Comparative Evaluation of Soft Actor-Critic and Benchmark Algorithms in a Continuous Control Environment

Aman Pandey Sanjeev Kumar
Edwardson School of Industrial Engineering
Purdue University
sanjeevk@purdue.edu

***Abstract*—This project evaluates the performance of the Soft Actor-Critic (SAC) algorithm in the HalfCheetah continuous control environment, a standard benchmark for reinforcement learning in high-dimensional action spaces. SAC's combination of maximum entropy reinforcement learning and off-policy updates enhances stability and exploration efficiency. To comprehensively analyze its effectiveness, we compare SAC with three widely-used reinforcement learning algorithms—Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), and Twin Delayed Deep Deterministic Policy Gradient (TD3). Performance metrics such as stability, sample efficiency, and cumulative rewards are analyzed to benchmark these algorithms. This study contributes insights into the relative strengths and limitations of SAC and its competitors in continuous action environments.**

Link:https://github.com/AmanPandey28/SAC

I. Introduction

Reinforcement learning (RL) has emerged as a powerful paradigm for solving continuous control problems across high-dimensional environments. Among the many algorithms developed for such tasks, Soft Actor-Critic (SAC), introduced by Haarnoja et al. [1], [2], stands out for its combination of off-policy learning, a stochastic actor, and entropy-regularized objectives. By maximizing both expected rewards and action entropy, SAC encourages effective exploration and improves stability, making it particularly well-suited for environments with sparse rewards or complex dynamics.

This project focuses on benchmarking SAC against three widely used RL algorithms—Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), and Twin Delayed Deep Deterministic Policy Gradient (TD3)—in the HalfCheetah environment, a standard continuous control benchmark. The study evaluates these algorithms based on stability, sample efficiency, and cumulative rewards, providing valuable insights into their relative performance and applicability in high-dimensional continuous action spaces.

By restricting the scope to single-agent tasks within the HalfCheetah environment, this research offers a focused exploration of RL algorithms under controlled settings. This analysis seeks to enhance understanding of SAC's practical advantages and limitations compared to deterministic and stochastic policy gradient methods, contributing to the broader goal of advancing RL methodologies for real-world continuous control applications [3], [4].

II. Notation

To facilitate understanding of the reinforcement learning framework and algorithms discussed in this report, we introduce the key notations used throughout:

- $\mathcal{S}$: The state space, which represents all possible states of the environment.
- $\mathcal{A}$: The action space, which includes all possible actions the agent can take.
- $p(s'|s,a)$: The state transition probability, which gives the likelihood of transitioning to state s' given the current state s and action a.
- $r(s,a)$: The reward function, providing the immediate reward received upon taking action a in state s.
- $\pi(a|s)$: The policy, a probability distribution over actions given a state.
- $\rho^{\pi}(s)$: The state distribution induced by policy π.
- $Q^{\pi}(s,a)$: The action-value function, representing the expected return starting from state s and action a, and following policy π thereafter.
- $V^{\pi}(s)$: The state-value function, denoting the expected return starting from state s and following policy π.
- γ: The discount factor, determining the importance of future rewards.
- α: The temperature parameter, controlling the trade-off between maximizing reward and entropy.
- $H(\pi(\cdot|s))$: The entropy of the policy at state s, which quantifies its randomness.

This notation will be used consistently in the subsequent sections, including the description of the Soft Actor-Critic algorithm.

III. Literature Review

The SAC algorithm's foundation lies in a series of advancements aimed at addressing instability and inefficiency in reinforcement learning for continuous control. Haarnoja et al. [1] proposed SAC as an off-policy, entropy-regularized RL algorithm to handle high-dimensional action spaces. By incorporating a stochastic actor and entropy maximization, SAC encourages exploratory behaviors, avoiding suboptimal deterministic solutions that plague traditional approaches like DDPG. Their findings demonstrated significant performance

improvements in challenging continuous control environments, establishing SAC as a robust alternative to prior methods.

