**Soft Actor-Critic:**

**Off-Policy Maximum Entropy Deep Reinforcement**

**Learning with a Stochastic Actor**

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine

Berkeley Artificial Intelligence Research, University of California,

Berkeley, USA. Correspondence to: Tuomas Haarnoja

<[haarnoja@berkeley.edu](mailto:haarnoja@berkeley.edu)>.

**Abstract**

Model-free deep reinforcement learning (RL) algorithms have been demonstrated on a range of challenging decision making and control tasks. However, these methods typically suffer from two major challenges: very high sample complexity and brittle convergence properties, which necessitate meticulous hyperparameter tuning.

모델이 없는 심층 강화 학습(RL) 알고리즘은 다양한 도전적인 의사 결정 및 제어 작업에서 증명되었습니다. 그러나 이러한 방법은 일반적으로 매우 높은 샘플 복잡성과 깨지기 쉬운 수렴 속성이라는 두 가지 주요 문제로 인해 세심한 하이퍼파라미터 조정이 필요합니다.

Both of these challenges severely limit the applicability of such methods to complex, real-world domains.

이러한 두 가지 문제는 복잡한 실제 영역에 대한 이러한 방법의 적용 가능성을 심각하게 제한합니다.

In this paper, we propose soft actor-critic, an offpolicy actor-critic deep RL algorithm based on the maximum entropy reinforcement learning framework.

본 논문에서는 소프트 행위자 비평가, 최대 엔트로피 강화 학습 프레임워크를 기반으로 한 비정책 행위자 심층 RL 알고리즘을 제안한다.

In this framework, the actor aims to maximize expected reward while also maximizing entropy. That is, to succeed at the task while acting as randomly as possible. Prior deep RL methods based on this framework have been formulated as Q-learning methods. By combining off-policy updates with a stable stochastic actor-critic formulation, our method achieves state-of-the-art performance on a range of continuous control benchmark tasks, outperforming prior on-policy and off-policy methods.

이 프레임워크에서 행위자는 기대 보상을 최대화하는 동시에 엔트로피도 최대화하는 것을 목표로 합니다. 즉, 가능한 한 무작위로 행동하면서 작업에 성공하는 것입니다. 이 프레임워크를 기반으로 하는 기존의 deep RL 방법은 Q-learning 방법으로 공식화되었습니다. 정책 외 업데이트를 안정적인 확률론적 행위자 비평가 공식과 결합하여 우리의 방법은 다양한 연속 제어 벤치마크 작업에서 최첨단 성능을 달성하여, 이전의 정책 및 정책 외 방법을 능가합니다.

Furthermore, we demonstrate that, in contrast to other off-policy algorithms, our approach is very stable, achieving very similar performance across different random seeds.

또한 다른 오프 정책 알고리즘과 달리 우리의 접근 방식이 매우 안정적이며 다른 임의 시드에서 매우 유사한 성능을 달성한다는 것을 증명해줍니다.

**1. Introduction**

Model-free deep reinforcement learning (RL) algorithms have been applied in a range of challenging domains, from games (Mnih et al., 2013; Silver et al., 2016) to robotic control (Schulman et al., 2015).

모델이 없는 심층 강화 학습(RL) 알고리즘은 게임(Mnih et al., 2013; Silver et al., 2016)에서 로봇 제어(Schulman et al., 2015)에 이르기까지 다양한 도전적인 영역에 적용되었습니다.

The combination of RL and high-capacity function approximators such as neural networks holds the promise of automating a wide range of decision making and control tasks, but widespread adoption of these methods in real-world domains has been hampered by two major challenges.

RL과 신경망과 같은 고용량 함수 근사기의 조합은 광범위한 의사 결정 및 제어 작업을 자동화할 수 있는 가능성을 제공하지만 실제 영역에서 이러한 방법의 광범위한 채택은 두 가지 주요 과제로 인해 방해를 받았습니다.

First, model-free deep RL methods are notoriously expensive in terms of their sample complexity. Even relatively simple tasks can require millions of steps of data collection, and complex behaviors with high-dimensional observations might need substantially more.

첫째, 모델이 없는 심층 RL 방법은 샘플 복잡성 면에서 악명이 높습니다. 비교적 단순한 작업일지라도 수백만 단계의 데이터 수집이 필요할 수 있으며 고차원 관찰이 포함된 복잡한 동작은 훨씬 더 많은 단계가 필요할 수 있습니다.

Second, these methods are often brittle with respect to their hyperparameters: learning rates, exploration constants, and other settings must be set carefully for different problem settings to achieve good results.

둘째, 이러한 방법은 하이퍼파라미터와 관련하여 취약한 경우가 많습니다. 좋은 결과를 얻으려면 다양한 문제 설정에 대해 학습률, 탐색 상수 및 기타 설정을 신중하게 설정해야 합니다.

Both of these challenges severely limit the applicability of model-free deep RL to real-world tasks.

이러한 두 가지 문제는 모델이 없는 deep RL을 실제 작업에 적용하는 것을 심각하게 제한합니다.

One cause for the poor sample efficiency of deep RL methods is on-policy learning: some of the most commonly used deep RL algorithms, such as TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017b) or A3C (Mnih et al., 2016), require new samples to be collected for each gradient step. This quickly becomes extravagantly expensive, as the number of gradient steps and samples per step needed to learn an effective policy increases with task complexity.

심층 RL 방법의 열악한 샘플 효율성에 대한 한 가지 원인은 정책에 따른 학습입니다. TRPO(Schulman et al., 2015), PPO(Schulman et al., 2017b) 또는 A3C(Mnih et al., 2016)와 같이 가장 일반적으로 사용되는 심층 RL 알고리즘 중 일부는 각 기울기 단계에 대해 새로운 샘플을 수집해야 합니다. 효과적인 정책을 학습하는 데 필요한 그래디언트 단계 및 단계당 샘플의 수가 작업 복잡성에 따라 증가하기 때문에 이는 빠르게 비용이 많이 들게 됩니다.

Off-policy algorithms aim to reuse past experience. This is not directly feasible with conventional policy gradient formulations, but is relatively straightforward for Q-learning based methods (Mnih et al., 2015).

정책 외 알고리즘은 과거 경험을 재사용하는 것을 목표로 합니다. 이것은 기존의 정책 기울기 공식에서는 직접적으로 실현 가능하지 않지만 Q-학습 기반 방법에서는 비교적 간단합니다(Mnih et al., 2015).

Unfortunately, the combination of off-policy learning and high-dimensional, nonlinear function approximation with neural networks presents a major challenge for stability and convergence (Bhatnagar et al., 2009). This challenge is further exacerbated in continuous state and action spaces, where a separate actor network is often used to perform the maximization in Q-learning.

불행히도 정책 외 학습과 신경망을 사용한 고차원, 비선형 함수 근사의 조합은 안정성과 수렴에 대한 주요 과제를 제시합니다(Bhatnagar et al., 2009). 이 문제는 Q-러닝에서 최대화를 수행하기 위해 별도의 행위자 네트워크가 자주 사용되는 연속 상태 및 동작 공간에서 더욱 악화됩니다.

A commonly used algorithm in such settings, deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015), provides for sample-efficient learning but is notoriously challenging to use due to its extreme brittleness and hyperparameter sensitivity (Duan et al., 2016; Henderson et al., 2017).

이러한 설정에서 일반적으로 사용되는 알고리즘인 DDPG(Deep Deterministic Policy Gradient)(Lillicrap et al., 2015)는 샘플 효율적인 학습을 제공하지만 극도의 취성 및 하이퍼파라미터 감도(Duan et al., 2016; Henderson et al., 2017)로 인해 사용은 악명높은 도전이다.

We explore how to design an efficient and stable model-free deep RL algorithm for continuous state and action spaces. To that end, we draw on the maximum entropy framework, which augments the standard maximum reward reinforcement learning objective with an entropy maximization term (Ziebart et al., 2008; Toussaint, 2009; Rawlik et al., 2012; Fox et al., 2016; Haarnoja et al., 2017).

연속 상태 및 동작 공간을 위한 효율적이고 안정적인 모델 없는 심층 RL 알고리즘을 설계하는 방법을 탐구합니다. 이를 위해 엔트로피 최대화 항으로 표준 최대 보상 강화 학습 목표를 증가시키는 최대 엔트로피 프레임워크를 사용합니다(Ziebart et al., 2008; Toussaint, 2009; Rawlik et al., 2012; Fox et al., 2016; Haarnoja et al., 2017).

Maximum entropy reinforcement learning alters the RL objective, though the original objective can be recovered using a temperature parameter (Haarnoja et al., 2017).

최대 엔트로피 강화 학습은 RL 목표를 변경하지만, 원래 목표는 온도 매개변수를 사용하여 복구할 수 있습니다(Haarnoja et al., 2017).

More importantly, the maximum entropy formulation provides a substantial improvement in exploration and robustness: as discussed by Ziebart (2010), maximum entropy policies are robust in the face of model and estimation errors, and as demonstrated by (Haarnoja et al., 2017), they improve exploration by acquiring diverse behaviors.

더 중요한 것은, 최대 엔트로피 공식은 탐색 및 견고성에서 상당한 개선을 제공한다는 것입니다. Ziebart(2010)가 논의한 바와 같이 최대 엔트로피 정책은 모델 및 추정 오류에 직면하여 견고하며, (Haarnoja et al., 2017)가 언급한 것처럼 그들은 다양한 행동을 습득함으로써 탐색을 향상시킵니다.

