Reward Shaping in RL: PPO vs. MORL in a Penguin Simulation

Benyamain Yacoob, Xinyang Zhang, Jamuna Narayanaswamy, Caitlin Snyder

Department of Electrical & Computer Engineering & Computer Science
University of Detroit Mercy
Detroit, MI, United States
{yacoobby, zhangxi24, jsugatu, snydercr}@udmercy.edu

***Abstract*—This paper presents a reinforcement learning project focused on training a penguin agent within a Unity environment using the ML-Agents toolkit to feed its baby penguin under multiple constraints. We compare two implementations: a simple Proximal Policy Optimization (PPO) approach with a single-task objective and a multi-objective reinforcement learning (MORL) setup incorporating time constraints and energy conservation through movement penalties. The simple PPO agent receives basic rewards for eating fish and feeding the baby, while the MORL agent features enhanced rewards for feeding and small penalties for actions to promote efficiency. Using TensorBoard logs and side-by-side video comparisons, we analyze agent behavior at each timestep to understand decision-making processes. Results indicate that the MORL agent completes episodes faster with more deliberate actions, avoiding the wandering behavior observed in the simple PPO agent. This improvement stems from a refined reward structure that balances primary goals with secondary objectives, demonstrating the critical role of reward shaping in agent performance. Our findings highlight how multi-objective frameworks can optimize complex tasks in simulated environments, providing insights into effective RL design.**

***Index Terms*—reinforcement learning, proximal policy optimization, multi-objective learning, penguin simulation, Unity ML-Agents**

I. Introduction

Reinforcement learning (RL) trains agents to make decisions by interacting with an environment to maximize cumulative rewards. In this project, we apply RL to train a penguin agent in a Unity environment using the ML-Agents toolkit. The primary objective is for the penguin to feed its baby penguin, operating under time constraints and energy conservation goals. Initially, we considered additional objectives such as energy preservation through a health system and potentially defending against a critic agent, though these were prioritized based on feasibility. Our aim is to compare two approaches: a simple Proximal Policy Optimization (PPO) implementation focused on a single task, and a MORL-inspired framework that adapts multi-objective principles to integrate additional constraints. This comparison seeks to understand how reward structures influence agent behavior and efficiency in simulated tasks. By analyzing the differences in performance, we aim to gain insights into designing effective RL systems for complex, multi-faceted objectives.

II. Development Process

The project began with setting up a simulation environment in Unity, leveraging the ML-Agents toolkit to create a gymnasium-like space for RL training. The environment was designed with random positioning of the penguin, its baby, and four fish to create varied training scenarios per episode, while fish were programmed to swim dynamically towards random targets with speeds randomized between 50% and 150% of a base value, increasing the challenge of capturing them. Initially, we implemented a simple PPO agent with a basic reward structure: +1 for eating fish and +1 for feeding the baby, alongside a small negative reward per step to encourage speed. Observing that this agent often wandered, we developed a MORL-inspired agent with a refined reward system: +1 for eating fish, +3 for feeding the baby to prioritize the primary goal, and small penalties (-0.0002) for moving forward or turning to promote energy efficiency. Training involved approximately 1 million steps for each implementation, configured with a batch size of $B = 128$, buffer size of $N = 2048$, and learning rate of $\alpha = 3 \times 10^{-4}$, using TensorBoard logs to track metrics like episode length, policy entropy, loss, and penalties, and side-by-side video comparisons to assess behavioral differences at specific timesteps [1], [2]. This iterative process refined the agent's decision-making for optimal performance, revealing why certain actions were taken under varying conditions (see Figure 1 for episode length comparison). The source code for this implementation is available online [3].

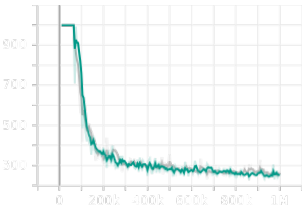

Fig. 1. Comparison of mean episode length between simple PPO and MORL agents over training, highlighting efficiency differences.

