Reinforcement Learning Summative Assignment Report

Student Name: Emmanuel Obolo Oluwapelumi
Video Recording: video rendering agent
GitHub Repository: github repo

1. Project Overview

This project implements a reinforcement learning system to optimize textbook distribution decisions in Rwanda's education system, addressing critical challenges in educational resource allocation. The system simulates an intelligent decision-making agent that evaluates school conditions and determines optimal textbook delivery actions to maximize learning outcomes while minimizing waste. Based on Rwanda's Foundational Learning Strategy 3, the project tackles real-world problems including insufficient textbook-to-student ratios, poor coordination between government grants and actual needs, and inefficient delivery logistics. The approach utilizes a custom Gymnasium environment where an agent learns to make delivery decisions based on comprehensive school characteristics including student enrollment, available resources, infrastructure quality, and historical performance data.Through comparative analysis of four reinforcement learning algorithms (DQN, PPO, REINFORCE, and Actor-Critic), the project identifies optimal strategies for educational resource distribution that can be applied to real-world policy implementation.

2. Environment Description

2.1 Agent(s)

The environment features a single intelligent agent representing a centralized textbook distribution coordinator within Rwanda's Ministry of Education. This agent embodies the decision-making capabilities of education policy administrators like Rwanda Education Board **(REB)** who must evaluate multiple schools simultaneously and determine optimal resource allocation strategies. The agent operates with complete observability of school conditions within its current focus but must learn to generalize across diverse educational contexts including rural and urban schools with varying infrastructure capabilities, student populations, and resource needs. The agent's primary limitation lies in its the fact that multiple states can not check all at once in order to know which schools need the Teaching and Learning materials, and its reliance on simulated rather than real-time school data, which have been simplified from this complex real-world distribution decisions into six distinct policy choices per school.

2.2 Action Space

The action space consists of four discrete actions that represent realistic policy interventions available to textbook distribution coordinators. **Action 0: Send Textbook Batch**; delivers a standard quantity of textbooks to the current school based on assessed needs. **Action 1: Hold Delivery or Flag for Follow-up**; postpones textbook distribution to optimize timing, await better conditions, or marks schools for additional assessment and intervention without immediate resource allocation. **Action 3: Send Teacher Guides**; prioritizes pedagogical support materials over textbooks when teacher preparedness is the limiting factor. **Action 4: Send Limited Supply**; provides reduced textbook quantities to prevent waste in schools with uncertain utilization capacity. These discrete actions capture the essential decision categories while maintaining computational tractability for reinforcement learning optimization.

2.3 State Space

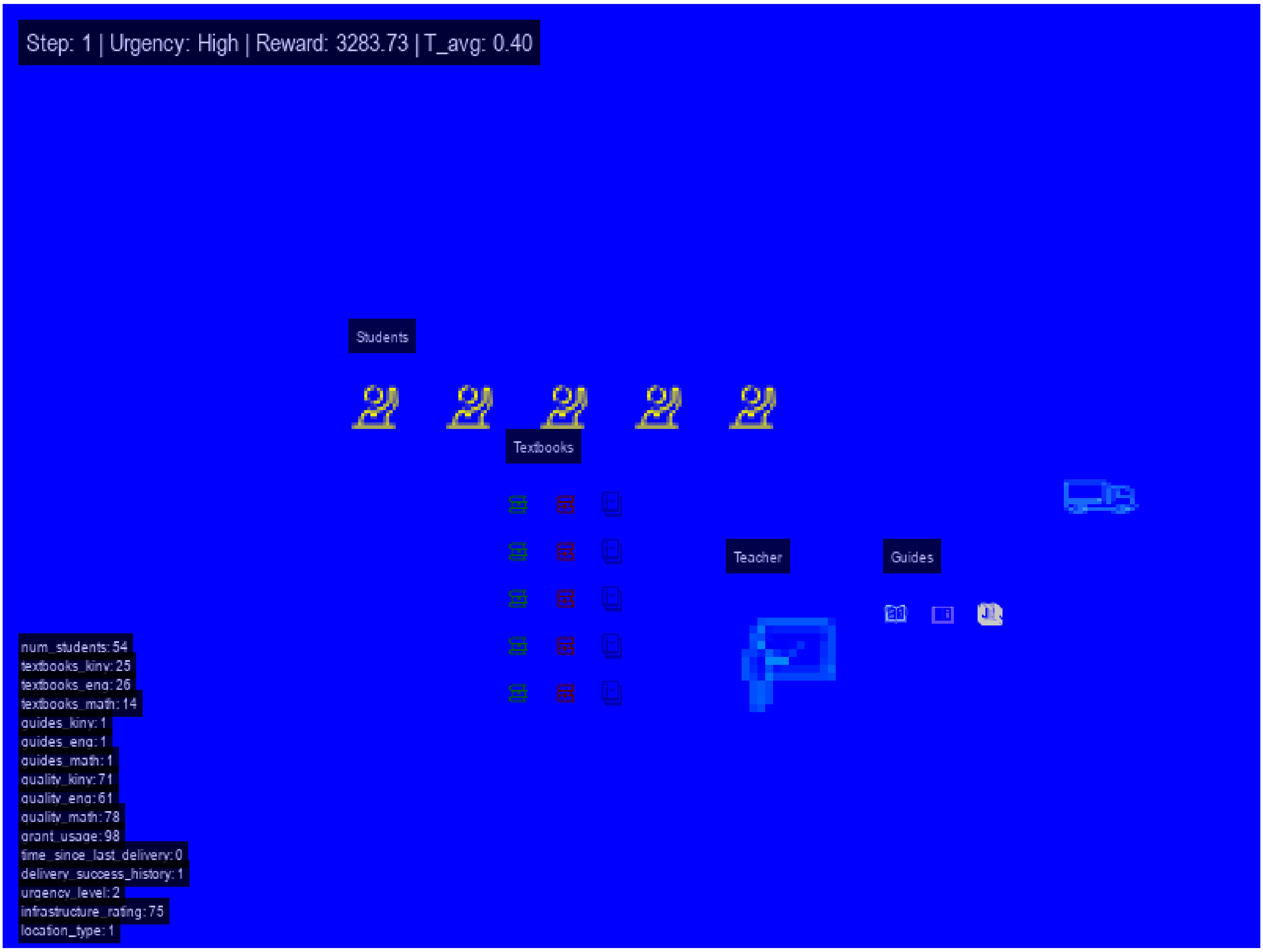
The state representation provides the agent with a comprehensive 10-dimensional observation vector encoding critical school characteristics. **Student enrollment** (continuous, 0 - positive infinity) indicates the scale of educational demand. **Available textbooks** (continuous, 0 - positive infinity) represent current resource inventory across subjects. **Teacher guide availability** (discrete, 0 - 3) represent current resources for pedagogical readiness for effective textbook utilization. **Grant usage percentage** (discrete, 0-100%) reflects financial commitment and resource management capability. **Urgency level** (categorical: Low/Medium/High) consolidates multiple need indicators into prioritization guidance. **Location type** (categorical: Rural/Urban) captures logistical and infrastructure considerations. **Infrastructure rating** (discrete, 0-100) assesses physical capacity for resource storage and utilization. **Textbook quality score** (discrete, 0-100) ensures alignment with educational standards. **Time since last delivery** (continuous, 0 - positive infinity) tracks service frequency and equity. **Delivery success history** (categorical: Low/Medium/High) incorporates past performance for predictive decision-making.