Prior work has proposed model-free deep RL algorithms that perform on-policy learning with entropy maximization (O’Donoghue et al., 2016), as well as off-policy methods based on soft Q-learning and its variants (Schulman et al., 2017a; Nachum et al., 2017a; Haarnoja et al., 2017). However, the on-policy variants suffer from poor sample complexity for the reasons discussed above, while the off-policy variants require complex approximate inference procedures in continuous action spaces.

이전 작업에서는 엔트로피 최대화(O'Donoghue et al., 2016)를 사용하여 정책 내 학습을 수행하는 모델 없는 심층 RL 알고리즘과 소프트 Q-학습 및 그 변형에 근거하여 off-policy 학습을 수행하는 모델 없는 심층 RL 알고리즘(Schulman et al. ., 2017a; Nachum et al., 2017a; Haarnoja et al., 2017)이 제안되었다. 그러나 on-policy 변형은 위에서 논의한 이유로 열악한 샘플 복잡성으로 고통받는 반면, off-policy 변형은 연속 작업 공간에서 복잡한 근사 추론 절차를 필요로 합니다.

In this paper, we demonstrate that we can devise an off-policy maximum entropy actor-critic algorithm, which we call soft actor-critic (SAC), which provides for both sample-efficient learning and stability.

이 논문에서 우리는 off-policy 최대 엔트로피 Actor-Critic 알고리즘, 즉 SAC(Soft Actor-Critic)를 고안할 수 있음을 보여두며, 이 알고리즘은 샘플 효율적인 학습과 안정성을 모두 제공한다.

This algorithm extends readily to very complex, high-dimensional tasks, such as the Humanoid benchmark (Duan et al., 2016) with 21 action dimensions, where off-policy methods such as DDPG typically struggle to obtain good results (Gu et al., 2016).

이 알고리즘은 21개의 작업 차원이 있는 휴머노이드 벤치마크(Duan et al., 2016)와 같은 매우 복잡한 고차원 작업으로 쉽게 확장되며(Gu et al. , 2016), 여기서 DDPG와 같은 off-policy 방법은 일반적으로 좋은 결과를 얻기 위해 고군분투한다.

SAC also avoids the complexity and potential instability associated with approximate inference in prior off-policy maximum entropy algorithms based on soft Q-learning (Haarnoja et al., 2017).

SAC는 또한 soft Q-learning을 기반으로 하는 이전의 off-policy 최대 엔트로피 알고리즘의 근사 추론과 관련된 복잡성 및 잠재적인 불안정성을 피합니다(Haarnoja et al., 2017).

We present a convergence proof for policy iteration in the maximum entropy framework, and then introduce a new algorithm based on an approximation to this procedure that can be practically implemented with deep neural networks, which we call soft actor-critic.

우리는 최대 엔트로피 프레임워크에서 정책 반복에 대한 수렴 증명을 제시한 다음 이 절차에 대한 근사를 기반으로 하는 새로운 알고리즘을 소개합니다. 이 알고리즘은 우리가 soft actor-critic라고 부르는 심층 신경망으로 실제로 구현할 수 있습니다.

We present empirical results that show that soft actor-critic attains a substantial improvement in both performance and sample efficiency over both off-policy and on-policy prior methods.

우리는 soft actor-critic가 off-policy 및 on-policy 이전 방법보다, 성능과 샘플 효율성 모두에서 상당한 개선을 달성한다는 것을 보여주는 경험적 결과를 제시합니다.

We also compare to twin delayed deep deterministic(TD3) policy gradient algorithm (Fujimoto et al., 2018), which is a concurrent work that proposes a deterministic algorithm that substantially improves on DDPG.

또한 DDPG를 실질적으로 개선하는 deterministic 알고리즘을 제안하는 작업인 트윈 지연 심층 결정론(TD3) 정책 기울기 알고리즘(Fujimoto et al., 2018)과 비교합니다.

**2. Related Work**

Our soft actor-critic algorithm incorporates three key ingredients: an actor-critic architecture with separate policy and value function networks, an off-policy formulation that enables reuse of previously collected data for efficiency, and entropy maximization to enable stability and exploration.

우리의 soft actor-critic algorithm은 세 가지 핵심 요소를 통합한 것: 분리 정책 및 가치 기능 네트워크가 있는 actor-critic 아키텍처, 효율성을 위해 이전에 수집된 데이터를 재사용할 수 있는 off-policy formulation, 안정성 및 탐색을 가능하게 하는 엔트로피 최대화

We review prior works that draw on some of these ideas in this section. Actor-critic algorithms are typically derived starting from policy iteration, which alternates between policy evaluation-computing the value function for a policy- and policy improvement-using the value function to obtain a better policy (Barto et al., 1983; Sutton & Barto, 1998).

우리는 이 섹션에서 이러한 아이디어 중 일부를 끌어내는 이전 작업을 검토합니다. Actor-critic 알고리즘은 일반적으로 policy evaluation(정책에 대한 가치 함수 계산)와 더 나은 정책을 얻기 위해 가치 함수를 사용하는 policy improvement 사이를 번갈아 하는 정책 반복에서 파생됩니다(Barto et al., 1983; Sutton & Barto , 1998).

In large-scale reinforcement learning problems, it is typically impractical to run either of these steps to convergence, and instead the value function and policy are optimized jointly.

대규모 강화 학습 문제에서 수렴을 위해 이러한 단계 중 하나를 실행하는 것은 일반적으로 비실용적이며, 대신 가치 함수와 정책이 공동으로 최적화됩니다.

In this case, the policy is referred to as the actor, and the value function as the critic. Many actor-critic algorithms build on the standard, on-policy policy gradient formulation to update the actor (Peters & Schaal, 2008), and many of them also consider the entropy of the policy, but instead of maximizing the entropy, they use it as an regularizer (Schulman et al., 2017b; 2015; Mnih et al., 2016; Gruslys et al., 2017).

이 경우 정책을 행위자(actor)라고 하고, 가치 함수를 critic(비판자)라고 합니다. 많은 actor-critic 알고리즘은 행위자를 업데이트하기 위해 표준 정책에 따른 정책(standard, on-policy policy) 기울기 공식을 기반으로 한다(Peters & Schaal, 2008). 그리고, 이 알고리즘의 대부분은 정책의 엔트로피를 고려하지만 엔트로피를 최대화하는 대신 정책의 엔트로피를 정규화기(regularizer)로 사용합니다(Schulman et al., 2017b; 2015; Mnih et al., 2016; Gruslys et al., 2017).

On-policy training tends to improve stability but results in poor sample complexity. There have been efforts to increase the sample efficiency while retaining robustness by incorporating off-policy samples and by using higher order variance reduction techniques(O’Donoghue et al., 2016; Gu et al., 2016).

On-policy 학습은 안정성을 향상시키는 경향이 있으나, 결과적으로 표본 복잡성이 낮습니다. off-policy 샘플을 통합하고 고차 분산 감소 기술을 사용하여 robustness를 유지하면서 샘플 효율성을 높이려는 노력이 있었습니다(O'Donoghue et al., 2016; Gu et al., 2016).

However, fully off-policy algorithms still attain better efficiency. A particularly popular off-policy actor-critic method, DDPG (Lillicrap et al., 2015), which is a deep variant of the deterministic policy gradient (Silver et al., 2014) algorithm, uses a Q-function estimator to enable off-policy learning, and a deterministic actor that maximizes this Q-function.

그러나, fully off-policy 알고리즘은 여전히 더 나은 효율성을 달성합니다. 특히 인기 있는 off-policy actor-critic 방법인 DDPG(Lillicrap et al., 2015)는 deterministic policy gradient(결정론적 정책 기울기)(Silver et al., 2014) 알고리즘의 심층 변형으로, off-policy 학습을 가능케하는 Q-함수 추정기(Q-function estimator)와 이 Q-함수를 최대화하는 deterministic actor(결정론적 행위자)를 사용한다.

As such, this method can be viewed both as a deterministic actor-critic algorithm and an approximate Q-learning algorithm.

따라서 이 방법은 deterministic actor-critic(결정론적 행위자 비평가) 알고리즘과 approximate Q-learning algorithm(근사 Q-학습 알고리즘)으로 볼 수 있습니다.

Unfortunately, the interplay between the deterministic actor network and the Q-function typically makes DDPG extremely difficult to stabilize and brittle to hyperparameter settings (Duan et al., 2016; Henderson et al., 2017).

불행히도, deterministic actor network(결정론적 행위자 네트워크)와 Q-함수 사이의 상호 작용은 일반적으로 DDPG를 안정화하기 매우 어렵게 만들고, 초매개변수 설정에 취약하게 만듭니다(Duan et al., 2016; Henderson et al., 2017).

As a consequence, it is difficult to extend DDPG to complex, high-dimensional tasks, and on-policy policy gradient methods still tend to produce the best results in such settings (Gu et al., 2016).

결과적으로 DDPG를 복잡한 고차원 작업으로 확장하는 것은 어렵고, on-policy policy gradient(정책에 따른 정책 기울기) 방법은 이러한 설정에서 여전히 최상의 결과를 생성하는 경향이 있습니다(Gu et al., 2016).

