Learning to Walk Efficiently

Joseph Dunne

Abstract

This paper proposes using Loss Adjusted Approximate Actor Prioritised Experience Replay (LA3P) with SAC to solve the BipedalWalkerV3 environment. In the basic version we find our approach to be sample efficient, with the first reward surpassing 300 occurring within 105 episodes. The majority of the episodes following sustain or improve upon this, with a max reward obtained of 333. For the hardcore version, we find that this approach potentially exceeds the current top performing model in sample efficiency. Our model achieves its first reward over 300 in 568 episodes compared to the top model's 900. In both environments, it produces agents that demonstrate significant fluency, speed and control.

1 Methodology

1.1 On Policy vs Off Policy

Due to our focus on sample efficiency, using off-policy algorithms was desirable, benefiting through learning from previous policies' experiences. Conversely, on-policy approaches would be limited to only the current policies experience. These experiences would likely break the assumption of independent and identically distributed (IID) data in our environment, with immediate future states highly correlating with current ones. Breaking this assumption would lead to biased gradients, diverging from the true policy-reward gradient. To improve independence, a larger number of experiences could be collected before taking a gradient step. However, doing so would reduce sample efficiency.

Another shortcoming of on-policy approaches is that rare experiences are easily forgotten. A policy has only one update to capitalise on a rare experience before any memory of it is discarded. This could be particularly problematic for the Hardcore version of the environment, where knowledge of how to approach rarely occurring obstacles, could be beneficial for future policies to learn from.

For these reasons, we limited our search of possible algorithms to off-policy.

1.2 Soft Actor Critic (SAC)

Off-policy learning is not directly compatible with most policy-based methods. Q-learning based methods are compatible, although they have poor performance with large observation spaces and continuous actions [8]; the Bipedal Walker environment contains both such features.

Soft Actor Critic (SAC), was introduced in 2018, and has been successfully applied to both large observation spaces and continuous control [5]. Unlike other actor-critic methods, SAC includes an entropy term in its policies objective function (see Equation 1). Consequently, agents both maximise expected return but also entropy. α controls the contribution of entropy to the objective function and influences an agent's exploration. When set too high, SAC reduces to a random policy. When set too low $\alpha \to 0$ and it reduces to a greedy RL algorithm that quickly converges.

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim p_\pi} \left[r'(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t)) \right] \tag{1}$$

α is a brittle hyperparameter for SAC, and is dependant on the environment. The authors addressed this by introducing automatic entropy tuning, finding that it significantly reduced the need for hyperparameter tuning per environment [4]. Furthermore, with this addition their results on continuous control were superior to that of other SOTA methods, including TD3, both in performance and sample efficiency.

Due to both its sample efficiency and suitability for continuous control, we decided to use SAC to apply to our environment.

1.3 Prioritised Experience Replay (PER)

The performance of off-policy learning can be significantly improved through prioritisation of experiences to select. Schaul et al. introduced Prioritised Experience Replay (PER), where experiences with higher importance are selected with greater probability [13]. They use TD error as a proxy for measuring how "surprising" certain transitions are, indicating the potential for the agent to learn something new. From their paper, they demonstrate its superiority for sample efficiency when compared to random sampling of experiences.

1.4 Loss Adjusted Approximate Actor Prioritised Experience Replay (LA3P)

Recent work has empirically shown PER to reduce the performance of actor-critic algorithms for continuous control [3] [9]. Saglam et al. proved this result theoretically, showing the estimated gradient from PER to diverge from the true gradient [12]. The authors propose an alternative, Loss Adjusted Approximate Actor Prioritsed Experience Replay (LA3P), demonstrating its effectiveness across a battery of continuous control environments. Figure 1 elucidates LA3P. Before training the actor and critic, the batch is split accordingly:

A hyperparameter $\lambda \in [0, 1)$ is used to indicate the proportion of the total batch, which is obtained through random sampling. This is used to train both the actor and critic.

The remaining $1 - \lambda$ is used to train the critic and actor according to prioritised sampling and inverse prioritised sampling respectively. Prioritised sampling, is defined as prioritising according to large TD error. Conversely, inverse prioritised sampling is defined as prioritising according to a small TD error.

