GENERALIZED MAXIMUM ENTROPY REINFORCEMENT LEARNING VIA REWARD SHAPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Entropy regularization is a commonly used technique in reinforcement learning to improve exploration and cultivate a better pre-trained policy for later adaptation. Recent studies further show that the use of entropy regularization can smooth the optimization landscape and simplify the policy optimization process, which indicates the value of integrating entropy into reinforcement learning. However, existing studies only consider the policy's entropy at the current state as an extra regularization term in the policy gradient or in the objective function, while the topic of integrating the entropy into the reward function has not been investigated. In this paper, we propose a shaped reward that includes the agent's policy entropy into the reward function. In particular, the agent's entropy at the *next state* is added to the immediate reward associated with the current state. The addition of the agent's policy entropy at the next state, instead of the policy entropy at the current state as used in the existing maximum entropy reinforcement learning framework, considers both state and action uncertainties. This distinguishes our work from the existing maximum entropy reinforcement learning framework via providing better action exploration and better control policies. We also show the addition of the agent's policy entropy at the next state yields new soft Q function and state value function that are concise and modular. Hence, the new reinforcement learning framework can be easily applied to the existing standard reinforcement learning algorithms while inheriting the benefits of employing entropy regularization. We further present a soft stochastic policy gradient theorem based on the shaped reward and propose a new practical reinforcement learning algorithm. Finally, a few experimental studies are conducted in the MuJoCo environment to demonstrate that our method can outperform the existing state-of-the-art reinforcement learning approaches.[1]

1 INTRODUCTION

Reinforcement learning (RL) (Sutton & Barto, 2018), one main research area in machine learning, deals with sequential decision-making problems, such as gaming (Silver et al., 2016; Berner et al., 2019), robotics manipulation (Kober et al., 2013; Johannink et al., 2019), marketing and advertising (Zhao et al., 2021). The advances of RL algorithms, especially deep RL algorithms, have shown the advantages of yielding human-level or better-than-human-level performance in, *e.g.*, Go and Atari games (Silver et al., 2016; Mnih et al., 2015). The continuous development of new RL theories as well as hardware can potentially play an important role in achieving human-level intelligence in the near future. The key research question for reinforcement learning is to find a control policy that maximizes the expectation of cumulative reward where the rewards are collected by the learning agent via interacting with the environment using the control policy. Since the cumulative reward acts as the metric for evaluating the performance of a control policy, the reward function that generates the immediate reward for the learning agent during the interacting process is an essential element in the algorithmic development of RL approaches.

The existing RL approaches can be mainly categorized into value-based, policy-based, and actor-critic methods. The value-based methods, such as Q-learning (Watkins & Dayan, 1992) and SARSA

[1] Source codes for this work can be found at https://github.com/ResearchSharedCode/Generalized-Maximum-Entropy-RL.

(Sutton & Barto, 2018), focus on learning the optimal value function and then deriving the control policy from the obtained value function via selecting the action with the best value. On the other hand, the policy-based methods, such as REINFORCE (Williams, 1992) and cross entropy method (Szita & Lörincz, 2006), focus on building a representation of the policy explicitly and optimizing its parameters through policy gradient (Sutton et al., 2000) or derivative free (Leonetti et al., 2012) techniques to find the optimal policy directly. A combination of the two fields is the actor-critic method (Konda & Tsitsiklis, 2000), which updates the control policy in an approximate gradient direction based on information provided by the value function. Although there are a variety of approaches for solving RL problems, the classical ones can hardly be used to solve tasks with large state and/or action spaces. Thanks to the power of deep neural networks and the improved computing capabilities such as graphics processing units (GPU), deep reinforcement learning, an integration of deep neural networks and RL, can potentially solve the decision making tasks with large state and action spaces. In particular, the neural network serves as a high-capacity function approximator and hence provides a superior tool for modeling the value function and the control policy. Important algorithms have been developed recently, such as DQN (Mnih et al., 2015), DDPG (Lillicrap et al., 2015), TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017), and A3C (Mnih et al., 2016). However, the important exploration-exploitation issue (Wang et al., 2019) has not been properly addressed in these methods. In other words, the learning agent may not explore the environment enough to yield stable and good control policies, especially in the hard-exploration (Ecoffet et al., 2019) problems.