Our method instead combines off-policy actor-critic training with a stochastic actor, and further aims to maximize the entropy of this actor with an entropy maximization objective. We find that this actually results in a considerably more stable and scalable algorithm that, in practice, exceeds both the efficiency and final performance of DDPG.

대신 우리의 방법은 off-policy actor-critic training(비정책 행위자 비평 학습)을 stochastic actor(확률적 행위자)와 결합하고, 엔트로피 최대화 목표로 이 행위자의 엔트로피를 최대화하는 것을 목적으로 합니다. 우리는 이것이 실제로 DDPG의 효율성과 최종 성능을 모두 능가하는 훨씬 더 안정적이고 확장 가능한 알고리즘을 생성한다는 것을 발견했습니다.

A similar method can be derived as a zero-step special case of stochastic value gradients (SVG(0)) (Heess et al., 2015). However, SVG(0) differs from our method in that it optimizes the standard maximum expected return objective, and it does not make use of a separate value network, which we found to make training more stable.

stochastic value gradients(확률적 값 기울기)(SVG(0))의 0단계 특수 사례로 유사한 방법을 유도할 수 있습니다(Heess et al., 2015). 그러나 SVG(0)은 표준 최대 기대 수익 목표(standard maximum expected return objective)를 최적화하고, 분리(별도?) 가치 네트워크를 사용하지 않는다는 점에서 우리 방법과 다릅니다.

Maximum entropy reinforcement learning optimizes policies to maximize both the expected return and the expected entropy of the policy. This framework has been

used in many contexts, from inverse reinforcement learning (Ziebart et al., 2008) to optimal control (Todorov, 2008; Toussaint, 2009; Rawlik et al., 2012).

최대 엔트로피 강화 학습은 정책을 최적화하여 정책의 기대 수익과 기대 엔트로피를 모두 최대화합니다. 이러한 프레임워크는 inverse reinforcement learning(역 강화 학습)(Ziebart et al., 2008)에서 최적 제어(Todorov, 2008; Toussaint, 2009; Rawlik et al., 2012)에 이르기까지 많은 상황에서 사용됩니다.

In guided policy search (Levine & Koltun, 2013; Levine et al., 2016), the maximum entropy distribution is used to guide policy learning towards high-reward regions.

가이드된 정책 검색(Levine & Koltun, 2013; Levine et al., 2016)에서 최대 엔트로피 분포는 높은 보상 영역으로 정책 학습을 안내하는 데 사용됩니다.

More recently, several papers have noted the connection between Q-learning and policy gradient methods in the framework of maximum entropy learning (O’Donoghue et al., 2016; Haarnoja et al., 2017; Nachum et al., 2017a; Schulman et al., 2017a).

보다 최근에 여러 논문에서 최대 엔트로피 학습의 프레임워크에서 Q-학습과 정책 기울기 방법 간의 연결에 대해 언급했습니다(O'Donoghue et al., 2016; Haarnoja et al., 2017; Nachum et al., 2017a; Schulman et al., 2017a). 알., 2017a).

While most of the prior model-free works assume a discrete action space, Nachum et al. (2017b) approximate the maximum entropy distribution with a Gaussian and Haarnoja et al. (2017) with a sampling network trained to draw samples from the optimal policy. Although the soft Q-learning algorithm proposed by Haarnoja et al. (2017) has a value function and actor network, it is not a true actor-critic algorithm: the Q-function is estimating the optimal Q-function, and the actor does not directly affect the Q-function except through the data distribution.

이전의 모델이 없는 작업의 대부분은 이산적인 작업 공간을 가정하는 반면 Nachum et al. (2017b)은 Gaussian을 사용하여 최대 엔트로피 분포를 근사화, Haarnoja et al. (2017)은 최적의 정책으로부터 샘플을 추출하도록 훈련된 샘플링 네트워크를 사용하여 사용하여 최대 엔트로피 분포를 근사화합니다. 비록 Haarnoja et al.에 의해 제안된 soft Q-learning 알고리즘(2017)은 가치 함수와 행위자 네트워크를 가지고 있지만, true actor-critic(진정한 행위자 비평가) 알고리즘이 아닙니다 : Q-함수는 optimal Q-function(최적의 Q-함수)를 추정하고, 행위자(actor)는 데이터 분포를 통하지 않고는 Q-함수에 직접적인 영향을 미치지 않습니다.

Hence, Haarnoja et al. (2017) motivates the actor network as an approximate sampler, rather than the actor in an actor-critic algorithm.

따라서 Haarnoja et al. (2017)에서 actor 네트워크는 actor-critic 알고리즘에서 actor가 아닌 근사 샘플러로서 동기를 부여합니다.

Crucially, the convergence of this method hinges on how well this sampler approximates the true posterior. In contrast, we prove that our method converges to the optimal policy from a given policy class, regardless of the policy parameterization.

결정적으로, 이 방법의 수렴은 이 샘플러가 실제 사후에 얼마나 잘 근접하는지에 달려 있습니다. 대조적으로, 우리는 우리의 방법이 정책 매개 변수화에 관계없이 주어진 정책 클래스에서 최적의 정책으로 수렴함을 증명합니다.

Furthermore, these prior maximum entropy methods generally do not exceed the performance of state-of-the-art off-policy algorithms, such as DDPG, when learning from scratch, though they may have other benefits, such as improved exploration and ease of fine-tuning. In our experiments, we demonstrate that our soft actor-critic algorithm does in fact exceed the performance of prior state-of-the-art off-policy deep RL methods by a wide margin.

또한 이들 사전 최대 엔트로피 방법은 일반적으로 scratch로부터 학습할 때 DDPG와 같은 오프 정책 알고리즘의 성능을 초과하지 않지만, 개선된 탐색 및 fine-tuning의 용이성과 같은 다른 이점이 있을 수 있습니다. 우리의 실험에서 우리는 our soft actor-critic algorithm(우리의 연성 행위자 비평가 알고리즘)이 실제로 이전의 off-policy deep RL methods(오프 정책 심층 RL 방법)의 성능을 크게 능가한다는 것을 보여줍니다.

**3. Preliminaries**

We first introduce notation and summarize the standard and maximum entropy reinforcement learning frameworks.

먼저 표기법을 소개하고 표준 최대 엔트로피 강화 학습 프레임워크를 요약

**3.1. Notation**

We address policy learning in continuous action spaces.

우리는 지속적인 행동 공간에서 정책 학습을 다룹니다.

We consider an infinite-horizon Markov decision process (MDP), defined by the tuple ( , , , ), where

the state space and the action space are continuous, and

the unknown state transition probability represents the probability density of the next state 

given the current state and action at .

The environment emits a bounded reward on each transition. We will use and to denote the state and state-action marginals of the trajectory distribution induced by a policy .

**3.2. Maximum Entropy Reinforcement Learning**

Standard RL maximizes the expected sum of rewards



We will consider a more general maximum entropy objective (see e.g. Ziebart (2010)), which favors stochastic policies by augmenting the objective with the expected entropy of the policy over :

텍스트이(가) 표시된 사진

자동 생성된 설명

The temperature parameter determines the relative importance of the entropy term against the reward, and thus controls the stochasticity of the optimal policy.

The maximum entropy objective differs from the standard maximum expected reward objective used in conventional reinforcement learning, though the conventional objective can be recovered in the limit as .

최대 엔트로피 목표는 기존의 강화 학습에서 사용되는 표준 최대 기대 보상 목표와 다르다.

For the rest of this paper, we will omit writing the temperature explicitly, as it can always be subsumed into the reward by scaling it by .

이 논문의 하단에서는 온도를 명시적으로 작성하는 것을 생략할 것입니다.

온도를 로 스케일링하여 항상 보상에 포함될 수 있기 때문에.

This objective has a number of conceptual and practical advantages. First, the policy is incentivized to explore more widely, while giving up on clearly unpromising avenues. Second, the policy can capture multiple modes of nearoptimal behavior. In problem settings where multiple actions seem equally attractive, the policy will commit equal probability mass to those actions. Lastly, prior work has observed improved exploration with this objective (Haarnoja et al., 2017; Schulman et al., 2017a), and in our experiments, we observe that it considerably improves learning speed over state-of-art methods that optimize the conventional RL objective function.

이 objective(목표)는 많은 개념적 및 실제적 이점을 가지고 있습니다. 첫째, 이 정책은 분명히 유망하지 않은 길을 포기하면서 더 광범위하게 탐색하도록 장려됩니다. 둘째, 정책은 최적에 가까운 행동의 여러 모드를 포착할 수 있습니다. 여러 작업이 똑같이 매력적으로 보이는 문제 설정에서 정책은 해당 작업에 동일한 확률을 적용합니다. 마지막으로, 이전 작업(Haarnoja et al., 2017; Schulman et al., 2017a)에서는 이 목표를 사용하여 개선된 탐색을 관찰했으며, 우리의 방법이 기존의 RL 목적 함수를 최적화하는 기존 방법보다 학습 속도를 상당히 향상시키는 것을 관찰했습니다.

We can extend the objective to infinite horizon problems by introducing a discount factor to ensure that the sum of expected rewards and entropies is finite. Writing down the maximum entropy objective for the infinite horizon discounted case is more involved (Thomas, 2014) and is deferred to Appendix A.

기대 보상과 엔트로피의 합이 finite(유한)하다는 것을 보장하도록 discount factor(할인 인자)을 도입하여, objective(목표)를 infinite horizon problems으로 확장할 수 있습니다. infinite horizon discounted case(사례)에 대한 최대 엔트로피 목적을 기록하는 것은 더 복잡하며(Thomas, 2014) 부록 A로 이월됩니다.