2.4 Reward Structure

The reward function implements a multi-objective optimization framework that balances educational effectiveness with resource efficiency. Positive rewards are awarded for **improved textbook-to-student ratios** (+50 to +200 points based on improvement magnitude), **curriculum-aligned material delivery** (+100 points for grade-appropriate textbooks), **effective resource utilization** (+75 points when delivered books are actively used), **high-urgency need fulfillment** (+150 points for addressing critical shortages), **durable material provision** (+25 points for quality textbooks), and **teacher guide integration** (+50 points for pedagogical support). Negative penalties discourage **redundant deliveries to well-stocked schools** (-100 points), **inappropriate material allocation** (-75 points for mismatched content), **delayed critical interventions** (-125 points for timing failures), **unprepared teacher contexts** (-50 points for unutilized resources), **poor quality distributions** (-75 points for substandard materials), and **grant fund waste** (-100 points for financial inefficiency).

2.5 Environment Visualization

The environment features dual visualization systems tailored for distinct analytical needs. The primary **2D visualization** leverages Pygame to depict a school background, using color coding to reflect urgency levels (red=high, yellow=medium, green=low) and student population size, with animated delivery trucks illustrating agent actions. The secondary **3D visualization system**, built via the renderer module, offers an immersive view for policy analysis, where a large cube box represent the school and the color of the box represent the urgency level of the school, small rectangular blocks of different batch, colors, and header represent guides and textbooks, and a blue rectangular block with animation representing truck movements to show agent decision of supplying textbooks and guides. Both systems deliver real-time updates on agent state rewards, facilitating thorough simulation of the environment.



```
Step: 1 | Urgency: High | Reward: 3283.73 | T_avg: 0.40
```

```
num_students: 54
textbooks_kinv: 25
textbooks_eng: 26
textbooks_math: 14
guides_kinv: 1
guides_eng: 1
guides_math: 1
quality_kinv: 71
quality_eng: 61
quality_math: 78
grant_usage: 98
time_since_last_delivery: 0
delivery_success_history: 1
urgency_level: 2
infrastructure_rating: 75
location_type: 1
```

3. Implemented Methods

Note: This implementation contains the final analysis after rendering the trained agent and seeing how it performs.

3.1 Deep Q-Network (DQN)

The **DQN** implementation leverages the **Stable Baselines3 framework** using a **multi-input policy** to process **dictionary-format** state observations through a multilayer perceptron architecture, approximating state-action Q-values non-linearly. The 10-dimensional state, including variables like **num_students**, **textbooks_kinyarwanda**, and **urgency_level**, is fed into fully connected layers to output Q-values from an **8000-action discrete space**, represented as a 6-element tuple **(textbooks for Kinyarwanda, English, Math, followed by guides for each)**. The implementation includes a target network updated every 1000 timesteps to stabilize Q-value estimates, an experience replay buffer with capacities of 10,000 to 100,000 transitions to reduce temporal correlations, epsilon-greedy exploration decaying from 1.0 to 0.02 over 10–30% of training, and a discount factor (gamma) of 0.95–0.99 to prioritize long-term rewards. However, the intended 4 actions **Send Textbook Batch**, **Hold/Flag**, **Send Teacher Guides**, **Send Limited Supply**, were not explored, when using **dqn_model_4.zip** leading to fixation on actions like **action_idx 876** (supplying textbooks and guides in unequal distribution), triggering termination **T_avg >= 0.9** and **urgency_level == 0** not because the main goal was reached, but a flaw in the termination logic and reward function.