To address the exploration issue, several techniques have been proposed. For example, the classical epsilon-greedy technique forces the learning agents to select random actions with probability ϵ and execute the optimal actions with the probability of $1 - \epsilon$. Hence, the learning agents have a probability of ϵ to randomly explore the action space. Another technique is Boltzmann exploration, which represents the policy as a Boltzmann distribution. Consequently, the stochastic behaviors will encourage the agent to explore the environment. To the best of our knowledge, the existing strategies mainly focus on solving the exploration problem in a utilitarian way with limited efforts on creating an exploration-driven deep reinforcement learning framework. To fill in this gap, we seek to propose a generalized reinforcement learning structure that enables the learning agent to be driven by "curiosity". It is worth mentioning that an exploration strategy adopted in (Zheng et al., 2018) shares a similar idea. In particular, the strategy, inspired by the work in psychology (Oudeyer & Kaplan, 2008), is to augment the environment reward (extrinsic reward) with an additional bonus signal (intrinsic reward). As the strategy involves the process of designing a reward function, its scope is different from this paper.

In this paper, we focus on developing a new exploration-driven deep reinforcement learning approach based on the entropy regularization. In particular, we reshape the reward function (Ng et al., 1999) by including the policy's entropy *at the next state* into the immediate reward of the current state, where the policy's entropy is considered as the intrinsic reward. We name this new approach "generalized maximum entropy reinforcement learning". Different from the existing maximum entropy reinforcement learning that includes the policy's entropy at the current state as a regularizer that only considers action uncertainty, the proposed approach considers both state and action uncertainties. The process of encoding the state uncertainty into the entropy value can provide better exploration because the action entropy at the current state is computed based on a known state without state uncertainty. In other words, adding the policy's entropy at the next state directly connects the future policy's entropy subject to both state and action uncertainties with the policy improvement. Hence, our approach can provide better exploration. Moreover, the shaped reward function has three additional benefits: (1) a more concise definition of both soft Q-function and state value function than the ones from the existing maximum entropy reinforcement learning framework, (2) a general structure that is applicable to various reinforcement learning algorithms for better exploration, and (3) a simpler estimation of the policy gradient because there is no need to train a soft value network. To further demonstrate the modularity of the shaped reward, we present a soft stochastic policy gradient theorem based on the new reward and propose a simple and practical off-policy soft stochastic policy gradient algorithm. Finally, we compare the new algorithm with prior state-of-the-art RL approaches in several Mujuco environments. The results show our algorithm can outperform the existing algorithm in most simulation environments, which illustrates the effectiveness of the proposed exploration-driven method.

2 RELATED WORK

Entropy regularization is a commonly used technique in reinforcement learning to improve exploration, cultivate a better pre-trained policy for later adaptation (Haarnoja et al., 2017), and stay robust in the face of adversarial perturbations (Ziebart, 2010). A recent study (Ahmed et al., 2019) has further shown that the entropy can smooth the optimization and simplify the policy optimization process, which indicates the extra benefit of considering the policy's entropy in reinforcement learning. In the A3C algorithm (Mnih et al., 2016), the authors found that adding the policy's entropy to the objective function can improve exploration. However, the entropy term is only employed to regularize the policy gradient with the value function remaining the same as defined in the standard reinforcement learning. Meanwhile, a maximum entropy reinforcement learning framework was formulated when considering the policy's entropy term in the value functions.

Recently, a few approaches have been proposed based on the aforementioned framework. Among them, SAC (Haarnoja et al., 2018) is one of the notable methods. In particular, SAC defined its state value function as the Q-function plus the policy's entropy at the same state, where the Q-function can be computed iteratively using the Bellman backup operator. With the calculated value functions, the authors then proposed the policy improvement step by minimizing Kullback-Leibler divergence between the current policy and the policy expressed in an energy-based model. SAC has been proved to be a successful deep RL method. However, it is not straightforward for applications with discrete action. In addition, SAC does not need to calculate and include entropy in the Q-function while our approach directly includes entropy in the Q-function. In contrast to SAC, the estimation of our policy gradient is simpler because we do not have to train a soft value network. Another relevant work is reported in (Shi et al., 2019), which follows the same problem setting but updates the policy through policy gradient. The gradient can be simply derived by taking the derivative of both the Q-function and entropy regularizer with respect to the policy parameters. However, estimating the Q-function is a key issue. Specifically, an extra target policy is required for calculating the entropy regularizer when estimating the Q-function. Moreover, the policy gradient proposition can hardly be generalized to the standard reinforcement learning approaches.