Prior methods have proposed directly solving for the optimal Q-function, from which the optimal policy can be recovered (Ziebart et al., 2008; Fox et al., 2016; Haarnoja et al., 2017).

이전의 방법들은 최적의 정책을 복구할 수 있는 최적의 Q-함수에 대해 직접 푸는 것을 제안했습니다(Ziebart et al., 2008; Fox et al., 2016; Haarnoja et al., 2017).

We will discuss how we can devise a soft actor-critic algorithm through a policy iteration formulation, where we instead evaluate the Q-function of the current policy and update the policy through an off-policy gradient update.

policy iteration formulation(정책 반복으로 수식화)을 통해 soft actor-critic algorithm(연성 행위자 비평가 알고리즘)을 고안하는 방법에 대해 논의할 것입니다. 여기에서 대신 현재 정책의 Q-함수를 평가하고 off-policy gradient update(정책 외 그라디언트 업데이트)를 통해 정책을 업데이트합니다.

Though such algorithms have previously been proposed for conventional reinforcement learning, our method is, to our knowledge, the first off-policy actor-critic method in the maximum entropy reinforcement learning framework.

이러한 알고리즘은 기존의 강화 학습에 대해 이전에 제안되었으나, 우리가 아는 한 우리의 방법은 최대 엔트로피 강화 학습 프레임워크에서 first off-policy actor-critic method(정책 외 행위자 비평가 방법 중 첫 번째 방법)입니다.

**4. From Soft Policy Iteration to Soft Actor-Critic**

Our off-policy soft actor-critic algorithm can be derived starting from a maximum entropy variant of the policy iteration method. We will first present this derivation, verify that the corresponding algorithm converges to the optimal policy from its density class, and then present a practical deep reinforcement learning algorithm based on this theory.

우리의 off-policy soft Actor-Critic 알고리즘은 policy iteration method(정책 반복 방법)의 최대 엔트로피 변형에서 시작하여 유도할 수 있습니다.

먼저 이 유도를 제시하고, 해당 알고리즘이 밀도 클래스에서 최적의 정책으로 수렴하는지 확인한 다음 이 이론을 기반으로 하는 실용적인 심층 강화 학습 알고리즘을 제시합니다.

**4.1. Derivation of Soft Policy Iteration**

We will begin by deriving soft policy iteration, a general algorithm for learning optimal maximum entropy policies that alternates between policy evaluation and policy improvement in the maximum entropy framework.

최대 엔트로피 프레임워크에서 정책 평가와 정책 개선을 번갈아 가며 최적의 최대 엔트로피 정책을 학습하기 위한 일반적인 알고리즘인 soft policy iteration(소프트 정책 반복)을 유도하는 것으로 시작합니다.

Our derivation is based on a tabular setting, to enable theoretical analysis and convergence guarantees, and we extend this method into the general continuous setting in the next section.

우리의 유도는 이론적 분석 및 수렴 보장을 가능하게 하기 위해 표 설정을 기반으로 하며, 다음 섹션에서 이 방법을 general continuous setting(일반 연속 설정)으로 확장합니다.

We will show that soft policy iteration converges to the optimal policy within a set of policies which might correspond, for instance, to a set of parameterized densities.

우리는 soft policy iteration(소프트 정책 반복)이, 예를 들면 일련의 매개변수화된 밀도 집합에 해당할 수 있는 정책 집합 내에서, 최적의 정책으로 수렴한다는 것을 보여줄 것입니다.

In the policy evaluation step of soft policy iteration, we wish to compute the value of a policy according to the maximum entropy objective in Equation 1.

For a fixed policy, the soft Q-value can be computed iteratively, starting from any function



and

repeatedly applying a modified Bellman backup operator



given by

텍스트이(가) 표시된 사진

자동 생성된 설명

is the soft state value function.

We can obtain the soft value function for any policy 

by repeatedly applying as formalized below.

텍스트이(가) 표시된 사진

자동 생성된 설명

In the policy improvement step, we update the policy towards the exponential of the new Q-function.

정책 개선 단계에서, 우리는 새로운 Q-함수의 지수에 대한 정책(the policy towards the exponential of the new Q-function)을 업데이트합니다.

This particular choice of update can be guaranteed to result in an improved policy in terms of its soft value.

위의 업데이트 방법은 soft value(소프트 가치) 측면에서 개선된 정책의 결과를 보장할 수 있습니다.

Since in practice we prefer policies that are tractable, we will additionally restrict the policy to some set of policies , which can correspond, for example, to a parameterized family of distributions such as Gaussians.

실제로 우리는 다루기 쉬운 정책을 선호하기 때문에, 우리는 정책을 일련의 정책들 (예를 들어, 가우시안과 같은 매개변수화된 분포 집합에 대응될 수 있다.)의 일부로 추가로 제한할 것입니다.

To account for the constraint that , we project the improved policy into the desired set of policies.

제약 조건 을 설명하기 위해, 개선된 정책(improved policy)을 desired set of policies(원하는 정책 집합)에 투영합니다.

While in principle we could choose any projection, it will turn out to be convenient to use the information projection defined in terms of the Kullback-Leibler divergence.

원칙적으로 우리는 어떤 투영도 선택할 수 있지만, Kullback-Leibler divergence의 관점에서 정의된 정보 투영을 사용하는 것이 편리할 것입니다.

In the other words, in the policy improvement step, for each state, we update the policy according to

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

The partition function normalizes the distribution, and while it is intractable in general, it does not contribute to the gradient with respect to the new policy and can thus be ignored, as noted in the next section.

partition function(분할 함수) 는 분포를 정규화하고, (일반적으로 다루기 힘들지만) gradient with respect to the new policy(새 정책에 대한 기울기)에 기여하지 않으므로 다음 섹션에서 설명하는 것처럼 무시할 수 있습니다.

For this projection, we can show that the new, projected policy has a higher value than the old policy with respect to the objective in Equation 1.

이 projection에 대해, 예측을 위해 우리는 식 1의 목표와 관련하여 새로운 projected 정책이 old policy with respect to the objective in Equation 1.(식1의 목적함수에 대한 이전 정책)보다 더 높은 가치를 갖는다는 것을 보여줄 수 있습니다. 이

We formalize this result in Lemma 2.

결과를 보조정리 2에서 공식화합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

The full soft policy iteration algorithm alternates between the soft policy evaluation and the soft policy improvement steps, and it will provably converge to the optimal maximum entropy policy among the policies in (Theorem 1).

full soft policy iteration algorithm(전체 소프트 정책 반복 알고리즘)은 소프트 정책 평가와 소프트 정책 개선 단계를 번갈아 수행하며, (정리 1)의 정책들 중 최적의 최대 엔트로피 정책으로 수렴함을 증명할 것입니다.

Although this algorithm will provably find the optimal solution, we can perform it in its exact form only in the tabular case. Therefore, we will next approximate the algorithm for continuous domains, where we need to rely on a function approximator to represent the Q-values, and running the two steps until convergence would be computationally too expensive. The approximation gives rise to a new practical algorithm, called soft actor-critic.

이 알고리즘은 최적의 솔루션을 찾을 수 있지만, 우리는 단지 tabular case(테이블 형식)의 exact form(정확한 형태)로 수행할 수 있습니다. 따라서 우리는 다음으로 algorithm for continuous domains(연속 영역에 대한 알고리즘)을 근사화할 것이다. 이렇게 하기 위해, function approximator(함수 근사기)를 사용해서 Q-값을 나타내고, 수렴이 계산적으로 너무 비쌀 때까지 두 단계를 실행합니다. 이 근사화는 soft actor-critic(연성 배우 비평가)이라고 하는 새로운 실용적인 알고리즘을 제공한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**4.2. Soft Actor-Critic**

As discussed above, large continuous domains require us to derive a practical approximation to soft policy iteration.

위에서 논의한 바와 같이, large continuous domains(큰 연속 도메인)은 practical approximation to soft policy iteration(소프트 정책 반복에 대한 실용적인 근사치)를 도출해야 합니다.

To that end, we will use function approximators for both the Q-function and the policy, and instead of running evaluation and improvement to convergence, alternate between optimizing both networks with stochastic gradient descent.

이를 위해, Q-함수와 정책 모두에 함수 근사기들을 사용하고 수렴에 대한 평가 및 개선을 실행하는 대신 확률적 경사하강법으로 두 네트워크를 최적화하는 방법을 번갈아 사용합니다.

We will consider

a parameterized state value function ,

soft Q-function , and

a tractable policy .

The parameters of these networks are , , and .

For example, the value functions can be modeled as expressive neural networks, and the policy as a Gaussian with mean and covariance given by neural networks.

We will next derive update rules for these parameter vectors.

The state value function approximates the soft value. There is no need in principle to include a separate function approximator for the state value, since it is related to the Q-function and policy according to Equation 3.

수식 3에 의하면, 그것은 Q-function 및 정책과 관련이 있기 때문에, 원칙적으로 separate function approximator for the state value(state 값에 대한 별도의 function approximator)를 포함할 필요는 없다.

This quantity can be Soft Actor-Critic estimated from a single action sample from the current policy without introducing a bias, but in practice, including a separate function approximator for the soft value can stabilize training and is convenient to train simultaneously with the other networks.

