

# Enhancing Sample Efficiency and Stability in BipedalWalker-v2 Using an Improved Actor-Critic Algorithm

## 2. Short Summary

This project is all about making the Actor-Critic algorithm work faster and more smoothly in the BipedalWalker-v2 environment from OpenAI Gym. The main question here is: how can I tweak the Actor-Critic algorithm to speed up learning and make it more stable in this environment? It's an interesting challenge because BipedalWalker requires controlling a robot to walk across uneven terrain, which is a tough problem with lots of variables in both the state and action spaces. If I can improve sample efficiency and stability, it could make reinforcement learning more practical for real-world robotics.

## 3. Objectives

- Implement a baseline Actor-Critic algorithm for the BipedalWalker-v2 environment using PyTorch.
- Enhance the algorithm by incorporating by following this improvement to Actor-Critic <https://arxiv.org/pdf/1801.01290>

## 4. Methodology

Baseline Algorithm: Implement a standard Actor-Critic algorithm with:

- A neural network for the actor (policy) that outputs mean and standard deviation for continuous actions.
- A neural network for the critic (value function) that estimates the state value.

Enhancements:

- Prioritized Experience Replay: Use a replay buffer that prioritizes transitions with higher TD error.
- Advantage Normalization: Normalize advantages to reduce variance in policy updates.
- Adaptive Exploration: Use entropy regularization to encourage exploration early in training and reduce it as the policy improves.
- Training: Train the agent in the BipedalWalker-v2 environment and monitor cumulative rewards and training stability.

## 5. Evaluation

Qualitative Results:

Learning curves showing cumulative rewards vs. episodes for the baseline and enhanced algorithms.

Visualization of the bipedal walker's behavior during training

Quantitative Results:

Measure the final performance (average reward over the last 100 episodes).

Analyze training stability across 5 independent runs

## 6. Environment

Primary Environment: BipedalWalker-v2 from OpenAI Gym pytorch and Gym

## 7. References

[https://www.gymnasium.dev/environments/box2d/bipedal\\_walker/](https://www.gymnasium.dev/environments/box2d/bipedal_walker/)

Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.

Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine

<https://arxiv.org/abs/1801.01290>