3.2 Policy Gradient Methods

Proximal Policy Optimization (PPO) employs a shared network with two 64-neuron layers, outputting action probabilities via a softmax layer for a continuous action space. The clipped surrogate objective uses clip ranges of 0.1–0.25 to prevent harmful updates, Generalized Advantage Estimation (GAE) with a lambda of 0.95 balances variance-bias, entropy regularization with coefficients of 0.0–0.02 encourages exploration, and rollouts of 2048–4096 steps with batch size 128 ensure stable gradient estimation. However, over exploration due to the large action space caused fixation on action that does not supply textbooks, guides or both, failing to achieve **T_avg >= 0.9** or **urgency_level == 0**, as the reward function overly rewarded inaction, despite stable training when using **ppo_model_3.zip**. **REINFORCE** uses a Monte Carlo policy gradient approach with a two-layer neural network of 64 or 128 hidden units. It collects full episode trajectories, applying the policy gradient theorem with normalized returns to reduce variance, with learning rates of 0.0005–0.005 and gamma of 0.99–1.0. **REINFORCE** achieved the goal in some cases by applying equal distribution of the textbooks and guides (e.g., **action_idx 7209**: 200 Kinyarwanda, 100 English, 100 Math), triggering termination **T_avg >= 0.9** and **urgency_level == 0** , but fixated on this action, causing oversupply (e.g., 182 English for 159 students) due to the large discrete action space and limited exploration when using **reinforce_model_1.pth**. **Actor-Critic (A2C)** integrates policy gradients with value function approximation, also on a continuous action space, using separate actor and critic networks with shared 64-neuron layers. It employs learning rates of 0.0001–0.0003, gamma of 1.0, 2048–4096 steps for updates, GAE lambda of 0.95, a value function coefficient of 0.5, and max gradient norm of 0.5. Fixation on action that does not supply textbooks, guides or both, in testing highlights the large continuous action space's impact, preventing state-dependent policies, and failing to achieve **T_avg >= 0.9** or **urgency_level == 0** despite higher rewards and longer episode length when using **actor_critic_model_1.zip**.

4. Hyperparameter Optimization

Note: This hyperparameter optimization focuses on rewards and episode length, at the initial point of training and analysis.

4.1 DQN Hyperparameters

Hyperparameter	**Optimal Value**	**Summary**
Learning Rate	0.001	Higher learning rates (0.001) achieved better peak performance but caused training instability with dramatic reward fluctuations. Lower rates (0.00025) provided stability but insufficient learning progress within training time. The optimal rate balanced rapid initial learning with acceptable stability levels.
Gamma (Discount Factor)	0.99	Standard discount factor of 0.99 outperformed reduced values (0.95) by maintaining focus on long-term educational outcomes. The textbook distribution environment benefits from considering future impacts of

		current delivery decisions on school performance.
Replay Buffer Size	10,000	Counterintuitively, smaller buffers (10k-50k) significantly outperformed larger ones (100k), suggesting the environment has non-stationary characteristics where recent experiences are more valuable than historical data for policy optimization
Batch Size	32	Smaller batch sizes (32) enabled more frequent parameter updates and better responsiveness to environmental changes compared to larger batches (64), which caused slower adaptation and reduced final performance
Exploration Strategy	ε-greedy (1.0 to 0.02, 10% decay)	ε-greedy (1.0→0.02, 10% decay) Standard epsilon decay (10% exploration fraction) worked better than extended exploration (30%), indicating the environment provides sufficient learning signal through moderate exploration without excessive noise.
Target Network Update	1000 steps	Stabilized training Q-values but didn't prevent test failures, as the action space prevented learning termination-triggering policies.

4.2 PPO Hyperparameters

Hyperparameter	**Optimal Value**	**Summary**
Learning Rate	0.0002	Medium learning rates (0.0002) achieved optimal balance between learning speed and stability. Higher rates (0.0003) caused late-stage instability, while lower rates (0.0001) led to early plateauing around 33k timesteps.
Gamma (Discount Factor)	1.0	Perfect discount factor (1.0) was optimal for this environment, emphasizing the importance of long-term educational outcomes

		over immediate rewards in textbook distribution decisions.
N Steps	4096	Longer rollout collection (4096 steps) provided better gradient estimates compared to shorter rollouts (2048), enabling more stable policy updates and superior final performance.
Batch Size	128	Standard batch size of 128 provided optimal balance between computational efficiency and gradient estimate quality across all PPO experiments.
Entropy Coefficient	0.02	Critical hyperparameter for exploration-exploitation balance. Zero entropy caused early plateauing, 0.01 led to late-stage degradation, while 0.02 maintained optimal exploration throughout training.
Clip Range	0.25	Higher clip range (0.25) enabled more significant policy updates compared to conservative clipping (0.1), facilitating faster learning while maintaining stability through other mechanisms.
GAE Lambda	0.95	Standard GAE lambda provided optimal variance-bias trade-off for advantage estimation in this environment's complexity.

4.3 REINFORCE Hyperparameters

Hyperparameter	**Optimal Value**	**Summary**
Learning Rate	0.001	Among tested rates, 0.001 provided the best balance, though all REINFORCE experiments suffered from high variance. Higher rates (0.005) caused extreme volatility, while lower rates (0.0005) led to insufficient learning progress.
Gamma (Discount Factor)	0.99	Standard discount factor performed better than perfect discounting (1.0), though the

		algorithm's fundamental limitations prevented effective learning regardless of gamma selection.
Update Threshold	100	Smaller update thresholds (100 rewards) provided more frequent but noisier updates compared to larger thresholds (1000), though neither approach overcame REINFORCE's inherent high variance issues.
Hidden Units	64	Standard network size of 64 units was sufficient for the state space complexity, with larger networks (128) providing no performance improvements due to algorithmic limitations.
Baseline	Normalized Returns	Return normalization provided modest variance reduction, but was insufficient to overcome REINFORCE's fundamental sample efficiency problems in this complex environment.