This quantity(이 양)는 바이어스(bias)를 도입하지 않고 current policy(현재 정책)의 single action sample(단일 작업 샘플)로부터 추정한 Soft Actor-Critic일 수 있으나, 실제로는 separate function approximator for the soft value(soft 값에 대한 별도의 함수 근사기)를 포함하면 학습을 안정화할 수 있고, 다른 네트워크와 동시에 학습하는 것이 편리합니다.

The soft value function is trained to minimize the squared residual error

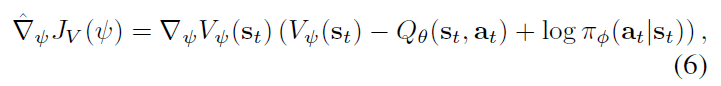
텍스트이(가) 표시된 사진

자동 생성된 설명

where

is the distribution of previously sampled states and actions(이전에 샘플링된 상태 및 작업의 분포), or a replay buffer(재생 버퍼).

The gradient of Equation 5 can be estimated with an unbiased estimator



where the actions are sampled according to the current policy, instead of the replay buffer.

The soft Q-function parameters can be trained to minimize the soft Bellman residual

텍스트이(가) 표시된 사진

자동 생성된 설명

The update makes use of a target value network , where can be an exponentially moving average of the value network weights, which has been shown to stabilize training (Mnih et al., 2015).

Alternatively, we can update the target weights to match the current value function weights periodically (see Appendix E).

대안적인 방법으로, current value function weights(현재 값 함수 가중치)와 주기적으로 일치하도록 target weights(목표 가중치)를 업데이트할 수 있습니다(부록 E 참조).

Finally, the policy parameters can be learned by directly minimizing the expected

KL-divergence in Equation 4:

마지막으로, policy parameters(정책 매개변수)는 식 4의 expected KL-divergence를 직접 최소화함으로써 학습될 수 있습니다.

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

There are several options for minimizing .

A typical solution for policy gradient methods is to use the likelihood ratio gradient estimator (Williams, 1992), which does not require backpropagating the gradient through the policy and the target density networks.

policy gradient methods(정책 기울기 방법)에 대한 typical solution(일반적인 솔루션)은 the policy and the target density networks(정책 및 목표 밀도 네트워크)를 통해 기울기를 역전파할 필요가 없는 likelihood ratio gradient estimator를 사용하는 것입니다(Williams, 1992).

However, in our case, the target density is the Q-function, which is represented by a neural network and can be differentiated, and it is thus convenient to apply the reparameterization trick instead, resulting in a lower variance estimator.

그러나, 우리의 경우, target density(목표 밀도)는 신경망으로 표현되는 Q-함수로, Q-함수는 미분할 수 있으므로 대신 reparameterization trick(재매개변수화 트릭)을 적용하는 것이 편리하며, 궁극적으로 낮은 분산 추정기로 만들 수 있다.

To that end, we reparameterize the policy using a neural network transformation

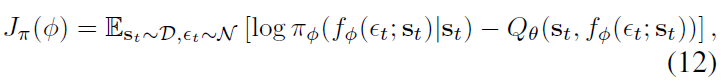


텍스트이(가) 표시된 사진

자동 생성된 설명

where is an input noise vector, sampled from some fixed distribution, such as a spherical Gaussian.

We can now **rewrite the objective in (10)** as



where is defined implicitly in terms of , and we have noted that the partition function is independent of and can thus be omitted.

We **can approximate the gradient of (12)** with

텍스트이(가) 표시된 사진

자동 생성된 설명

where is evaluated at . **This unbiased gradient estimator** extends the DDPG style policy gradients (Lillicrap et al., 2015) to any tractable stochastic policy.

Our algorithm also makes **use of two Q-functions** to mitigate positive bias in the policy improvement step that is known to degrade performance of value based methods (Hasselt,

2010; Fujimoto et al., 2018).

우리의 알고리즘은 또한 value based methods(가치 기반 방법)의 성능을 저하시키는 것으로 알려진 정책 개선 단계에서의 positive bias를 완화시키기 위해 2개의 Q-함수들을 사용한다(Hasselt, 2010; Fujimoto et al., 2018).

In particular, we parameterize two Q-functions, with parameters , and train them independently to optimize .

특히, 우리는 두 개의 Q-함수를 매개변수화((parameters 의 함수)하고, 를 최적화하기 위해 두 개의 Q-함수를 독립적으로 학습시킨다.

We then use the minimum of the Q-functions for the value gradient in Equation 6 and policy gradient in Equation 13, as proposed by Fujimoto et al. (2018).

그런 다음 Fujimoto et al. (2018)에 의해 제안된 식 (6)의 값의 기울기와 식 (13)의 정책 기울기에 대해 Q-함수의 최소값을 사용합니다.

Although our algorithm can learn challenging tasks, including a 21-dimensional Humanoid, using just a single Q-function, we found two Q-functions significantly speed up training, especially on harder tasks.

우리 알고리즘은 단일 Q-함수를 사용하여 21차원 휴머노이드를 포함한 어려운 작업을 학습할 수 있지만, 특히 더 어려운 작업에서 두 개의 Q-함수가 학습 속도를 크게 향상시킨다는 것을 발견했습니다.

The complete algorithm is described in Algorithm 1. The method alternates between collecting experience from the environment with the current policy and updating the function approximators using the stochastic gradients from batches sampled from a replay buffer.

전체 알고리즘은 알고리즘 1에 설명되어 있습니다. 이 방법은 현재 정책의 환경에서 경험을 수집하는 것과 재생 버퍼에서 샘플링된 배치의 확률적 기울기를 사용하여 함수 근사치를 업데이트하는 방법을 번갈아 사용합니다.

In practice, we take a single environment step followed by one or several gradient steps (see Appendix D for all hyperparameter). Using off-policy data from a replay buffer is feasible because both value estimators and the policy can be trained entirely on off-policy data. The algorithm is agnostic to the parameterization of the policy, as long as it can be evaluated for any arbitrary state-action tuple.

실제로 우리는 단일 환경 단계를 수행한 후 하나 또는 여러 그라디언트 단계를 수행합니다(모든 하이퍼파라미터에 대해서는 부록 D 참조). 재생 버퍼에서 off-policy data(정책 외 데이터)를 사용하는 것은 가치 추정기와 정책이 모두 off-policy data(정책 외 데이터)에 대해 전적으로 학습될 수 있기 때문에 실현 가능합니다. 알고리즘은 arbitrary state-action tuple(임의의 상태-동작 튜플)에 대해 평가될 수 있는 한, 알고리즘은 정책의 매개변수화에 영향을 받지 않습니다.

**5. Experiments**

The goal of our experimental evaluation is to understand how the sample complexity and stability of our method compares with prior off-policy and on-policy deep reinforcement learning algorithms.

우리의 실험 평가의 목표는 우리 방법의 샘플 복잡성과 안정성이 이전의 off-policy(정책 외) 및 on-policy(정책 기반) 심층 강화 학습 알고리즘에 대해 비교하는 방법을 이해하는 것입니다.

We compare our method to prior techniques on a range of challenging continuous control tasks from the OpenAI gym benchmark suite (Brockman et al., 2016) and also on the rllab implementation of the Humanoid task (Duan et al., 2016).

우리는 **OpenAI 체육관 벤치마크 제품군(Brockman et al., 2016)과 Humanoid 작업의 rllab 구현(Duan et al., 2016)**의 a range of challenging continuous control tasks(도전적인 연속 제어 작업 범위)에 관한 이전 기술과 우리의 방법을 비교합니다.

Although the easier tasks can be solved by a wide range of different algorithms, the more complex benchmarks, such as the 21-dimensional Humanoid (rllab), are exceptionally difficult to solve with off-policy algorithms (Duan et al., 2016).

더 쉬운 작업은 다양한 알고리즘으로 해결할 수 있지만 21차원 휴머노이드(rllab)와 같은 더 복잡한 벤치마크는 off-policy algorithms(정책 외 알고리즘)으로 해결하기가 매우 어렵습니다(Duan et al., 2016).

The stability of the algorithm also plays a large role in performance: easier tasks make it more practical to tune hyperparameters to achieve good results, while the already narrow basins of effective hyperparameters become prohibitively small for the more sensitive algorithms on the hardest benchmarks, leading to poor performance (Gu et al., 2016).

알고리즘의 안정성은 또한 성능에 큰 역할을 합니다 : 작업이 쉬워지면 하이퍼파라미터를 조정하여 좋은 결과를 얻는 것이 더 실용적입니다. 반면, 효과적인 하이퍼파라미터의 좁은 유역은 가장 어려운 벤치마크에서 더 민감한 알고리즘에 대해 엄청나게 작아져 결국 낮은 성능을 갖게 합니다(Gu et al., 2016).

We compare our method to deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015), an algorithm that is regarded as one of the more efficient off-policy deep RL methods (Duan et al., 2016); proximal policy optimization (PPO) (Schulman et al., 2017b), a stable and effective on-policy policy gradient algorithm; and soft Q-learning (SQL) (Haarnoja et al., 2017), a recent off-policy algorithm for learning maximum entropy policies.