3 PRELIMINARIES

In this section, we will briefly review the standard reinforcement learning and maximum entropy reinforcement learning frameworks.

3.1 STANDARD REINFORCEMENT LEARNING

The goal of reinforcement learning is to find a policy π that maximizes the expectation of discounted cumulative reward through trial and error (Sutton & Barto, 2018). The learning process itself can be modeled as an agent interacting with the environment modeled by a Markov decision process $\mathcal{M}$, which is tuple of $< S, A, T, R, \gamma >$. In particular, S is the set of states. A is the set of actions. T is a state transition function specifying, for each state, action, and next state, the probability that the next state occurs. R is the reward function that describes the reward associated with a state or a state-action pair. γ is the discount factor. Here, we assume that the environment reward function $r \in R$ is determined by the current state $s \in S$ and action $a \in A$, *i.e.*, $r_t = r(s_t, a_t)$, where t is the time step. Accordingly, the standard reinforcement learning problem can be formulated as

$$\pi = \arg\max_{\pi} \sum_{t=0}^{\infty} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\Big[\gamma^t r(s_t, a_t)\Big], \tag{1}$$

where $a_t = \pi(a_t|s_t)$ and ρ_π is state-action marginal distribution given a policy π. The collected rewards can formulate value functions, which are critical components in the optimal policy searching process, especially for deep reinforcement learning. In particular, the value functions can either generate optimal policy directly or supervise the policy learning step. Their definitions, *i.e.*, state-action value $Q^\pi(s_t, a_t)$ and the state value $V^\pi(s_t)$, are given as follows

$$\begin{aligned} Q^\pi(s_t, a_t) &= r(s_t, a_t) + \sum_{i>t} \gamma^{i-t} \mathbb{E}_{(s_i,a_i)\sim\rho_\pi}\big[r(s_i, a_i)\big] \\ V^\pi(s_t) &= \mathbb{E}_{a_t\sim\pi(a_t|s_t)} Q^\pi(s_t, a_t) \end{aligned}. \tag{2}$$

The Q-function and state value function defined in equation 2 satisfy the Bellman backup equation in equation 3, which is a useful tool for estimating the value functions.

$$Q^{\pi}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)} V^{\pi}(s_{t+1}). \tag{3}$$

The standard reinforcement learning framework provides a fundamental platform for the development of various RL algorithms. However, the learning agent under this setting may encounter exploration issues. In other words, the corresponding RL algorithms are often not robust, especially in the presence of model and estimation errors.

3.2 Maximum entropy reinforcement learning

To address the exploration issue and improve robustness, a new type of reinforcement learning algorithm, namely, maximum entropy reinforcement learning, has been developed. The goal of maximum entropy reinforcement learning is to maximize the expectation of discounted cumulative reward augmented with an entropy regularized term, given by

$$\pi = \arg\max_{\pi} \sum_{t=0}^{\infty} \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi}} \Big[\gamma^t \big(r(s_t, a_t) + \alpha \mathcal{H}(\cdot|s_t) \big) \Big], \tag{4}$$

where $\mathcal{H}(\cdot|s_t) = -\mathbb{E}_{a_t \sim \pi(a_t|s_t)} \log \pi(a_t|s_t)$ and α is the temperature parameter that controls the degree of entropy regularization. Under the maximum entropy reinforcement learning setting, the learning agent aims to succeed at the task while acting as randomly as possible, which encourages exploration and helps prevent early convergence to sub-optimal policies. The Q-function here is redefined as soft Q-function given by (Haarnoja et al., 2018)

$$\widetilde{Q}^{\pi}(s_t, a_t) = r(s_t, a_t) + \sum_{i>t} \gamma^{i-t} \mathbb{E}_{(s_i, a_i) \sim \rho_{\pi}} \big[r(s_i, a_i) + \alpha \mathcal{H}(\cdot|s_i) \big]. \tag{5}$$