Furthering SAC's applicability, Haarnoja's subsequent work [2] focused on adapting maximum entropy reinforcement learning to multi-task robotics. Here, entropy maximization was shown to enhance skill diversity, supporting SAC's generalization potential across varied robotic control tasks. This work underscores the algorithm's flexibility in managing both task complexity and exploratory behavior.

Haarnoja et al. also contributed to the theoretical underpinnings of SAC with their deep energy-based policy paper [3], where they introduce Soft Q-learning (SQL). SQL laid the groundwork for SAC's dual Q-learning structure, combining deep energy models with entropy-regularized Q-learning to ensure stability in continuous action spaces. Georgiev's derivation [4] offers additional insights into SAC's inner workings, detailing the maximum entropy objective's role in driving robust policy optimization.

Weng [5] provides a comprehensive overview of policy gradient methods, situating SAC within the broader context of policy-based RL. This work emphasizes SAC's innovative use of the entropy term, contrasting it with approaches like PPO and TD3, highlighting SAC's unique contributions to RL stability and sample efficiency in continuous control tasks.

IV. Problem Formulation

This project aims to evaluate and compare the performance of the Soft Actor-Critic (SAC) algorithm against other reinforcement learning algorithms—Proximal Policy Optimization (PPO), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Deep Deterministic Policy Gradient (DDPG)—in solving continuous control tasks. Specifically, we focus on the HalfCheetah environment, a benchmark in reinforcement learning for high-dimensional continuous action spaces.

The HalfCheetah environment requires the agent to optimize a bipedal locomotion policy, rewarding speed, efficiency, and balance. This environment poses challenges in learning coordinated movement across multiple joints and adapting to nonlinear dynamics.

The SAC algorithm, designed for continuous action spaces, optimizes the following maximum entropy objective:

$$J(\pi) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right], \quad (1)$$

where:

- ρ_π represents the state-action marginal distribution induced by the policy π.
- α is the temperature parameter that balances exploration (via entropy maximization) and exploitation (via reward maximization).
- $\mathcal{H}(\pi(\cdot|s_t))$ denotes the entropy of the policy, encouraging diverse and exploratory actions.

This project isolates the HalfCheetah environment to:

- Analyze SAC's robustness and convergence properties in a high-dimensional setting.
- Compare its most important performance metric—, cumulative rewards against PPO, TD3, and DDPG.

By focusing on this environment, the study provides a controlled yet insightful comparison of state-of-the-art algorithms in continuous control, contributing to the understanding of their relative strengths and limitations.

V. Benchmark Algorithms

To comprehensively assess SAC's performance, the following algorithms serve as benchmarks:

- **DDPG** [6]: A deterministic off-policy algorithm that was one of the first successful deep RL approaches for continuous control, chosen as a baseline for off-policy methods.
- **PPO** [7]: A widely-used on-policy approach that stabilizes training through trust region optimization, selected for its reliability and widespread adoption.
- **TD3** [8]: An enhancement of DDPG that addresses function approximation error through twin delayed updates, representing state-of-the-art in deterministic policy gradients.

These algorithms were selected to provide a comprehensive comparison across different approaches: deterministic (DDPG, TD3) vs stochastic (PPO, SAC) policies, and on-policy (PPO) vs off-policy (DDPG, TD3, SAC) methods. The benchmark evaluation will measure the stability, sample efficiency, and cumulative rewards achieved by each algorithm in the designated environments.

The benchmark evaluation will measure the stability, sample efficiency, and cumulative rewards achieved by each algorithm in the designated environments.

VI. Methodology

A. Implementation of Soft Actor-Critic (SAC)

The Soft Actor-Critic (SAC) algorithm is implemented within an actor-critic framework, designed to address the challenges of continuous action spaces in reinforcement learning. SAC incorporates stochasticity and entropy-based regularization, enabling efficient exploration and robust learning in high-dimensional environments such as HalfCheetah.