우리는 우리의 방법을 보다 효율적인 오프 정책 심층 RL 방법 중 하나로 간주되는 알고리즘인 DDPG(Deep Deterministic Policy Gradient)(Lillicrap et al., 2015)(Duan et al., 2016) ; 근위 정책 최적화(PPO)(Schulman et al., 2017b), 안정적이고 효과적인 on-policy(정책 내) 정책 기울기 알고리즘; 및 최대 엔트로피 정책을 학습하기 위한 최근 off-policy(오프 정책) 알고리즘인 soft Q-learning(SQL)(Haarnoja et al., 2017)와 비교합니다

Our SQL implementation also includes two Q-functions, which we found to improve its performance in most environments. We additionally compare to twin delayed deep deterministic policy gradient algorithm (TD3) (Fujimoto et al., 2018), using the author-provided implementation.

우리의 SQL 구현에는 또한 대부분의 환경에서 성능을 향상시키는 두 개의 Q-함수가 포함되어 있습니다.

저자가 제공한 구현을 사용하여 twin delayed deep deterministic policy gradient algorithm(TD3)(Deep Deterministic Policy Gradient Algorithm)(Fujimoto et al., 2018)와 추가로 비교합니다.

This is an extension to DDPG, proposed concurrently to our method, that first applied the double Q-learning trick to continuous control along with other improvements.

이것은 DDPG의 확장으로, 우리의 방법에서 현재 제안한 것이며, 다른 개선 사항과 함께 연속 제어에 이중 Q-학습 트릭을 먼저 적용한다.

We have included trust region path consistency learning (Trust-PCL) (Nachum et al., 2017b) and two other variants of SAC in Appendix E. We turned off the exploration noise for evaluation for DDPG and PPO.

부록 E에 신뢰 영역 경로 일관성 학습(Trust-PCL)(Nachum et al., 2017b)과 SAC의 다른 두 가지 변형을 포함했습니다. DDPG 및 PPO에 대한 평가를 위해 탐색 노이즈를 해제했습니다.

For maximum entropy algorithms, which do not explicitly inject exploration noise, we either evaluated with the exploration noise (SQL) or use the mean action (SAC). The source code of our SAC implementation1 and videos2 are available online.

탐색 잡음을 명시적으로 주입하지 않는 최대 엔트로피 알고리즘의 경우, 탐색 잡음(SQL)으로 평가하거나 mean action(평균 동작)(SAC)을 사용합니다. **SAC 구현 1 및 비디오 2의 소스 코드는 온라인에서 확보할 수 있습니다.**

**1. github.com/haarnoja/sac**

**2. sites.google.com/view/soft-actor-critic**

**5.1. Comparative Evaluation**

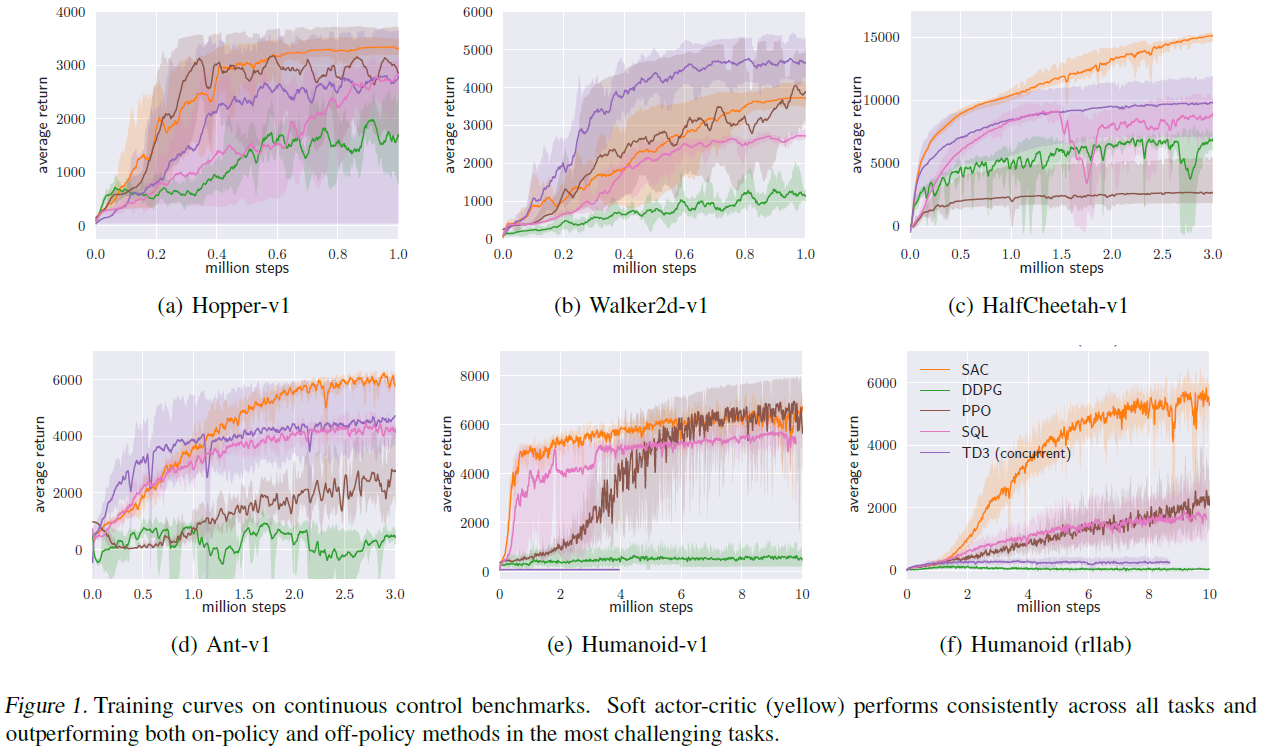


Figure 1 shows the total average return of evaluation rollouts during training for DDPG, PPO, and TD3. We train five different instances of each algorithm with different random seeds, with each performing one evaluation rollout every 1000 environment steps. The solid curves corresponds to the mean and the shaded region to the minimum and maximum

returns over the five trials.

그림 1은 DDPG, PPO 및 TD3에 대해 학습 동안의 평가 롤아웃의 총 평균 수익을 보여줍니다. 우리는 서로 다른 random 시드를 사용하여 각 알고리즘의 5개의 서로 다른 인스턴스를 학습시겼으며, 1000 environment steps(1000개의 환경 단계) 마다 하나의 평가 롤아웃을 각각 수행하였다.

실선은 평균에 해당하고 음영 영역은 5번의 시도의 최소 및 최대 returns이다.

The results show that, overall, SAC performs comparably to the baseline methods on the easier tasks and outperforms them on the harder tasks with a large margin, both in terms

of learning speed and the final performance. For example, DDPG fails to make any progress on Ant-v1, Humanoidv1, and Humanoid (rllab), a result that is corroborated by prior work (Gu et al., 2016; Duan et al., 2016).

결과는 전반적으로 SAC가 더 쉬운 작업에 대해 baseline methods(기준 방법)과 비슷한 수준으로 수행하고, 더 어려운 작업에 대해서는 두 가지 측면(학습 속도와 최종 성능)에서 큰 차이로 성능을 능가한다는 것을 보여줍니다.

예를 들어, DDPG는 Ant-v1, Humanoidv1 및 Humanoid(rllab)에서 어떠한 진전도 이루지 못했으며, 이는 이전 작업에 의해 확인된 결과입니다(Gu et al., 2016; Duan et al., 2016).

SAC also learns considerably faster than PPO as a consequence of the large batch sizes PPO needs to learn stably on more high-dimensional and complex tasks. Another maximum entropy RL algorithm, SQL, can also learn all tasks, but it is slower than SAC and has worse asymptotic performance.

SAC는 또한 PPO가 더 고차원적이고 복잡한 작업에서 안정적으로 학습해야 하는 큰 배치 크기의 결과로 PPO보다 훨씬 빠르게 학습합니다. 또 다른 최대 엔트로피 RL 알고리즘인 SQL도 모든 작업을 학습할 수 있지만 SAC 보다 느리고 점근적 성능이 더 나쁩니다.

The quantitative results attained by SAC in our experiments also compare very favorably to results reported by other methods in prior work (Duan et al., 2016; Gu et al., 2016; Henderson et al., 2017), indicating that both the sample efficiency and final performance of SAC on these benchmark tasks exceeds the state of the art. All hyperparameters used in this experiment for SAC are listed in Appendix D.

우리 실험에서 SAC가 얻은 정량적 결과는 이전 작업(Duan et al., 2016; Gu et al., 2016; Henderson et al., 2017)에서 다른 방법으로 보고된 결과와 매우 유리하게 비교되어 두 샘플 모두 이러한 벤치마크 작업에 대한 SAC의 효율성과 최종 성능은 최신 기술을 능가합니다. SAC에 대해 이 실험에 사용된 모든 하이퍼파라미터는 부록 D에 나열되어 있습니다.

**5.2. Ablation Study (절제 연구)**

The results in the previous section suggest that algorithms based on the maximum entropy principle can outperform conventional RL methods on challenging tasks such as the humanoid tasks. In this section, we further examine which particular components of SAC are important for good performance.

We also examine how sensitive SAC is to some of the most important hyperparameters, namely reward scaling and target value update smoothing constant.

이전 섹션의 결과는 최대 엔트로피 원칙에 기반한 알고리즘이 휴머노이드 작업과 같은 도전적인 작업에서 기존 RL 방법보다 성능이 우수할 수 있음을 시사합니다. 이 섹션에서는 SAC의 어떤 특정 구성 요소가 좋은 성능을 위해 중요한지 자세히 조사합니다.