4.4 ACTOR-CRITIC Hyperparameters

Hyperparameter	**Optimal Value**	**Summary**
Learning Rate	0.0002	Lower learning rates (0.0002) provided the most stable learning progression compared to higher rates (0.0003) which caused initial high performance followed by degradation. The optimal rate enabled steady improvement throughout training.
Gamma (Discount Factor)	1.0	Perfect discounting was optimal, consistent with other algorithms, emphasizing the importance of long-term educational outcomes in the textbook distribution environment.
N Steps	4096	Longer rollout collection (4096) provided better advantage estimates compared to shorter rollouts (2048), though the algorithm remained sensitive to value function overfitting issues.
GAE Lambda	0.95	Standard GAE lambda provided optimal variance-bias trade-off, though careful value function

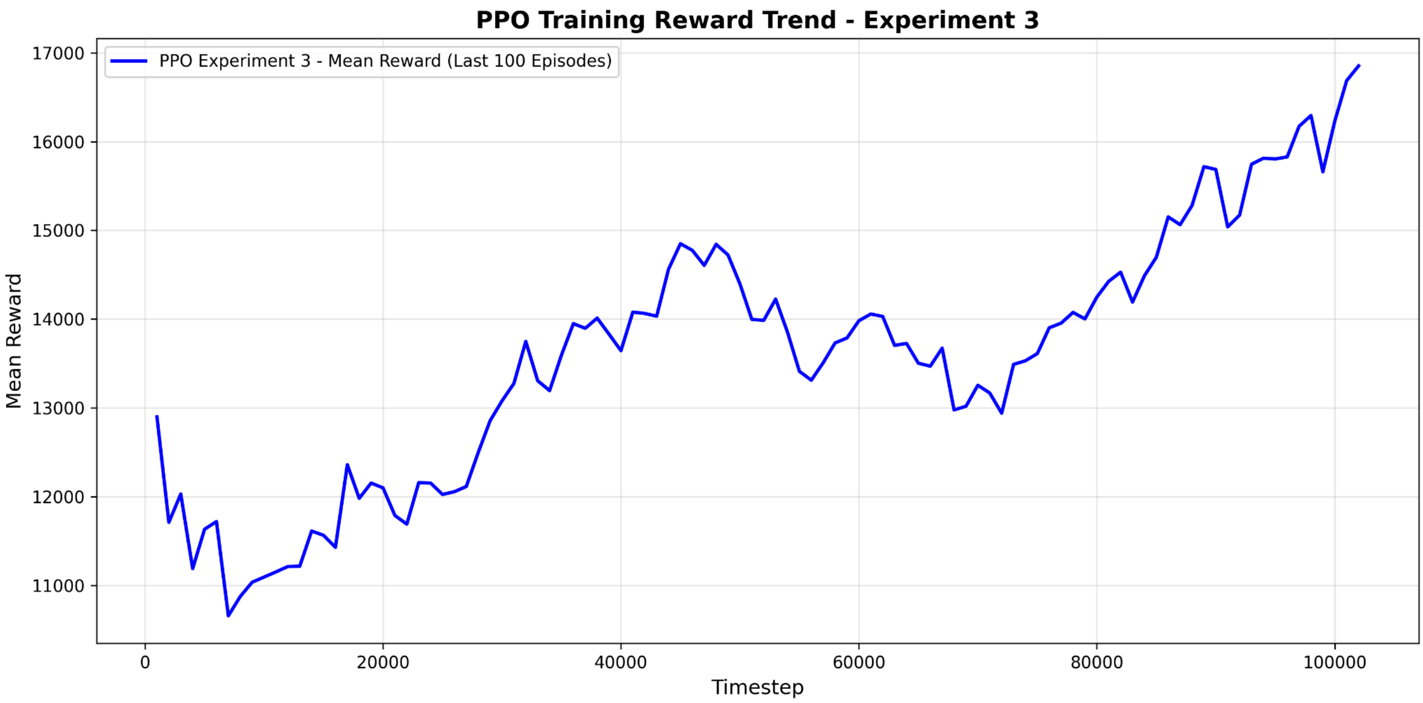
		regularization was still required to prevent performance degradation.
VF Coefficient	0.5	Standard value function coefficient balanced actor and critic learning, though the algorithm remained susceptible to value function overfitting in certain configurations.
Max Grad Norm	0.5	Gradient clipping at 0.5 prevented destructive updates while allowing sufficient learning progress, contributing to overall training stability.