To ensure that a similar Bellman backup equation as equation 3 holds under the maximum entropy reinforcement learning setting, the new state value, named as soft state value, is defined as

$$\widetilde{V}^{\pi}(s_t) = \mathbb{E}_{a_t \sim \pi(a_t|s_t)} \Big[\widetilde{Q}^{\pi}(s_t, a_t) - \alpha \log \pi(a_t|s_t) \Big]. \tag{6}$$

With the redefined Q-function and state value function, different RL approaches, *e.g.*, SQL (Haarnoja et al., 2017) and SAC (Haarnoja et al., 2018), have been developed. However, the definitions of value functions are more complicated than the ones in standard reinforcement learning algorithms. Moreover, the previous RL procedures and theories are not applicable in the maximum entropy RL setting. Hence, it is important to create a new RL setting that can not only comply with the existing RL algorithms but also inherit the advantages of the maximum entropy reinforcement learning. To address this need, we will provide the formulation of a new generalized maximum entropy reinforcement learning setting in the next section.

4 Generalized maximum entropy reinforcement learning

We consider a new framework by including the policy entropy term at the next state in the environment reward function, *i.e.*,

$$r^{\pi}(s_t, a_t) = r(s_t, a_t) + \alpha \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)} \big[\mathcal{H}(\cdot|s_{t+1}) \big]. \tag{7}$$

Note that we enclose the policy's entropy at the next state here, which is different from the previous entropy regularization that considers its entropy at the current state. Intuitively, we believe that maximizing the policy's entropy at the next state should achieve better exploration than maximizing the entropy at the current state because the current one has already been visited. In addition, the entropy term is not used as a regularizer but an intrinsic reward. Hence, the value functions under this framework can be directly obtained from the standard RL framework. In other words, the new framework is a more generalized version of maximum entropy reinforcement learning, which keeps the maximum entropy properties and is also applicable to various RL algorithms. We name the new framework as generalized maximum entropy reinforcement learning. Hence, we formulate the new generalized maximum entropy reinforcement learning as

$$\pi = \arg\max_{\pi} \sum_{t=0}^{\infty} \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \beta^{\pi}} \Big[\gamma^t r^{\pi}(s_t, a_t) \Big], \tag{8}$$

where β^π is the state-action-state marginal distribution given a policy π. Based on the new framework, the structure of the soft Q-function and soft state value function are given by

$$\begin{aligned} Q^\pi_{\text{soft}}(s_t, a_t) &= r^\pi(s_t, a_t) + \sum_{i>t} \gamma^{i-t} \mathbb{E}_{(s_i, a_i, s_{i+1}) \sim \beta_\pi} \left[r^\pi(s_i, a_i) \right] \\ V^\pi_{\text{soft}}(s_t) &= \mathbb{E}_{a_t \sim \pi(a_t|s_t)} Q^\pi_{\text{soft}}(s_t, a_t). \end{aligned} \tag{9}$$

Hence, the Bellman backup equation in equation 3 is also satisfied, namely,

$$Q^\pi_{\text{soft}}(s_t, a_t) = r^\pi(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)} V^\pi_{\text{soft}}(s_{t+1}). \tag{10}$$

To further demonstrate the modularity of the new value functions, we next present a generalized soft policy gradient theorem.

A typical way to solve reinforcement learning is the policy gradient method (Sutton et al., 2000). In particular, the policy gradient method targets at modeling and optimizing the policy directly, where the policy π is modeled as a parameterized function with respect to θ. Without loss of generality, let the loss function for the policy under the generalized maximum entropy reinforcement learning setting be denoted as $J(\theta)$, *i.e.*,

$$J(\theta) = \sum_{t=0}^{\infty} \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \beta^{\pi_\theta}} \left[\gamma^t r^{\pi_\theta}(s_t, a_t) \right], \tag{11}$$

we have the following theorem regarding the gradient of the loss function.