또한 SAC가 가장 중요한 하이퍼파라미터, 즉, reward scaling and target value update smoothing constant(보상 규모 조정 및 목표 값 업데이트 평활 상수)에 얼마나 민감한지 조사합니다.

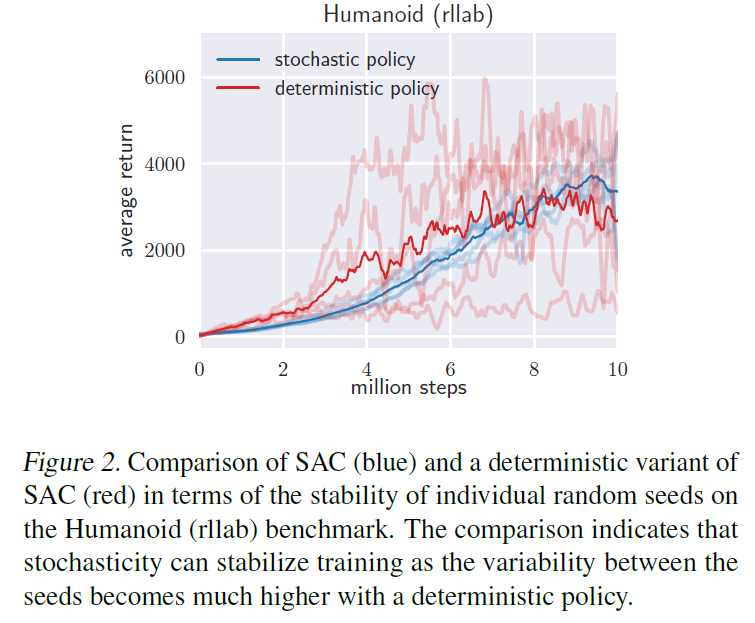
**Stochastic vs. deterministic policy.**

Soft actor-critic learns stochastic policies via a maximum entropy objective. The entropy appears in both the policy and value function. In the policy, it prevents premature convergence of the policy variance (Equation 10). In the value function, it encourages exploration by increasing the value of regions of state space that lead to high-entropy behavior (Equation 5).

Soft Actor-Critic은 최대 엔트로피 목표를 통해 확률적 정책을 학습합니다. 엔트로피는 정책 기능과 가치 기능 모두에 나타납니다. 정책에서 정책 분산의 조기 수렴을 방지합니다(식 10). 가치 함수에서는 높은 엔트로피 행동으로 이어지는 상태 공간 영역의 가치를 증가시켜 탐색을 장려합니다(식 5).

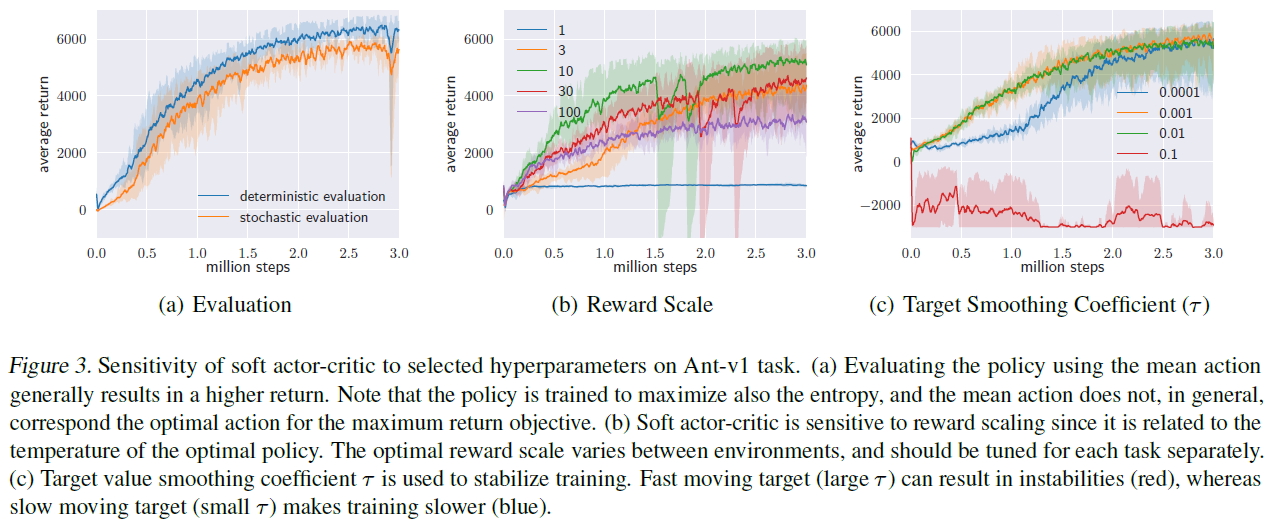
To compare how the stochasticity of the policy and entropy maximization affects the performance, we compare to a deterministic variant of SAC that does not maximize the entropy and that closely resembles DDPG, with the exception of having two Q-functions, using hard target updates, not having a separate target actor, and using fixed rather than learned exploration noise. Figure 2 compares five individual runs with both variants, initialized with different random seeds.

정책 및 엔트로피 최대화의 확률이 성능에 미치는 영향을 비교하기 위해 엔트로피를 최대화하지 않고 DDPG와 매우 유사한 SAC의 결정론적 변형을 비교합니다. 별도의 타겟 액터가 없고 학습된 탐색 노이즈가 아닌 고정된 노이즈를 사용합니다. 그림 2는 서로 다른 임의 시드로 초기화된 두 변이를 사용한 5개의 개별 실행을 비교합니다.



Soft actor-critic performs much more consistently, while the deterministic variant exhibits very high variability across seeds, indicating substantially worse stability. As evident from the figure, learning a stochastic policy with entropy maximization can drastically stabilize training. This becomes especially important with harder tasks, where tuning hyperparameters is challenging. In this comparison, we updated the target value network weights with hard updates, by periodically overwriting the target network parameters to match the current value network (see Appendix E for a comparison of average performance on all benchmark tasks).

Soft Actor-Critic은 훨씬 더 일관되게 수행하는 반면, 결정론적 변형은 시드 간에 매우 높은 변동성을 나타내므로 안정성이 상당히 나빠짐을 나타냅니다. 그림에서 알 수 있듯이 엔트로피를 최대화하여 확률적 정책을 학습하면 훈련을 대폭 안정화할 수 있습니다. 이는 하이퍼파라미터 조정이 어려운 어려운 작업에서 특히 중요합니다. 이 비교에서 우리는 현재 가치 네트워크와 일치하도록 타겟 네트워크 매개변수를 주기적으로 덮어써서 하드 업데이트로 타겟 가치 네트워크 가중치를 업데이트했습니다(모든 벤치마크 작업의 평균 성능 비교는 부록 E 참조).



**Policy evaluation.**

Since SAC converges to stochastic policies, it is often beneficial to make the final policy deterministic at the end for best performance. For evaluation, we approximate the maximum a posteriori action by choosing the mean of the policy distribution. Figure 3(a) compares training returns to evaluation returns obtained with this strategy indicating that deterministic evaluation can yield better performance. It should be noted that all of the training curves depict the sum of rewards, which is different from the objective optimized by SAC and other maximum entropy RL algorithms, including SQL and Trust-PCL, which maximize also the entropy of the policy.

SAC는 확률적 정책으로 수렴하기 때문에 최상의 성능을 위해 최종 정책을 최종적으로 결정적으로 만드는 것이 종종 유리합니다. 평가를 위해 정책 분포의 평균을 선택하여 최대 사후 조치를 근사합니다. 그림 3(a)는 훈련 수익을 이 전략으로 얻은 평가 수익과 비교하여 결정론적 평가가 더 나은 성과를 낼 수 있음을 나타냅니다. 모든 훈련 곡선은 보상의 합을 나타냅니다. 이는 SAC 및 정책의 엔트로피도 최대화하는 SQL 및 Trust-PCL을 포함한 기타 최대 엔트로피 RL 알고리즘에 의해 최적화된 목표와 다릅니다.

**Reward scale.**

Soft actor-critic is particularly sensitive to the scaling of the reward signal, because it serves the role of the temperature of the energy-based optimal policy and thus controls its stochasticity. Larger reward magnitudes correspond to lower entries. Figure 3(b) shows how learning performance changes when the reward scale is varied: For small reward magnitudes, the policy becomes nearly uniform, and consequently fails to exploit the reward signal, resulting in substantial degradation of performance.

Soft Actor-Critic은 에너지 기반 최적 정책의 온도 역할을 하여 확률성을 제어하기 때문에 보상 신호의 크기 조정에 특히 민감합니다. 더 큰 보상 규모는 더 낮은 항목에 해당합니다. 그림 3(b)는 보상 규모가 변할 때 학습 성능이 어떻게 변하는지 보여줍니다. 작은 보상 크기의 경우 정책이 거의 균일해지며 결과적으로 보상 신호를 활용하지 못하여 성능이 크게 저하됩니다.

For large reward magnitudes, the model learns quickly at first, but the policy then becomes nearly deterministic, leading to poor local minima due to lack of adequate exploration.

큰 보상 규모의 경우 모델은 처음에는 빠르게 학습하지만 이후 정책은 거의 결정적이 되어 적절한 탐색 부족으로 인해 로컬 최소값이 좋지 않게 됩니다.