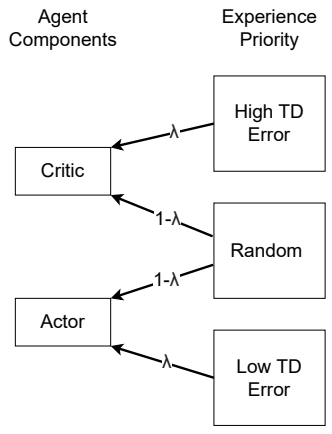

Figure 1: Sample collections for the actor and critic using the three methods of random sampling, high TD error and low TD error. The arrows indicate what method is used, and the proportion of the total batch which is allocated to the sampling technique.

The intuition for this design is that a teacher (critic), cannot teach a student (actor) well if they do not understand the subject well themself. A low TD error indicates that the critic has a better understanding, and thus is used for training the actor. On the other hand, a high TD error is more useful for training the critic since it implies there is more to learn.

Instability can arise from training actors and critics on entirely differing experiences [7]. Since high and low TD error are opposites, samples collected by prioritising them will be mutually exclusive. To mitigate this, the authors of LA3P propose also training the actor and critic on a shared set of λ random experiences.

We therefore apply LA3P in this paper due to its empirical success, and the theoretically proven shortcomings of PER for actor-critic continuous control.

1.4.1 Recurrent Neural Networks (RNNs)

RNNs allow agents to learn with a memory of past observations, rewards and actions. They have been successfully applied in other video game environments such as Dota [2], Starcraft [14] and the Atari games [1]. For continuous control tasks, with partial observability, Yang et al. found Recurrent SAC (RSAC) to be the most reliable when compared to other recurrent versions of popular RL algorithms [15]. Furthermore, the recurrent version exceeded the performance of the normal version of SAC, when limited to partial observations of the environment. We therefore experiment with a recurrent version of SAC for the basic environment. We decide not to scale it for hardcore, due to limitations we encounter.

1.5 Architecture

For our critic we utilise two value networks, where we take the lower Q value. Each value network has the one hidden layer and is outlined in Table 1.

Table 1: Value Network Architecture

Layer Type	**Input Dims**	**Output Dims**
Linear Layer 1	24 + 4	256
Relu 1	256	256
Linear Layer 2	256	256
Relu 2	256	256
Linear Layer 3	256	1
Relu 3	1	1

Our policy network also has one hidden layer. It is used to produce both a mean and standard deviation for each action, creating a normal distribution which is then sampled from. Once sampled we apply tanh to scale the values between -1 and 1. The architecture of our policy is shown in Table 2.

Table 2: Policy Network Architecture

Layer Type	**Input Dims**	**Output Dims**
Linear Layer 1	24	256
Relu 1	256	256
Linear Layer 2	256	256
Relu 2	256	256
Mean Linear Layer	256	4
Log Std Linear Layer	256	4

2 Convergence Results

Results were obtained using Google Colab with Python 3.10. Training was conducted using only the CPU. Our code was largely based upon the code from the paper introducing LA3P [11].

As recommended by the paper introducing LA3P, we use a value of $\lambda = 0.5$ and reward scaling for the replay buffer by a factor of 5. However, we differ in that we use a batch size of 512 instead of the proposed 256, finding it to yield more stable reward graphs, and increased AUC. We experimented with a batch size of 1024 however found insufficient improvements in AUC to justify the increased training time.

Figure 2 shows our selected model's reward graph for the basic environment. Our model learns quickly, reaching its first reward exceeding 300 at episode 105. From then on the overwhelming majority of the rewards are greater than 300, with occasional reductions in performance. We attribute these to the stochastic nature of SAC. Since it is sampling from a normal distribution, it will occasionally sample extreme values that cause unexpected movements. Nevertheless, the AUC of the graph is large, and full convergence is reached from episode $1,200$ onwards. In addition, we obtain a max reward of 333 and quantitatively the videos demonstrate our robot's movements to be fluent, realistic and quick.

Our experiments with using LSTMs did not justify it replacing our standard model. We found that they took more episodes to converge and had reward graphs which were more unstable. We experimented by adding an additional layer in the policy and value network, reducing the reward stored in the replay buffer when the agent fell over and increasing the batch size. We were unable with any of these adjustments to produce models that were more sample efficient, as evaluated by the number of episodes to exceed a reward of 300. We believe this could be due to LSTMs requiring large amounts of data to effectively learn.