Theorem 4.1. *Assume that the policy $\pi_\theta(a|s)$ is differentiable with respect to its parameter θ. The gradient $\nabla_\theta J(\theta)$ can be computed as*

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{(s,a,s') \sim \beta^{\pi_\theta}} \left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}_{\text{soft}}(s, a) + \alpha \nabla_\theta \mathcal{H}(\cdot|s') \right]. \tag{12}$$

Proof. The proof is given in Appendix A.1. □

5 Soft Stochastic Policy Gradient Algorithm

In this section, we will present a practical deep RL algorithm named as soft stochastic policy gradient (SSPG) approach, which is derived based on the generalized maximum entropy reinforcement learning setting described in Section 4.

5.1 Soft State Value Approximation

Based on equation 9, the soft state value function can be approximated by the sampled soft Q-function, *i.e.*,

$$V^{\pi_\theta}_{\text{soft}}(s_t) \approx \frac{1}{N} \sum_{i=1}^{N} Q^{\pi_\theta}_{\text{soft}}(s_t, a^i_t), \tag{13}$$

where N is the number of sampled actions in the current policy distribution $\pi_\theta(a_t|s_t)$. Hence, the policy π_θ can be trained by maximizing the soft state value given by

$$J_V(\theta) = \mathbb{E}_{s_t \sim d^{\pi_\theta}(s_t)} V^{\pi_\theta}_{\text{soft}}(s_t) \approx \mathbb{E}_{s_t \sim d^{\pi_\theta}(s_t)} \frac{1}{N} \sum_{i=1}^{N} Q^{\pi_\theta}_{\text{soft}}(s_t, \pi_\theta(a^i_t|s_t)). \tag{14}$$

5.2 Soft Q-function Approximation

The soft Q-function $Q^{\pi_\theta}_{\text{soft}}(s_t, a_t)$ can be represented by one function approximator. In particular, we model it by a neural network parameterized with ϕ for large continuous environments, which is denoted as $Q^{\pi_\theta}_\phi(s_t, a_t)$. The soft Q-function parameters can be trained to minimize the soft Bellman

residual given by

$$
\begin{aligned}
J_Q(\phi) &= \mathbb{E}_{(s_t,a_t)\sim\rho^{\pi_\theta}}\Big[\frac{1}{2}\big(Q_\phi^{\pi_\theta}(s_t,a_t) - r^{\pi_\theta}(s_t,a_t) - \gamma\mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)}V_{\text{soft}}^{\pi_\theta}(s_{t+1})\big)^2\Big] \\
&= \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\beta^{\pi_\theta}}\Big[\frac{1}{2}\big(Q_\phi^{\pi_\theta}(s_t,a_t) - r^{\pi_\theta}(s_t,a_t) - \gamma V_{\text{soft}}^{\pi_\theta}(s_{t+1})\big)^2\Big] \\
&\approx \mathbb{E}_{(s_t,a_t,s_{t+1})\sim M}\Big[\frac{1}{2}\big(Q_\phi^{\pi_\theta}(s_t,a_t) - r^{\pi_\theta}(s_t,a_t) - \gamma\frac{1}{N}\sum_{i=1}^{N}Q_\phi^{\pi_\theta}(s_{t+1},a^i)\big)^2\Big] \\
&= \mathbb{E}_{(s_t,a_t,s_{t+1})\sim M}\Big[\frac{1}{2}\big(Q_\phi^{\pi_\theta}(s_t,a_t) - r(s_t,a_t) - \alpha\mathcal{H}(\cdot|s_{t+1}) - \gamma\frac{1}{N}\sum_{i=1}^{N}Q_\phi^{\pi_\theta}(s_{t+1},a^i)\big)^2\Big],
\end{aligned}
\tag{15}
$$

where M is the replay memory that stores history transitions and $a^i \sim \pi_\theta(\cdot|s_{t+1})$ is sampled from the current policy π_θ. The goal of optimizing equation 15 is to find a parameter ϕ such that $Q_\phi^{\pi_\theta}(s_t,a_t)$ matches $r(s_t,a_t) + \alpha\mathcal{H}(\cdot|s_{t+1}) + \gamma\frac{1}{N}\sum_{i=1}^{N}Q_\phi^{\pi_\theta}(s_{t+1},a^i)$. Notice that the entropy $\mathcal{H}(\cdot|s_{t+1})$ is purely determined by the current policy π_θ. To keep the gradient information of the entropy term for the optimization process in equation 14, we propose to use $Q_\phi^{\pi_\theta}(s_t,a_t)$ to represent the standard Q-function, *i.e.*,

