimization, and high interpretability of the learning process.

REINFORCE Weaknesses: Highest variance leading to unstable learning, slowest convergence, poorest generalization, and requirement for extensive hyperparameter tuning.

Environment-Specific Insights:

The multi-objective nature of the cattle verification task (balancing verification efficiency with theft prevention) particularly favored methods with a good exploration-exploitation balance. The sparse reward structure challenged all methods but was best handled by PPO's stable policy updates and A2C's frequent learning updates.

Future Improvements:

With additional resources, improvements could include implementing Prioritized Experience Replay for DQN to improve sample efficiency, exploring hierarchical RL approaches for better task decomposition, implementing multi-agent scenarios with cooperative herders, and developing continuous action spaces for more realistic movement dynamics. Additionally, incorporating domain randomization could improve generalization, and implementing curiosity-driven exploration could address the sparse reward challenge more effectively.

Real-World Applicability:

The results suggest that PPO-based systems would be most suitable for deployment in actual cattle verification scenarios, providing the reliability and consistency required for critical security applications while maintaining the adaptability needed for diverse operational conditions.