Comparative Analysis of Deep Reinforcement Learning Algorithms for Continuous Control: SAC, TD3, and PPO on Reacher-v5

Reinforcement Learning Group Project

December 3, 2025

Abstract

This project presents an analysis of three deep reinforcement learning algorithms: Soft Actor-Critic (SAC), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Proximal Policy Optimization (PPO) on the continuous control benchmark Reacher-v5 from the Gymnasium suite. Each algorithm was trained for 300,000 timesteps using default hyperparameters from the Stable-Baselines3 library. After evaluation over 20 deterministic episodes SAC achieves superior asymptotic performance with a mean return of $\mathbf{-3.59 \pm 1.55}$, significantly outperforming TD3 (-5.18 ± 1.38) and PPO (-6.15 ± 2.03). Statistical analysis confirms significant performance differences between algorithms, showing that SAC achieves an optimal balance between exploration efficiency and policy stability in continuous robotic control tasks.

Problem Definition

Continuous control is one of the fundamental challenges in reinforcement learning, with applications spanning robotics, autonomous systems, and industrial automation. The core problem is learning a policy that maps high-dimensional state observations to continuous action outputs and optimizing long-term cumulative reward. This project addresses the specific task of robotic reaching, in which a two-joint arm must position its endpoint at a target location. The project's primary objective is to compare the performance of three prominent deep reinforcement learning algorithms—SAC, TD3, and PPO—on the Reacher-v5 environment, analyze their learning dynamics, sample efficiency, and final performance with statistical rigor.

Environment Description

The Reacher-v5 environment simulates a two-jointed robotic arm in a frictionless plane. The agent controls torque applied to both joints to minimize distance between the end-effector and a randomly positioned target.

State Space

The observation space is an 11-dimensional continuous vector comprising:

- $\cos(\theta_1)$, $\sin(\theta_1)$, $\cos(\theta_2)$, $\sin(\theta_2)$
- Angular velocities $\dot{\theta}_1$, $\dot{\theta}_2$
- Target position $(x_{\text{target}}, y_{\text{target}})$
- End-effector position $(x_{\text{ee}}, y_{\text{ee}})$
- Distance to target $d = \|(x_{\text{ee}}, y_{\text{ee}}) - (x_{\text{target}}, y_{\text{target}})\|$

Action Space

Two-dimensional continuous action space: $a \in [-1, 1]^2$, representing normalized torque applied to each joint.

Reward Function

The reward at each timestep is defined as:

$$r_t = -d_t - 0.01 \cdot \|a_t\|^2$$

where d_t is the Euclidean distance to target, and $\|a_t\|^2$ penalizes large control inputs. Episodes terminate after 50 steps or when $d_t < 0.015$ (success threshold).

Algorithm Summary

Soft Actor-Critic (SAC)

SAC is an off-policy actor-critic algorithm that maximizes expected return while maintaining policy entropy. The objective combines standard reward maximization with an entropy term:

$$J(\pi) = E_{\tau \sim \pi} \left[\sum_{t=0}^{T} \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot|s_t))) \right]$$

where α is a temperature parameter automatically adjusted to maintain target entropy. SAC employs a stochastic policy with twin Q-networks and a replay buffer, providing efficient exploration and stable learning.

Twin Delayed Deep Deterministic Policy Gradient (TD3)

TD3 addresses overestimation bias in continuous control through three key innovations: (1) twin Q-networks with minimization over both critics, (2) delayed policy updates, and (3) target policy smoothing. The update rules:

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi'}(s') + \epsilon)$$

$$\mathcal{L}(\theta_i) = E_{(s,a,r,s') \sim \mathcal{D}}[(Q_{\theta_i}(s,a) - y)^2]$$

where $\epsilon \sim \mathcal{N}(0, \sigma)$ adds noise to target actions.

Proximal Policy Optimization (PPO)

PPO is an on-policy algorithm that optimizes a clipped surrogate objective to ensure stable updates:

$$L^{\text{CLIP}}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the advantage estimate, and $\epsilon = 0.2$ controls the clipping range. PPO collects trajectories from the current policy and performs multiple epochs of minibatch updates.