$$
\tilde{J}_Q(\phi) = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim M}\Big[\frac{1}{2}\big(Q_\phi^{\pi_\theta}(s_t,a_t) - r(s_t,a_t) - \gamma\frac{1}{N}\sum_{i=1}^{N}Q_{\bar{\phi}}^{\pi_\theta}(s_{t+1},a^i)\big)^2\Big], \tag{16}
$$

where $\bar{\phi}$ is the parameter for the target network $Q_{\bar{\phi}}^{\pi_\theta}(s_t,a_t)$ and this target network is used to stabilize the Q-function learning process. The soft Q-function can then be approximated by

$$
Q_{\text{soft}}^{\pi_\theta}(s_t,a_t) = Q_\phi^{\pi_\theta}(s_t,a_t) + \alpha\mathbb{E}_{s_{t+1}\sim M}\sum_{i=t}^{\infty}\gamma^{i-t}\mathcal{H}(\cdot|s_{i+1}). \tag{17}
$$

5.3 Pseudocode

The pseudocode of SSPG is provided in Algorithm 1. In particular, the proposed approach saves the history transitions and uses it to update the neural network parameter ϕ. The current policy is then updated based on the policy gradient, which is calculated from the soft Q-function. To improve exploration at the start of training, the agent takes actions sampled from a uniform random distribution over valid action space when the number of running steps is below the start steps. Afterwards, the action sampling procedure is conducted by its own policy distribution.

6 Experiments

In this section, we will compare our method with several prior reinforcement learning algorithms on a range of challenging continuous control tasks in the MuJoCo environment (Todorov et al., 2012). In particular, we compare our method with (1) trust region policy optimization (TRPO) (Schulman et al., 2015), (2) proximal policy optimization (PPO) (Schulman et al., 2017), and (3) soft actor critic (SAC) (Haarnoja et al., 2018). We also extract the existing results from (Shi et al., 2019) for deep soft policy gradient (DSPG) method and deep deterministic policy gradient (DDPG), and results from (Wang & Ba, 2019) for model-based policy planning methods in action space (POPLIN-A), model-based policy planning methods in parameter space (POPLIN-P) methods, and twin delayed deep deterministic policy (TD3), respectively. The extracted results are compared with those from our method in a tabular form.

6.1 Setup

In the implementation of our SSPG method, we construct two neural networks to represent the soft Q-function and the policy, respectively. To stabilize the soft Q-function learning process, a target network is created and updated based on the learned parameter ϕ. We also adopt the idea of