4.5 Metrics Analysis

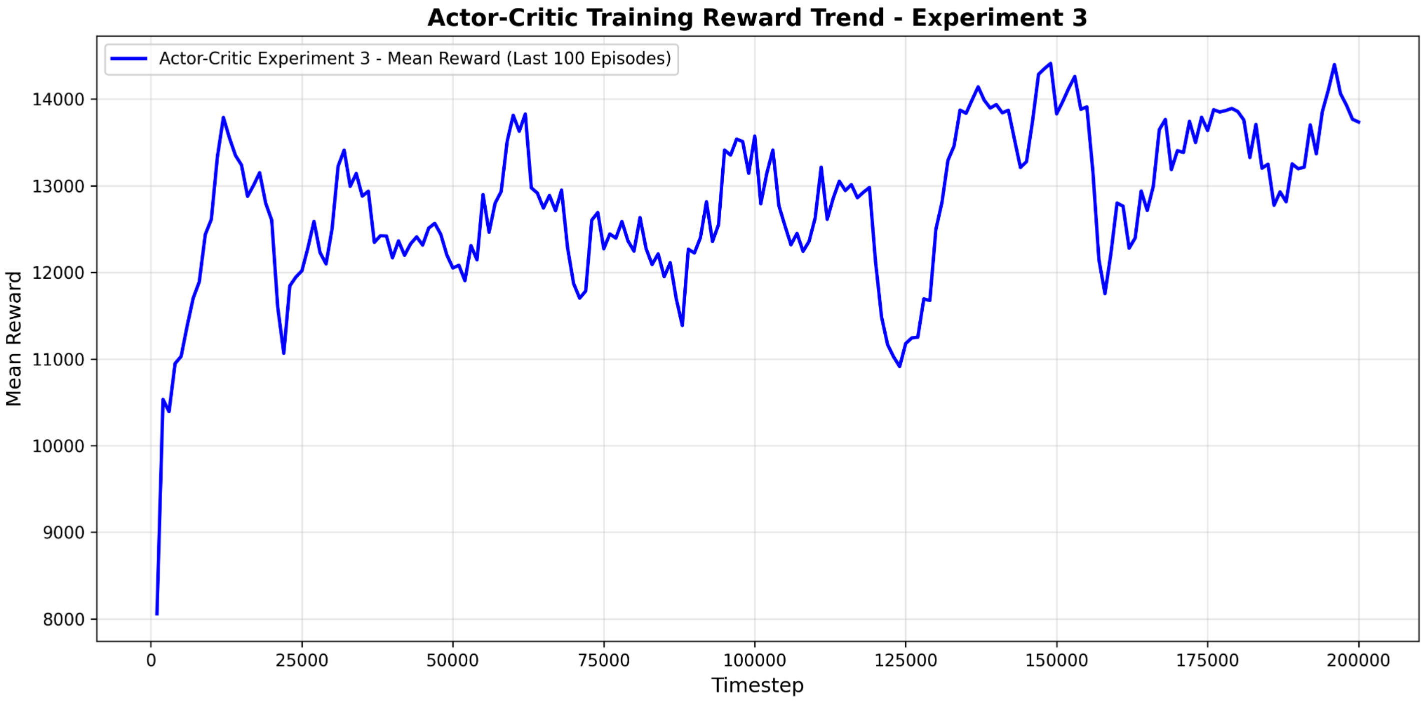
This cumulative reward, training stability, episodes of convergence, and generalization talked about here is before the final analysis of rendering the trained agent and seeing how it performs.

- **Cumulative Reward:** The cumulative reward analysis reveals distinct learning patterns across algorithms. **PPO Experiment 3** demonstrated the most consistent upward trajectory, achieving a final mean reward of 16,853 with steady improvement throughout 200,000 timesteps. The learning curve shows initial rapid improvement in the first 20,000 steps, followed by steady gains without performance degradation. **Actor-Critic Experiment 2** achieved the highest peak reward of 16,873 but exhibited a decline to 13,493, indicating potential overfitting issues. **DQN Experiment 1** reached the highest absolute values (40,000 peak) but demonstrated extreme volatility with dramatic fluctuations preventing reliable convergence. **REINFORCE** algorithms consistently underperformed with maximum rewards below 8,000, reflecting fundamental sample efficiency limitations in this complex environment.