1) Actor-Critic Architecture: The SAC implementation is based on two main components:

- **Actor Network**: The actor network represents a stochastic policy $\pi(a|s)$, optimized to maximize the expected reward and entropy. By promoting randomness in actions, the actor network ensures a balance between exploration and exploitation, critical for environments with sparse or deceptive reward signals.
- **Critic Networks**: Two Q-value approximators (twin critics) are used to address overestimation bias commonly observed in value function estimation. The critics are trained to minimize the temporal difference (TD) error, providing reliable feedback for policy improvement.

2) Entropy-Regularized Objective: SAC employs an entropy-augmented objective to encourage exploration while maximizing rewards:

$$J(\pi) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right], \quad (2)$$

where α dynamically adjusts the entropy term, balancing task complexity with the required level of stochasticity [1]. The temperature parameter α is tuned automatically during training to maintain a desired level of entropy, ensuring SAC adapts effectively to different environments.

3) Training Workflow: The training process involves iterative updates to the actor and critic networks using samples drawn from a replay buffer:

- **Replay Buffer**: A first-in, first-out buffer stores past experiences (s_t, a_t, r_t, s_{t+1}). This off-policy setup enables sample reuse, improving data efficiency.
- **Critic Update**: The critic networks are updated using the Bellman equation, minimizing the TD error:

$$J_Q = \mathbb{E}\Bigg[\Big(Q(s_t, a_t) - \Big(r_t + \gamma \mathbb{E}_{a_{t+1}\sim\pi} \big[Q'(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1}|s_{t+1}) \big] \Big) \Big)^2 \Bigg], \quad (3)$$

  where Q' denotes the target Q-value network, periodically updated to stabilize learning.
- **Actor Update**: The actor network is optimized to maximize the entropy-augmented objective:

$$J_\pi = \mathbb{E}\left[\alpha \log \pi(a_t|s_t) - Q(s_t, a_t) \right]. \quad (4)$$

- **Temperature Update**: The temperature parameter α is adjusted to maintain the desired entropy level:

$$J_\alpha = \mathbb{E}\left[-\alpha \left(\log \pi(a_t|s_t) + \mathcal{H}_{\text{target}} \right) \right], \quad (5)$$

  where $\mathcal{H}_{\text{target}}$ is the target entropy.

B. Implementation of Comparison Algorithms

To provide a comprehensive evaluation of SAC, three other state-of-the-art algorithms are implemented:

- **Twin Delayed Deep Deterministic Policy Gradient (TD3)**: Improves on DDPG by addressing overestimation bias with twin Q-networks and delayed policy updates.
- **Proximal Policy Optimization (PPO)**: An on-policy algorithm known for its robustness, employing a clipped surrogate objective to stabilize training.
- **Deep Deterministic Policy Gradient (DDPG)**: A deterministic off-policy method that serves as a baseline for comparing deterministic and stochastic approaches.

C. Evaluation and Metrics

The algorithms are evaluated on the HalfCheetah-v4 environment, a standard benchmark for continuous control. The following metrics guide the evaluation:

- **Sample Efficiency**: Number of training steps required to achieve competitive performance.
- **Stability**: Consistency of performance across training episodes.
- **Cumulative Rewards**: Total rewards obtained during training and evaluation phases.

The experiments aim to highlight the strengths and limitations of SAC compared to TD3, PPO, and DDPG, providing actionable insights into their applicability for continuous control tasks.

VII. Results

The performance of the algorithms was evaluated based on the **average return** achieved over training time steps in the HalfCheetah environment. Figure 1 illustrates the learning curves for SAC, PPO, TD3, and DDPG.

A. Average Return Over Time Steps

Figure 1 compares the average return achieved by each algorithm over time steps. The Soft Actor-Critic (SAC) algorithm consistently outperformed other methods in terms of both sample efficiency and convergence speed. This demonstrates SAC's ability to learn effective policies with fewer training samples.