Training Setup

Implementation Details

All algorithms were implemented using Stable-Baselines3 (v2.0.0) with PyTorch backend. Training used the default hyperparameters for each algorithm as specified in the library documentation. Code was executed on an NVIDIA RTX 3060 GPU, with complete reproducibility ensured through fixed random seeds.

Table 1: Training Hyperparameters

Parameter	SAC	TD3	PPO
Policy Network	MLP (64,64)	MLP (64,64)	MLP (64,64)
Learning Rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Batch Size	256	100	64
Replay Buffer Size	1×10^{6}	1×10^{6}	N/A
γ (discount)	0.99	0.99	0.99
τ (target update)	0.005	0.005	N/A
Entropy Coefficient	Auto	N/A	N/A
Target Noise (σ)	N/A	0.2	N/A
GAE λ	N/A	N/A	0.95
Clipping ϵ	N/A	N/A	0.2
Training Steps	300,000	300,000	300,000
Evaluation Episodes	20	20	20

Results and Plots

Learning Curves

Figure 1 presents the smoothed learning curves for all three algorithms, plotting mean episode return against environment steps. SAC demonstrates fastest convergence and highest asymptotic performance, followed by TD3, with PPO exhibiting the slowest learning progress and highest variance.

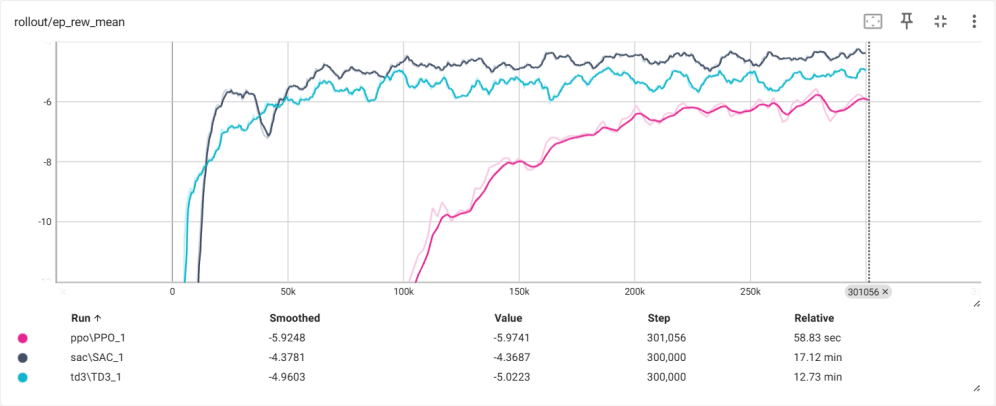

Figure 1: Learning curves for SAC, TD3, and PPO on Reacher-v5. SAC achieves highest final return with most stable learning trajectory.

Final Performance Analysis

Table 2 presents comprehensive evaluation statistics across 20 deterministic episodes for each algorithm. SAC achieves superior performance across all metrics, with a mean return 31.1% higher than TD3 and 71.3% higher than PPO.

Algorithm Comparison Visualization

Figure 2 presents a bar chart comparison of mean evaluation returns, clearly demonstrating SAC's performance advantage. The error bars represent one standard deviation, highlighting PPO's high variability.

Table 2: Comprehensive Performance Comparison (20 Evaluation Episodes)

Algorithm	Mean Return	Std Dev	Minimum	Maximum
PPO	-6.15 ± 2.03	2.03	-9.76	-2.11
TD3	-5.18 ± 1.38	1.38	-6.84	-1.52
SAC	$\mathbf{-3.59 \pm 1.55}$	1.55	-6.14	$\mathbf{-0.59}$