With the right reward scaling, the model balances exploration and exploitation, leading to faster learning and better asymptotic performance. In practice, we found reward scale to be the only hyperparameter that requires tuning, and its natural interpretation as the inverse of the temperature in the maximum entropy framework provides good intuition for how to adjust this parameter.

올바른 보상 규모 조정을 통해 모델은 탐색과 활용의 균형을 유지하여 더 빠른 학습과 더 나은 점근적 성능으로 이어집니다. 실제로, 우리는 보상 규모가 튜닝이 필요한 유일한 하이퍼파라미터라는 것을 발견했으며, 최대 엔트로피 프레임워크에서 온도의 역수에 대한 자연스러운 해석은 이 파라미터를 조정하는 방법에 대한 좋은 직관을 제공합니다.

**Target network update.**

It is common to use a separate target value network that slowly tracks the actual value function to improve stability. We use an exponentially moving average, with a smoothing constant , to update the target value network weights as common in the prior work (Lillicrap et al., 2015; Mnih et al., 2015).

안정성을 향상시키기 위해 실제 가치 함수를 천천히 추적하는 별도의 목표 가치 네트워크를 사용하는 것이 일반적입니다. 우리는 평활 상수와 함께 지수 이동 평균을 사용하여 이전 작업에서 일반적으로 목표 값 네트워크 가중치를 업데이트합니다(Lillicrap et al., 2015; Mnih et al., 2015).

A value of one corresponds to a hard update where the weights are copied directly at every iteration and zero to not updating the target at all. In Figure 3(c), we compare the performance of SAC when varies. Large can lead to instabilities while small can make training slower. However, we found the range of suitable values of to be relatively wide and we used the same value (0.005) across all of the tasks.

값이 1이면 모든 반복에서 가중치가 직접 복사되는 하드 업데이트에 해당하고 대상을 전혀 업데이트하지 않는 경우 0입니다. 그림 3(c)에서는 SAC가 변할 때의 성능을 비교합니다. 크면 불안정해질 수 있고 작으면 훈련 속도가 느려질 수 있습니다. 그러나 의 적절한 값의 범위가 비교적 넓다는 것을 발견했으며 모든 작업에서 동일한 값(0.005)을 사용했습니다.

In Figure 4 (Appendix E) we also compare to another variant of SAC, where instead of using exponentially moving average, we copy over the current network weights directly into the target network every 1000 gradient steps. We found this variant to benefit from taking more than one gradient step between the environment steps, which can improve performance but also increases the computational cost.

그림 4(부록 E)에서는 지수 이동 평균을 사용하는 대신 현재 네트워크 가중치를 1000 그래디언트 단계마다 대상 네트워크에 직접 복사하는 SAC의 다른 변형과도 비교합니다. 우리는 이 변형이 환경 단계 사이에 둘 이상의 그래디언트 단계를 수행함으로써 이점을 발견했으며, 이는 성능을 향상시킬 수 있지만 계산 비용도 증가시킵니다.

**6. Conclusion**

We present soft actor-critic (SAC), an off-policy maximum entropy deep reinforcement learning algorithm that provides sample-efficient learning while retaining the benefits of entropy maximization and stability. Our theoretical results derive soft policy iteration, which we show to converge to the optimal policy. From this result, we can formulate a soft actor-critic algorithm, and we empirically show that it outperforms state-of-the-art model-free deep RL methods, including the off-policy DDPG algorithm and the on-policy PPO algorithm. In fact, the sample efficiency of this approach actually exceeds that of DDPG by a substantial margin.

엔트로피 최대화 및 안정성의 이점을 유지하면서 샘플 효율적인 학습을 제공하는 정책 외 최대 엔트로피 심층 강화 학습 알고리즘인 SAC(Soft Actor-Critic)를 제시합니다. 우리의 이론적 결과는 최적의 정책으로 수렴하는 것을 보여주는 부드러운 정책 반복을 도출합니다. 이 결과로부터 우리는 soft Actor-Critic 알고리즘을 공식화할 수 있으며, 이것이 off-policy DDPG 알고리즘 및 on-policy PPO 알고리즘을 포함하는 최첨단 모델 없는 deep RL 방법보다 성능이 우수함을 경험적으로 보여줍니다. 사실, 이 접근 방식의 샘플 효율성은 실제로 DDPG의 효율성을 상당히 초과합니다.

Our results suggest that stochastic, entropy maximizing reinforcement learning algorithms can provide a promising avenue for improved robustness and stability, and further exploration of maximum entropy methods, including methods that incorporate second order information (e.g., trust regions (Schulman et al., 2015)) or more expressive policy classes is an exciting avenue for future work.

우리의 결과는 확률론적 엔트로피 최대화 강화 학습 알고리즘이 2차 정보를 통합하는 방법(예: 신뢰 영역(Schulman et al., 2015))을 포함한 최대 엔트로피 방법의 추가 탐색 및 개선된 견고성과 안정성을 위한 유망한 방법을 제공할 수 있음을 시사합니다. ) 또는 보다 표현적인 정책 수업은 향후 작업을 위한 흥미로운 방법입니다.

**D. Hyperparameters**

Table 1 lists the common SAC parameters used in the comparative evaluation in Figure 1 and Figure 4.

**표 1은 그림 1과 그림 4의 비교 평가에 사용된 공통 SAC 매개변수를 나열**합니다.

테이블이(가) 표시된 사진

자동 생성된 설명

Table 2 lists the reward scale parameter that was tuned for each environment.

**표 2는 각 환경에 맞게 조정된 보상 척도 매개변수를 나열합니다.**

테이블이(가) 표시된 사진

자동 생성된 설명

**E. Additional Baseline Results**

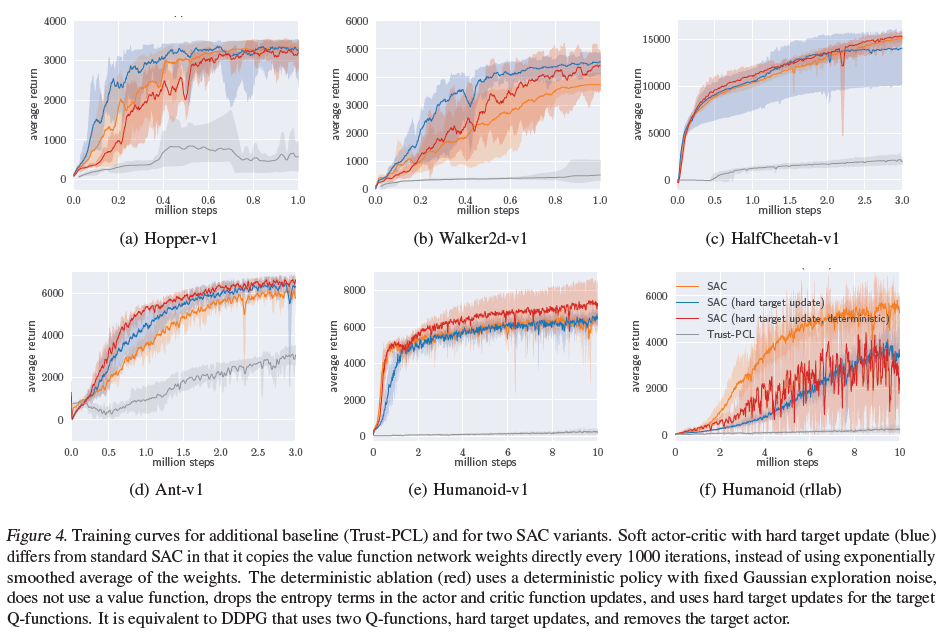


Figure 4 compares SAC to Trust-PCL (Figure 4. Trust-PC fails to solve most of the task within the given number of environment steps, although it can eventually solve the easier tasks (Nachum et al., 2017b) if ran longer. The figure also includes two variants of SAC: a variant that periodically copies the target value network weights directly instead of using exponentially moving average, and a deterministic ablation which assumes a deterministic policy in the value update (Equation 6) and the policy update (Equation 13), and thus strongly resembles DDPG with the exception of having two Q-functions, using hard target updates, not having a separate target actor, and using fixed exploration noise rather than learned. Both of these methods can learn all of the tasks and they perform comparably to SAC on all but Humanoid (rllab) task, on which SAC is the fastest.

그림 4는 SAC를 Trust-PCL과 비교합니다(그림 4. Trust-PC는 주어진 환경 단계 수 내에서 대부분의 작업을 해결하는 데 실패하지만 더 오래 실행하면 결국 더 쉬운 작업을 해결할 수 있습니다(Nachum et al., 2017b). 이 그림에는 SAC의 두 가지 변형도 포함되어 있습니다. 지수 이동 평균을 사용하는 대신 목표 값 네트워크 가중치를 직접 주기적으로 복사하는 변형과 값 업데이트(식 6) 및 정책 업데이트(식 13)에서 결정론적 정책을 가정하는 결정적 제거입니다. 따라서 두 개의 Q-함수가 있고, 하드 타겟 업데이트를 사용하고, 별도의 타겟 액터가 없고, 학습된 것보다 고정 탐색 노이즈를 사용한다는 점을 제외하고는 DDPG와 매우 유사합니다. 이 두 가지 방법 모두 모든 작업을 학습할 수 있으며 SAC가 가장 빠른 휴머노이드(rllab) 작업을 제외한 모든 작업에서 SAC와 비슷한 수준으로 수행합니다.