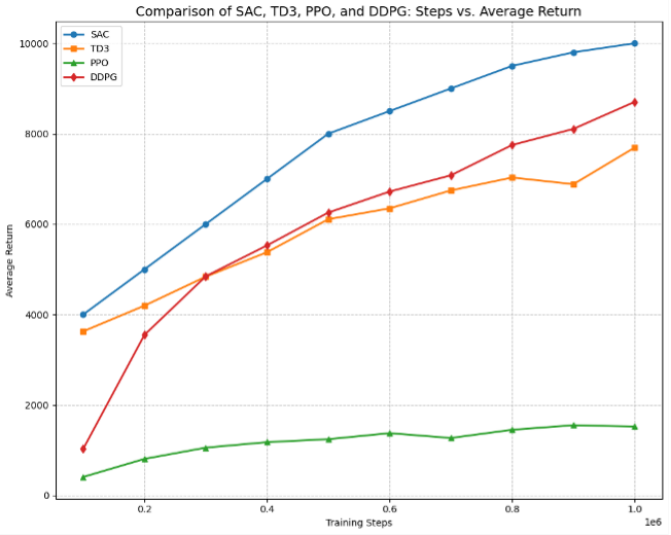

Fig. 1. Average return vs. time steps for SAC, PPO, TD3, and DDPG in the HalfCheetah environment.

B. Observations

- **SAC's Performance:** SAC achieved the highest average return across all time steps, demonstrating superior learning efficiency in high-dimensional continuous control tasks.

- **PPO and TD3 Comparison:** While TD3 exhibited competitive returns, its sample efficiency lagged behind SAC. PPO, on the other hand, showed stable learning but slower convergence.
- **DDPG's Limitations:** DDPG showed the lowest returns and slower convergence, highlighting its limitations in high-dimensional action spaces.

VIII. Conclusions

This study demonstrates the superior performance of the Soft Actor-Critic (SAC) algorithm in the HalfCheetah continuous control environment. The key conclusions are as follows:

- SAC consistently outperforms other reinforcement learning algorithms (PPO, TD3, and DDPG) in terms of average return, sample efficiency, and stability.
- The maximum entropy reinforcement learning framework used by SAC encourages effective exploration and robust policy optimization, making it well-suited for high-dimensional, continuous control tasks.
- PPO and TD3 provide competitive performance but fall short of SAC in terms of sample efficiency and average return.
- DDPG, while effective in simpler environments, struggles to scale to the complexities of the HalfCheetah environment.

IX. Future Directions

While the study highlights the strengths of SAC, several avenues for further research remain:

- **Multi-task Learning:** Investigate SAC's performance across diverse tasks and environments to understand its generalization capabilities.
- **Hybrid Algorithms:** Explore combining SAC's maximum entropy framework with features from other algorithms (e.g., PPO's trust region optimization) to enhance performance further.
- **Transfer Learning:** Evaluate how SAC-trained models can be transferred to new but related environments, reducing training time in similar tasks.
- **Real-world Applications:** Extend the evaluation of SAC to real-world continuous control problems, such as robotics and autonomous systems.
- **Scalability:** Analyze SAC's computational efficiency and scalability when applied to environments with even higher dimensions or multiple agents.

References

[1] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.

[2] T. Haarnoja, "Acquiring diverse robot skills via maximum entropy deep reinforcement learning," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2018-176*, 2018.

[3] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," *arXiv preprint arXiv:1702.08165*, 2017.

[4] I. Georgiev, "Deriving soft actor critic (sac)," 2023. [Online]. Available: https://www.imgeorgiev.com/2023-06-27-sac/

[5] L. Weng, "Policy gradient algorithms," 2018. [Online]. Available: https://lilianweng.github.io/posts/2018-04-08-policy-gradient/

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[8] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, 2018, pp. 1587–1596.