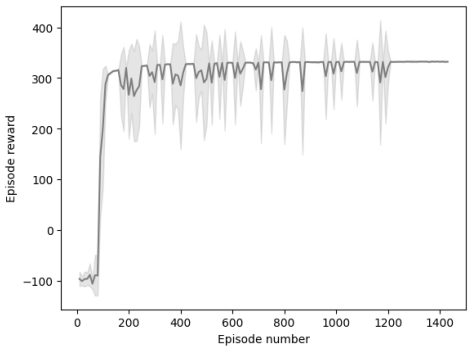

Figure 2: Our best model's reward graph in the basic environment

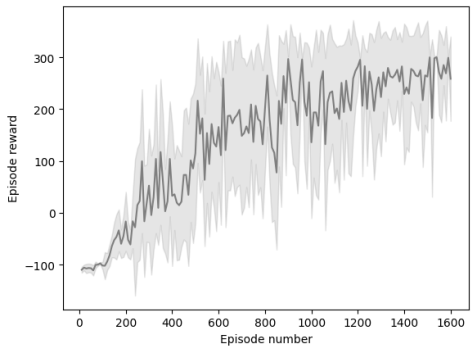

Figure 3: Our best model's reward graph in the hardcore environment

For the hardcore environment, due to to the increased difficulty, we reduce the punishment stored in the replay buffer (but do NOT alter the reward which is logged) when falling over, from -100 to -10. This allowed for a significant increase in performance when viewing the reward graph. We hypothesise that this is because with a larger punishment agents are more reluctant to attempt to climb obstacles. This is substantiated through the behaviour we observed in the videos of such models. Figure 3 shows our model's reward graph. We believe it challenges the current leading hardcore model, TD3 fork in sample efficiency [10]. Specifically, we achieve our first value over 300 within only 568 episodes whereas TD3 fork does so in 900 episodes [6].

3 Limitations

Due to the restrictions of 12 hours for running a Colab notebook, we were limited in how many episodes we could train our models for. As such, we do not get to see our hardcore model converge for the environment, although it reaches a mean just under 300.

Future Work

It is clear the SAC + LA3P is promising in the domain of continuous control. To further our work, it would be insightful to investigate whether this model converges in the hardcore environment. Additionally, although we use SAC, the authors published a version of TD3 that uses LA3P. We do not implement it since they report that their SAC version converges quicker for BipedalWalkerV3. However, it could be investigated whether with a larger batch size and hyperparameter tuning TD3 + LA3P could match its performance.

References

[1] Adrià Puigdomènech Badia et al. "Agent57: Outperforming the atari human benchmark". In: *International conference on machine learning*. PMLR. 2020, pp. 507–517.

[2] Christopher Berner et al. "Dota 2 with large scale deep reinforcement learning". In: *arXiv preprint arXiv:1912.06680* (2019).

[3] Scott Fujimoto, David Meger, and Doina Precup. "An equivalence between loss functions and non-uniform sampling in experience replay". In: *Advances in neural information processing systems* 33 (2020), pp. 14219–14230.

[4] Tuomas Haarnoja et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[5] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[6] Hong Hao. *TD3 FORK BipedalWalkerHardcore Colab Notebook*. `https://github.com/honghaow/FORK/blob/master/BipedalWalkerHardcore/TD3_FORK_BipedalWalkerHardcore_Colab.ipynb`. 2024.

[7] Vijay Konda and John Tsitsiklis. "Actor-critic algorithms". In: *Advances in neural information processing systems* 12 (1999).

[8] Hamid Maei et al. "Convergent temporal-difference learning with arbitrary smooth function approximation". In: *Advances in neural information processing systems* 22 (2009).

[9] Youngmin Oh et al. "Model-augmented prioritized experience replay". In: *International Conference on Learning Representations*. 2021.

[10] OpenAI Gym. *Leaderboard*. `https://github.com/openai/gym/wiki/Leaderboard`. 2024.

[11] Baturay Saglam et al. *Actor Prioritized Experience Replay*. `https://github.com/baturaysaglam/LA3P/tree/15bf010c6269b6dc3e758c31eb04feab51bfe404`. 2023.

[12] Baturay Saglam et al. "Actor prioritized experience replay". In: *Journal of Artificial Intelligence Research* 78 (2023), pp. 639–672.

[13] Tom Schaul et al. "Prioritized experience replay". In: *arXiv preprint arXiv:1511.05952* (2015).

[14] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782 (2019), pp. 350–354.

[15] Zhihan Yang and Hai Nguyen. "Recurrent off-policy baselines for memory-based continuous control". In: *arXiv preprint arXiv:2110.12628* (2021).