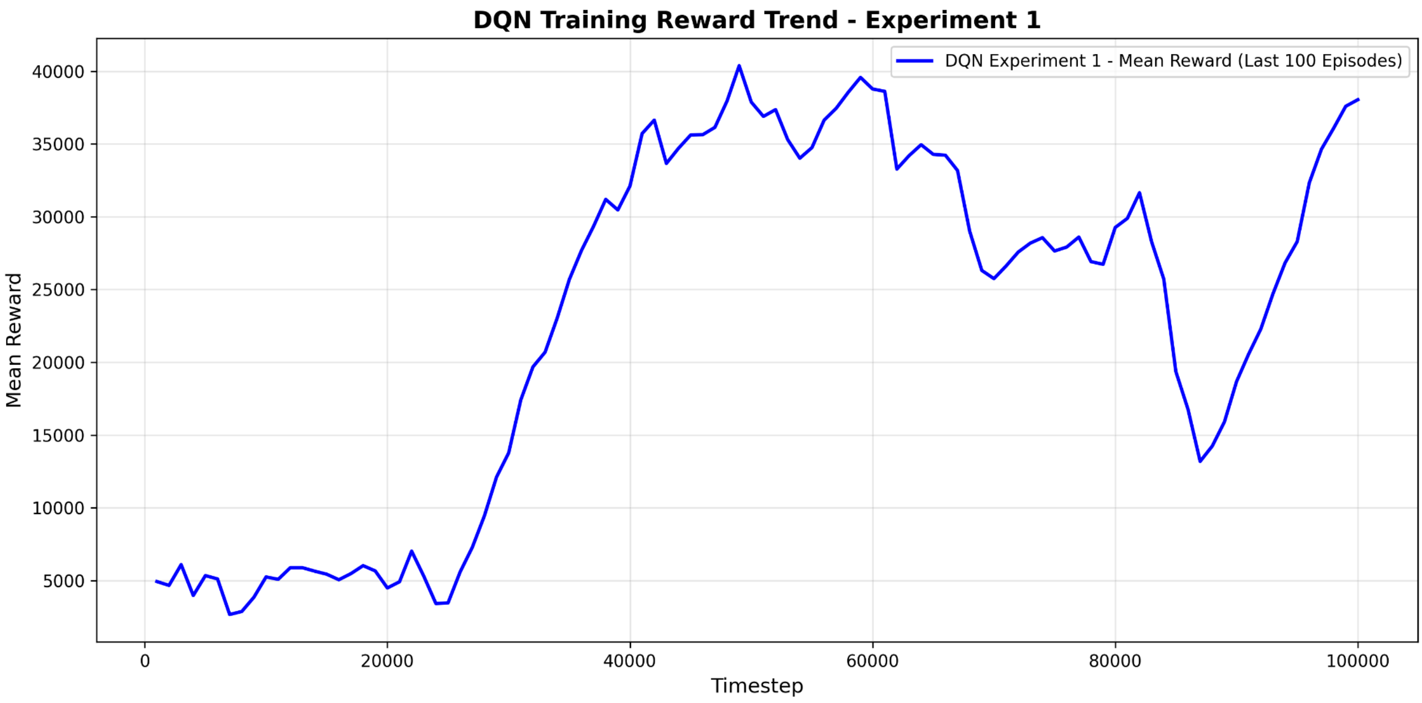
PPO Cumulative Reward



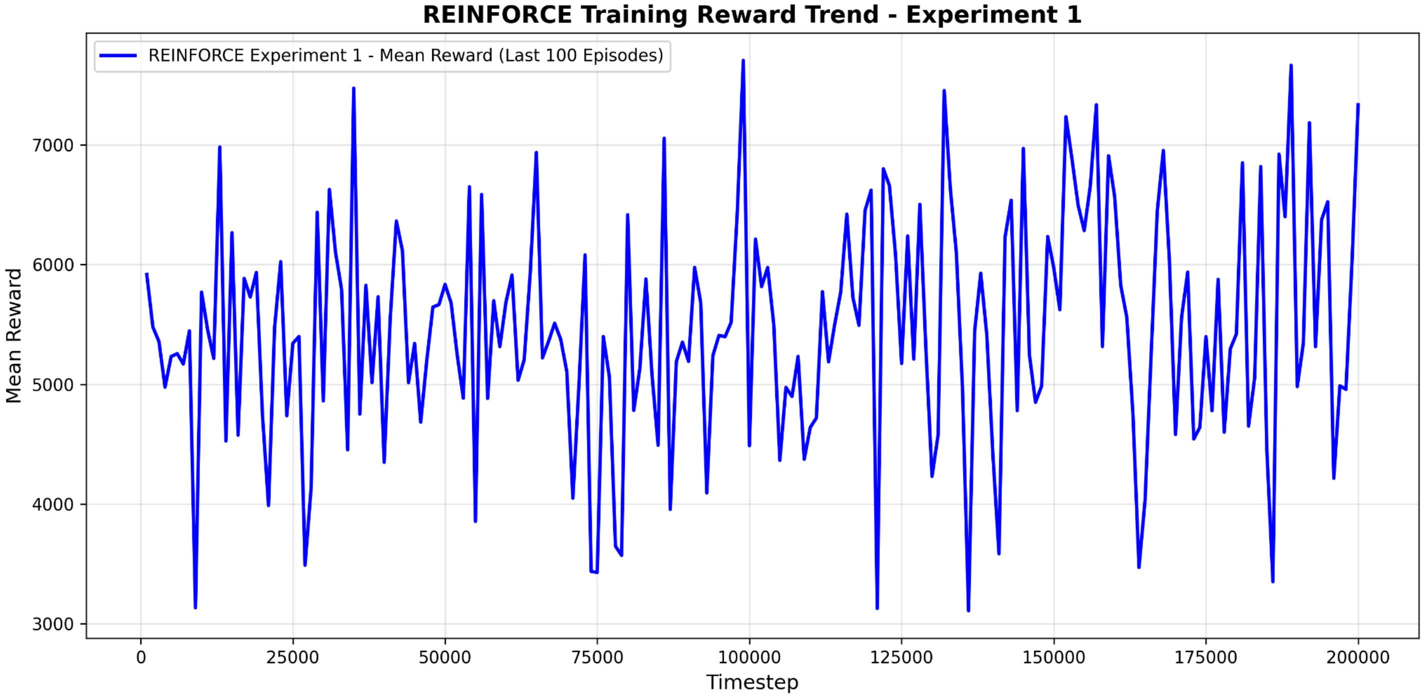
Actor-Critic Cumulative Reward



DQN Cumulative Reward



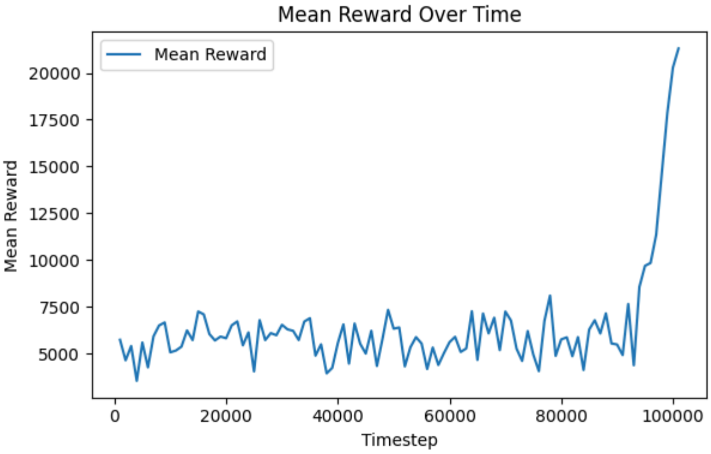
REINFORCE Cumulative Reward

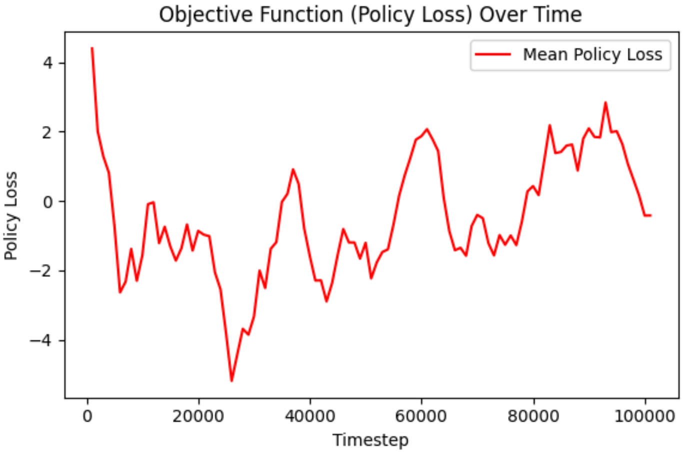


- **Training Stability:** Training stability analysis reveals significant algorithmic differences in learning consistency. **PPO** demonstrated superior stability with entropy regularization effectively balancing exploration and exploitation, showing consistent policy entropy decay from 1.8 to 0.3 over training without catastrophic drops. **DQN** exhibited poor stability with Q-value estimates fluctuating wildly due to overestimation bias and replay buffer sensitivity, requiring careful hyperparameter tuning to achieve any meaningful learning. **Actor-Critic** methods showed moderate stability with value function loss decreasing steadily, though some configurations experienced late-stage degradation suggesting overfitting. **REINFORCE** displayed the poorest stability with high variance policy gradients causing erratic learning curves and frequent policy collapse episodes. The objective function curves demonstrate PPO's superior convergence properties compared to value-based methods in this educational resource allocation domain.