III. Training Dynamics

To understand the behavioral differences between the simple PPO and MORL-inspired agents over the 1 million training steps, we analyzed various metrics through TensorBoard logs, focusing on timestep-specific trends. Early in training (0-200k steps), the MORL agent exhibited high training loss (around 0.009), indicating significant exploration as it adapted to multiple objectives like feeding and energy conservation. By 800k-1M steps, this loss decreased to near 0.001, suggesting convergence to a stable, optimized policy. In contrast, the PPO agent showed a steadier but less refined loss reduction, often stabilizing at a higher value, reflecting its single-task focus. Cumulative penalties for the MORL agent dropped from 900 to 100-200 over training, with fluctuations indicating ongoing trade-offs between movement and efficiency, while the PPO agent maintained higher penalties due to less emphasis on energy conservation (see Figure 2 for reward trends). These dynamics highlight why the MORL agent developed more deliberate actions, adapting to complex trade-offs over time, whereas the PPO agent prioritized feeding with less regard for efficiency.

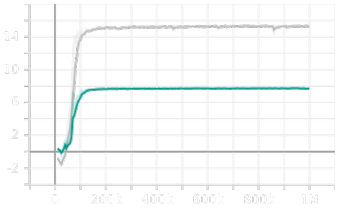

Fig. 2. Comparison of cumulative reward trends between simple PPO and MORL agents, showing efficiency in reward accumulation.

IV. Challenges Faced and Solutions Implemented

Several challenges emerged during development. Early on, achieving compatibility of the Unity environment across team machines posed logistical hurdles, requiring standardized configurations to enable collaboration. Algorithmically, the agent often became stuck in local minima, particularly in the MORL setup, where penalties for movement discouraged exploration, leading to stagnant behavior. Initial harsh penalties resulted in the agent avoiding actions altogether, accumulating negative rewards. To address this, we fine-tuned the penalty values to be less severe and adjusted hyperparameters to balance exploration with efficiency. These modifications, validated through iterative testing and analysis of training logs, resulted in more deliberate actions, as the MORL agent learned to prioritize feeding while minimizing unnecessary movements. This adjustment process was critical to overcoming initial setbacks and achieving the desired behavioral outcomes.

V. Conclusion

This project demonstrates that reward shaping significantly impacts agent performance in reinforcement learning tasks. The MORL implementation, with its emphasis on balancing multiple objectives like time efficiency and energy conservation, outperformed the simple PPO approach by completing episodes faster (approximately 248 steps versus 253 steps per episode) and exhibiting more focused behavior. The MORL agent also showed lower policy entropy (0.549 vs. 0.562), indicating more deterministic actions due to clearer goals and efficiency incentives (see Figure 3). Further, the MORL agent faced greater learning complexity, evidenced by a higher value loss (0.4058 vs. 0.1308) compared to PPO, reflecting the challenge of predicting state values in multi-objective trajectories (see Figure 4). Timestep-specific analyses of metrics such as loss and penalties revealed why behaviors diverged, with the MORL agent adapting to trade-offs over time. These findings highlight the importance of carefully designed reward structures over mere hyperparameter tuning, showing that clear goals and efficiency incentives drive superior outcomes, though they may require more tuning to balance complexity and performance. Looking ahead, integrating visual perception through camera and Lidar inputs could enhance semantic understanding for future tasks, building on this foundation for further exploration of multi-objective frameworks in complex simulated environments.

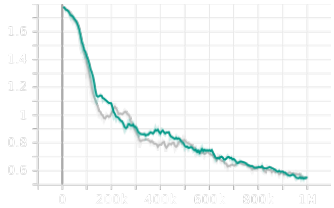

Fig. 3. Comparison of policy entropy between simple PPO and MORL agents, showing the MORL agent's more deterministic behavior.

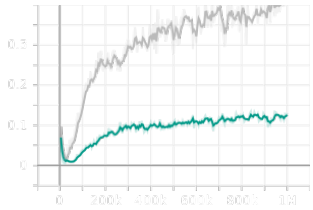

Fig. 4. Comparison of value loss between simple PPO and MORL agents, indicating higher learning complexity for MORL.

References

[1] Benyamain, "Simple penguin ppo." https://drive.google.com/file/d/15gYHr9aKlZCbjgRDDBc9PP3bAxlcj1CQ/view?usp=sharing, 2025.

[2] Benyamain, "Morl-inspired penguin ppo." https://drive.google.com/file/d/1UJpOPu3WH7DWjwwAOx4SwG7ptN0wvTAU/view?usp=sharing, 2025.

[3] Benyamain, "Penguins implementation." https://github.com/Benyamain/Penguins, 2025.