Algorithm 1 Soft Stochastic Policy Gradient (SSPG) Algorithm

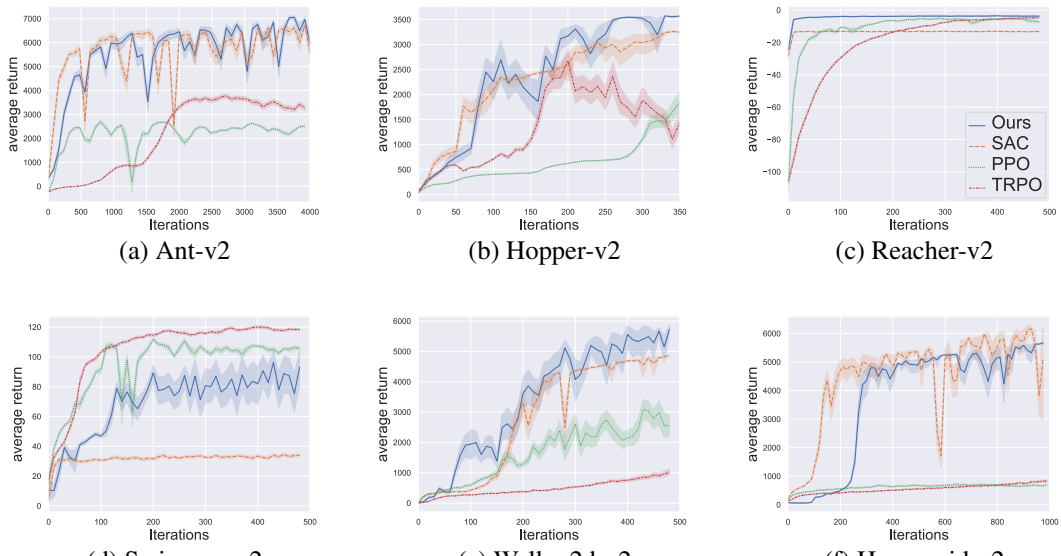
Initialize parameter vectors θ, ϕ,
Initialize the target network $Q_{\bar{\phi}}^{\pi_\theta}(s_t, a_t)$ with $\bar{\phi} \leftarrow \phi$
Initialize replay buffer M
Set batch_size, start_steps, target_update_interval, N, λ
numsteps = 0, updates = 0
for each iteration **do**
    Reset initial observation state s_0
    for each step **do**
        if numsteps > start_steps **then**
            Sample an action a_t: $a_t \sim \pi_\theta(\cdot|s_t)$
        else
            Sample a_t from a uniform distribution over valid actions
        Execute a_t and collect $r(s_t, a_t), s_{t+1}$
        Store transition $(s_t, a_t, r(s_t, a_t), s_{t+1})$ in M
        numsteps += 1
        if len(M) > batch_size **then**
            Sample batch_size transitions $(s_t, a_t, r(s_t, a_t), s_{t+1})$ from M
            Sample N actions a^i from $\pi_\theta(\cdot)$ for each state s_{t+1}
            Compute the critic loss $\tilde{J}_Q(\phi)$ in Eq. equation 16
            Update the critic by minimizing the loss
            Calculate the entropy $\mathcal{H}(\cdot|s_{t+1})$ based on $\pi_\theta(\cdot|s_{t+1})$
            Compute the actor loss in Eq. equation 14
            Update the actor by maximizing the loss
            updates += 1
        if updates $\mod$ target_update_interval = 0 **then**
            Update the target network: $\bar{\phi} \leftarrow \lambda\phi + (1-\lambda)\bar{\phi}$

double Q-learning, which is proved in (Hasselt, 2010) that performance can be improved in most environments. Meanwhile, the policy is represented by a multivariate Gaussian so that the entropy can be easily calculated regardless of the action space as $1/2 \ln \det(2\pi e\Sigma)$, where $\Sigma \in \mathbb{R}^{k\times k}$ is the covariance. Since the policy entropy is determined by the covariance only, the output of our neural network that models the policy should include both the location $\mu \in \mathbb{R}^k$ and covariance Σ. Note that this may be different from the standard setting for policy's neural networks, where the covariance is often fixed to some value. The hyperparameters of our algorithm are listed in Appendix A.2.

6.2 Results

Figure 1 shows the average return of the training rollouts for SAC, PPO, TRPO, and our method. The solid/dash curves correspond to the mean and the shaded region to the standard deviation of the returns over five trials with different random seeds. More specifically, we compare the performances among six different environments, *i.e.*, Ant-v2, Hopper-v2, Reacher-v2, Swimmer-v2, Walker2d-v2, and Humanoid-v2, where the environment specific parameters are provided in Appendix A.3.

We can see from the results that our method outperforms all baselines on the tasks Hopper-v2, Reacher-v2, and Walker2d-v2 and performs slightly better than SAC on the task (Ant-v2), both in terms of learning speed and the final performance. For the high dimensional task of Humanoid-v2, our method achieves comparably but much more stable results than SAC. The swimmer-v2 is unsolvable for all tested algorithms given that the reward threshold for solving is around 340 for the original swimmer-v1. As analyzed in (Wang & Ba, 2019), the velocity sensor is on the neck of the swimmer, which makes the swimmer extremely prone to local-minimum. It is of particular importance to emphasize that the proposed SSPG method performs comparably or outperforms SAC in all environments, which shows that the new generalized maximum entropy reinforcement learning framework can provide better exploration than the existing maximum entropy reinforcement learning framework to obtain good control policies.



(a) Ant-v2 (b) Hopper-v2 (c) Reacher-v2

(d) Swimmer-v2 (e) Walker2d-v2 (f) Humanoid-v2

Figure 1: Training curves on continuous control benchmarks. Our method (solid blue curve) performs consistently across all tasks and outperforming both on-policy and off-policy methods in the most challenging tasks.