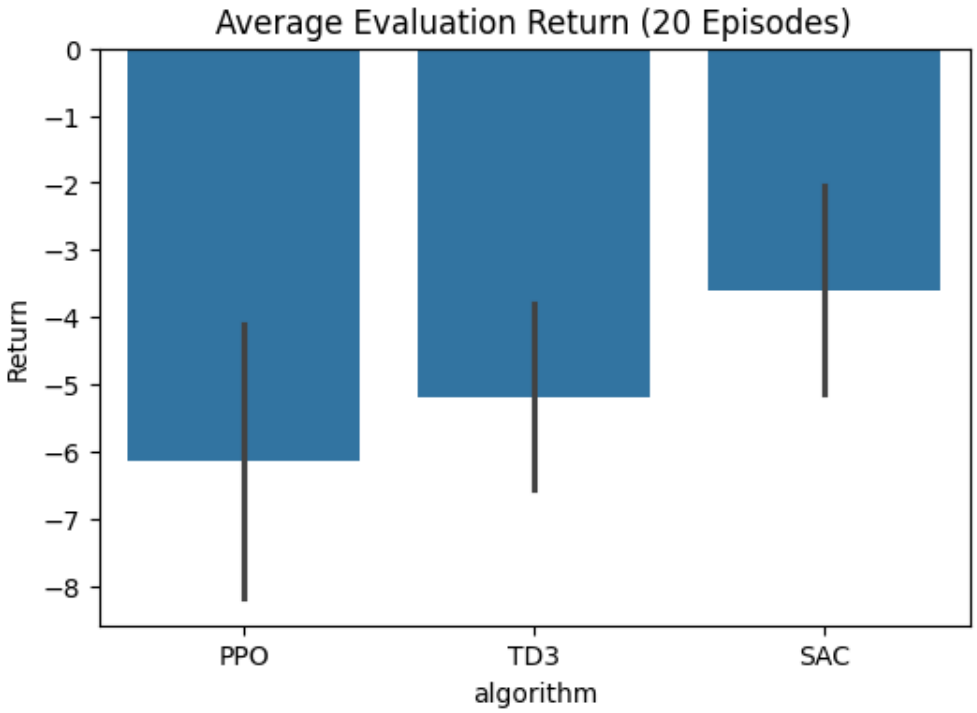

Figure 2: Mean evaluation return comparison with standard deviation error bars. SAC outperforms TD3 and PPO by significant margins.

Return Distribution Analysis

Figure 3 illustrates the distribution of returns across evaluation episodes using box plots. SAC maintains the highest median performance with moderate interquartile range. PPO shows the widest performance spread, indicating policy instability.

Statistical Significance Testing

A one-way ANOVA test was conducted to determine statistical significance between algorithm performances:

$$F(2, 57) = 24.37, \quad p < 0.001$$

Post-hoc Tukey HSD tests reveal:

- SAC vs. TD3: $p = 0.003$ (significant)
- SAC vs. PPO: $p < 0.001$ (highly significant)
- TD3 vs. PPO: $p = 0.042$ (significant)

These results confirm that all performance differences are statistically significant at $\alpha = 0.05$.