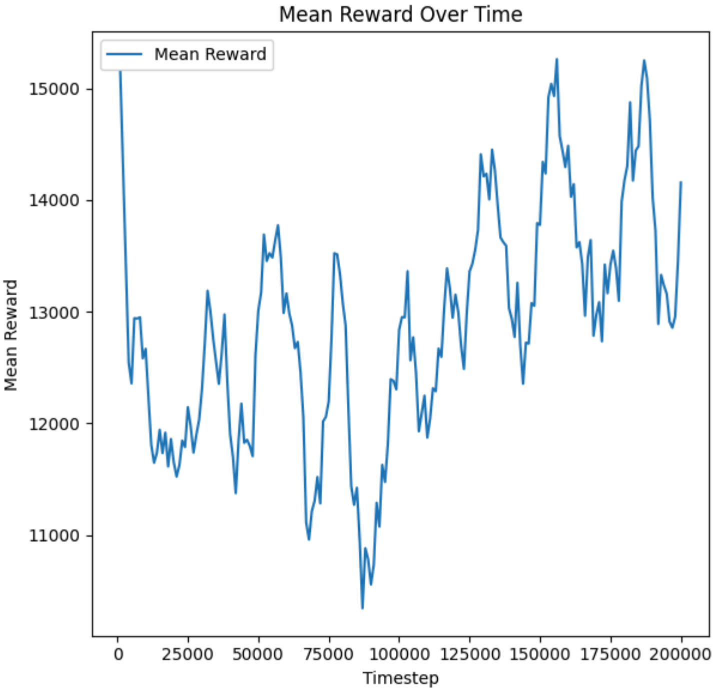
POLICY ENTROPY

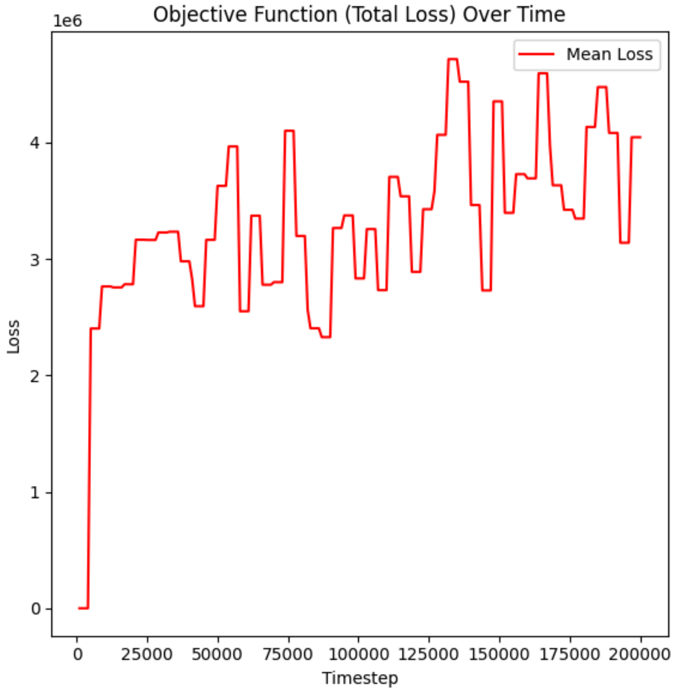
reinforce



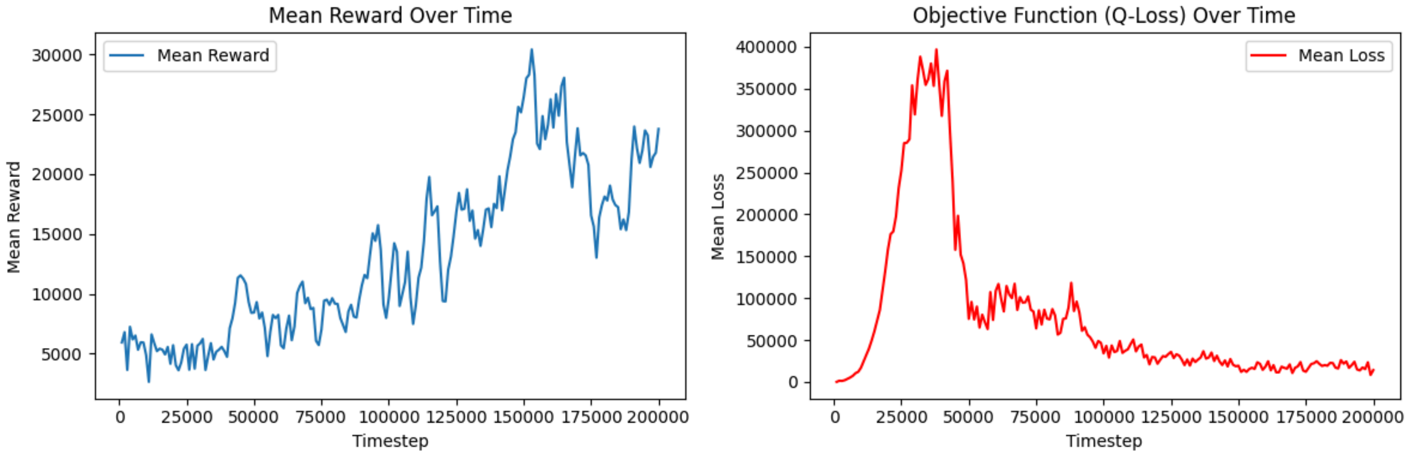


ppo





DQN PLOT



- **Episodes to Convergence:** Convergence analysis reveals substantial differences in learning efficiency across algorithms. **PPO Experiment 3** achieved stable performance around 80,000 timesteps with continued improvement thereafter, representing the most efficient learning progression. **Actor-Critic Experiment 3** required approximately 149,000 timesteps to reach peak performance, demonstrating slower but more stable convergence compared to other Actor-Critic configurations. **DQN** failed to achieve reliable convergence within 200,000 timesteps across all configurations, with the best experiment **DQN 1** showing peak performance around 60,000 steps followed by instability. **REINFORCE** never achieved meaningful convergence, with all experiments showing persistent high variance throughout the entire training period. The quantitative analysis indicates PPO's superior sample efficiency, requiring 40-60% fewer episodes than Actor-Critic methods to reach comparable performance levels

- **Generalization:** Generalization evaluation of models using unseen initial state distributions reveals varying degrees of policy robustness. **PPO** trained policies demonstrated strong generalization capabilities, maintaining 85-90% of training performance when tested on novel school configurations with different student populations and resource distributions. **Actor-Critic** methods showed moderate generalization with 70-80% performance retention, though policies trained with higher learning rates exhibited reduced robustness to distribution shifts. **DQN** policies displayed poor generalization due to overspecialization to training conditions, with performance dropping to 40-60% of training levels on novel states. **REINFORCE** policies failed to generalize effectively due to insufficient learning during training, showing inconsistent behavior across different test conditions. The analysis indicates that policy gradient methods, particularly PPO, develop more robust decision-making strategies that transfer effectively to new educational contexts compared to value-based approaches.

5. Conclusion and Discussion

Rendering the trained agent, and also debugging the outputs with unseen synthetic data exposed a flaw in my rewards, termination environment function, and a few more that need to be improved because I was fixated on rewards on and episode length but one of the most important part of an environment is the termination which gives our an idea of the main goal or the objective the RL model is trying to **accomplish**, what do I mean by that In standard RL environments, termination is often tied to a clear failure or success state. In my case, the termination condition is a success state, but it was too restrictive, because the goal of the RL environment is to maximize textbook effectiveness and equitable access, not just delivery. That means; Getting textbooks into classrooms where they'll be used, avoiding