6.3 Final performance on total rewards

To further compare with other state-of-the-art methods, we also extract the results from (Shi et al., 2019) and (Wang & Ba, 2019). In particular, the results on DSPG, DDPG, POPLIN-P, POPLIN-A, and TD3 are collected and presented in the Table 1. From the comparison, we can see that our method can achieve a comparably better final performance than other methods. In particular, our method outperforms the DSPG in all environments with a fairly large margin, which shows the advantage of the proposed approach to obtain better control policies via more exploration. Moreover, our methods can beat the model-based methods in most environments, further verifying that our method can provide a high degree of exploration to avoid sub-optimal policies.

Table 1: The results for the average total rewards in the last episode training.

Methods	HalfCheetah-v2	Ant-v2	Hopper-v2	Walker2d-v2	Swimmer-v2
Ours	9297.1±457.8	**6768.5±1107.5**	**3555.4±22.7**	**5722.9±324.9**	78.7±27.1
DSPG†	< 7500	< 3000	< 3000	< 5500	N/A
DDPG†	< 1000	< 1000	< 1500	< 1000	N/A
POPLIN-P	**12227.9 ± 5652.8**	2330.1±320.9	2055.2±613.8	597.0±478.8	**334.4±34.2***
POPLIN-A	4651.1± 1088.5	1148.4±438.3	202.5±962.5	-105.0±249.8	**344.9±7.1***
TD3	218.9± 593.9	870.1±283.8	1816.6±994.8	-516.4±812.2	72.1±130.9

† results obtained from Figure 1 in (Shi et al., 2019), where only the mean values are available.
* obtained with a modified swimmer by moving the sensor from the neck to the head.

7 Conclusion and future work

In this paper, we proposed to include the next state policy entropy in the current state's immediate reward to shape the reward and then formulated a generalized maximum entropy reinforcement learning framework. The new generalized maximum entropy reinforcement learning framework can provide a more concise definition of soft Q-function and state value function. Moreover, it can be easily applicable to the existing standard reinforcement learning algorithms. To demonstrate its modularity, a soft stochastic policy gradient theorem was derived based on the new setting. We further proposed a practical soft stochastic policy gradient algorithm and compared its performance

with other state-of-the-art RL approaches. The comparison shows that the proposed method can consistently outperform the existing methods.

Our future work includes (1) further implementation of the general soft policy gradient for different policy gradient algorithms and performance comparison with other methods, (2) evaluation of the proposed algorithms on physical robots, and (3) investigation of employing other regularization terms (Yang et al., 2019) rather than the Shannon entropy as the intrinsic reward.

REFERENCES

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International Conference on Machine Learning*, pp. 151–160, 2019.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Hado Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems*, pp. 2613–2621, 2010.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *International Conference on Robotics and Automation*, pp. 6023–6029, 2019.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pp. 1008–1014, 2000.

Matteo Leonetti, Petar Kormushev, and Simone Sagratella. Combining local and global direct derivative-free optimization for reinforcement learning. *International Journal of Cybernetics and Information Technologies*, 12, 2012.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1928–1937, 2016.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, pp. 278–287, 1999.

Pierre-Yves Oudeyer and Frederic Kaplan. How can we define intrinsic motivation? In *the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, Lund: LUCS, Brighton, 2008.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Wenjie Shi, Shiji Song, and Cheng Wu. Soft policy gradient method for maximum entropy deep reinforcement learning. *arXiv preprint arXiv:1909.03198*, 2019.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

István Szita and András Lörincz. Learning tetris using the noisy cross-entropy method. *Neural Computation*, 18(12):2936–2941, 2006.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Exploration versus exploitation in reinforcement learning: a stochastic control approach. *Available at SSRN 3316387*, 2019.

Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Wenhao Yang, Xiang Li, and Zhihua Zhang. A regularized approach to sparse optimal policy in reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 5940–5950, 2019.

Xiangyu Zhao, Changsheng Gu, Haoshenglun Zhang, Xiwang Yang, Xiaobing Liu, Hui Liu, and Jiliang Tang. DEAR: Deep reinforcement learning for online advertising impression in recommender systems. 2021.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In *Advances in Neural Information Processing Systems*, pp. 4644–4654, 2018.

Brian D Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010.