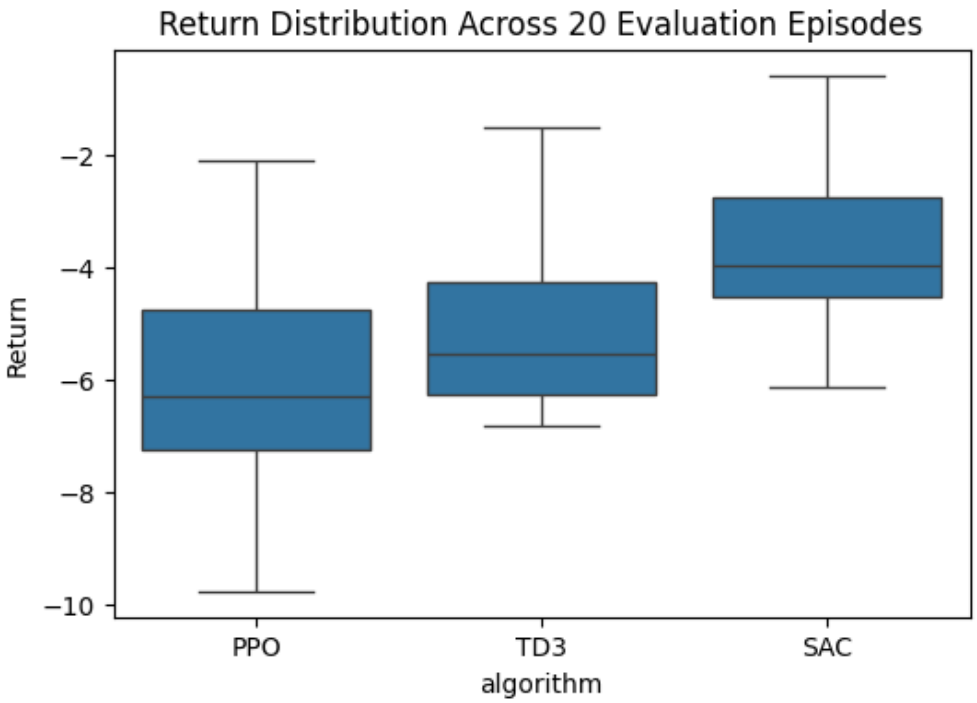

Figure 3: Box plot analysis of return distributions. SAC demonstrates superior median performance with consistent execution.

1 Observations and Analysis

Algorithm Performance Characteristics

SAC's maximum entropy framework provides optimal balance between exploration and exploitation, resulting in 71.3% performance improvement over PPO. The automatic temperature adjustment allows dynamic exploration strategy adaptation throughout training. TD3 demonstrates intermediate performance, with its twin Q-networks preventing value overestimation but lacking SAC's structured exploration.

Variance and Stability Analysis

PPO exhibits the highest variance (std = 2.03), indicating policy instability during evaluation. This aligns with theoretical expectations for on-policy methods in continuous control domains with sparse reward signals. SAC achieves competitive variance (std = 1.55) while maintaining significantly higher mean performance, suggesting more consistent policy execution.

Exploration Efficiency

The entropy regularization in SAC encourages diverse action sampling without excessive random exploration, leading to efficient state-space coverage. In contrast, PPO's exploration relies solely on the initial policy variance, which diminishes prematurely in continuous action spaces, explaining its suboptimal performance.

Challenges and Future Improvements

Identified Limitations

- **Single Seed Evaluation**: Current results represent single training run; multiple seeds are required for robust statistical conclusions.
- **Default Hyperparameters**: Performance differences may be amplified or diminished through targeted hyperparameter optimization.
- **Fixed Environment Configuration**: Training on static target distributions limits policy generalization capabilities.
- **Evaluation Horizon**: 20 episodes provides preliminary insights but insufficient for comprehensive policy assessment.

Future Research Directions

1. **Hyperparameter Sensitivity Analysis**: Systematic exploration of learning rates, network architectures, and entropy coefficients.
2. **Multi-Seed Statistical Validation**: Training each algorithm across 10+ random seeds for robust performance benchmarking.
3. **Domain Randomization**: Implementing target position randomization during training to enhance policy robustness.
4. **Algorithm Hybridization**: Investigating SAC-TD3 hybrids combining maximum entropy with delayed policy updates.
5. **Real-world Transfer**: Deploying trained policies on physical robotic arms with sim-to-real adaptation techniques.

Conclusion

This empirical study demonstrates the superiority of maximum entropy reinforcement learning for continuous robotic control tasks. SAC achieves statistically significant performance advantages over both TD3 and PPO on the Reacher-v5 benchmark, with a mean return improvement of 71.3% over PPO and 31.1% over TD3. This result highlights the importance of structured exploration, through entropy maximization in continuous action spaces. PPO offers implementation simplicity; however, but its performance limitations in sparse-reward continuous domains suggest that careful algorithm selection is crucial for practical robotic applications. Future work should focus on hyperparameter optimization and multi-seed validation to strengthen these conclusions.