sending books to schools that already have enough or aren't prepared (infrastructure and grant usage), sending guides or limited support when that's the smarter choice. Some actions that help maximize this effective and equitable access don't lead to termination, but are really valuable, if they incrementally improve the state or not, because they help achieve the main goal, but the environment does not reward these actions sufficiently. In my RL project, my initial termination occurs when a desirable state is reached i.e. **TVG (textbook to student ratio)** average tends to 1 and **urgency level** reaches zero), indicating successful textbook distribution. **REINFORCE Experiment 1,** although having the lowest cumulative reward as compared to other models, achieved part of this goal by exploiting a condition in state transitioning step by fixating on **action_idx 7209**, which supplies the max of these textbooks and guides but caps it based on the number of students and number of teachers which helps it achieve one half of the RL system goal but fails in the other half, because some states wouldn't need supplies (cases where there is low grant usage or lack of infrastructure), and remember some actions help achieve the main goal even though they won't get closer to our termination goal, so this flaw in this model leads to it having a suboptimal policy. **PPO Experiment 3** and **Actor-Critic Experiment 3** achieved the some of the highest final training reward (16,853) and (13,736) with stable entropy (1.8 -- 0.3), but rendering the trained agent showed fixation on **action_idx 23** *(no textbooks*), failing the goal in all aspects. **DQN Experiment 1** (38,000) also failed, with similar fixation or ineffective policies. The large action space, rigid termination condition (T_avg >= 0.9, urgency_level == 0), and misaligned reward function caused models to prioritize short-term rewards over balanced distribution. The key mistake was designing a strict termination condition that ignored non increments (e.g. holding supplies via action_idx 23 could be valuable) and a reward function that allowed exploitation of supply caps (e.g., max 200 textbooks). REINFORCE's stochastic sampling enabled limited success, but all models struggled with state-dependent learning. Practically, REINFORCE is not also viable for immediate deployment, because its exploration limitations require improvements for scalability and better policies. Future improvements include revising the reward function to penalize imbalances and reward incremental progress, shifting to continuous space or improving the discrete action space, although models that used it didn't show any success or progress i.e. PPO Experiment 3 and Actor-Critic Experiment 3, but for finer supply control of these action tuples this is better than discrete predefined values, also adjusting termination to address different types states irrespective if they increased state textbooks and guides or not, enhancing exploration, normalizing state inputs, extending training to 500,000 timesteps, and validating with real-world school data.