Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

Generalized Tsallis Entropy Reinforcement Learning and Its Application to Soft Mobile Robots

Kyungjae Lee[1], Sungyub Kim[2], Sungbin Lim[3], Sungjoon Choi[1], Mineui Hong[1], Jaein Kim[4], Yong-Lae Park[4] and Songhwai Oh[1]
[1]Dept. of Electrical and Computer Engineering, ASRI, Seoul National University,
[2]Graduate School of AI, Korea Advanced Institute of Science and Technology (KAIST),
[3]Dept. of Industrial Engineering, Ulsan National Institute of Science and Technology (UNIST),
[4]Dept. of Mechanical and Aerospace Engineering, Seoul National University
Email: {kyungjae.lee, mineui.hong, sungjoon.choi}@rllab.snu.ac.kr, sungyub.kim@kaist.ac.kr, sungbin@unist.ac.kr, {snu08mae, ylpark, songhwai}@snu.ac.kr

***Abstract*—In this paper, we present a new class of entropy-regularized Markov decision processes (MDPs), which will be referred to as Tsallis MDPs. that inherently generalize well-known maximum entropy reinforcement learning (RL) by introducing an additional real-valued parameter called an entropic index. Our theoretical result enables us to derive and analyze different types of optimal policies with interesting properties relate to the stochasticity of the optimal policy by controlling the entropic index. To handle complex and model-free problems, such as learning a controller for a soft mobile robot, we propose a Tsallis actor-critic (TAC) method. We first observe that different RL problems have different desirable entropic indices where using proper entropic index results in superior performance compared to the state-of-the-art actor-critic methods. To mitigate the exhaustive search of the entropic index, we propose a quick-and-dirty curriculum method of gradually increasing the entropic index which will be referred to as TAC with Curricula (TAC^2). TAC^2 shows comparable performance to TAC with the optimal entropic index. Finally, We apply TAC^2 to learn a controller of a soft mobile robot where TAC^2 outperforms existing actor-critic methods in terms of both convergence speed and utility.**

I. Introduction

Soft mobile robots have the potential to overcome challenging navigation tasks that conventional rigid robots are hard to achieve, such as exploring complex and unstructured environments, by using their high adaptability and robustness against changes around them [18]. Especially, a soft mobile robot using pneumatic actuators, which provide relatively high force-to-weight ratios, have been widely developed [26, 17]. Despite the fact that the pneumatic actuators combined with soft materials are beneficial to the adaptability and robustness of soft mobile robots, their behaviors are often hard to be modeled or controlled using a traditional method such as a feedback control [31], due to their inherent stochasticity.

To handle the absence of a dynamic model, some researches have employed a model-free reinforcement learning (RL) that does not require prior knowledge of dynamics [31, 33, 15, 16]. A model-free RL algorithm aims to learn a policy to effectively perform a given task through the trial and error without the prior knowledge about the environment, such as the dynamics of a soft robot, where the performance of policy is often measured by the sum of rewards. The absence of environmental information gives rise to an innate trade-off between exploration and exploitation during a learning process. If the algorithm decides to explore the environment, then, it will lose the chance to exploit the best decision based on collected experiences and vice versa. Such trade-off should be appropriately scheduled to learn an optimal policy through a small number of interactions with an environment. Especially, the efficiency of exploration becomes more important when training a soft mobile robot, as the properties of soft material can be changed or degraded if a robot exceeds its durability.

In this paper, we present a generalized framework for entropy-regularized RL problems with various types of entropy. The proposed framework is formulated as a new class of Markov decision processes with Tsallis entropy maximization, which is called Tsallis MDPs. The Tsallis entropy inherently generalizes a class of entropies, including the standard Shannon-Gibbs (SG) entropy by controlling a parameter, called an *entropic index* and Tsallis MDP introduces a unifying view on the use of various entropies in RL. We provide a comprehensive analysis of how different entropic indices in Tsallis MDPs result in different types of optimal policies and Bellman optimality equations.

Our theoretical results allow us to interpret the consequences of different types of entropy regularizations in RL. Specifically, different optimal policies resulting from entropic indices provide different exploration-exploitation trade-off behaviors as the entropic index affects the stochasticity of the corresponding optimal policy. This feature is often highly desirable in practice as sample complexity is highly affected by the exploration-exploitation trade-off and we could provide a systematic control over the trade-off by controlling the entropic index.

We empirically show that there exists an appropriate entropic index for each task and solving TAC with a proper entropic index outperforms existing actor-critic methods. Furthermore, we also propose quick-and-dirty curriculum learning of gradually increasing the entropic index to alleviate the demand of an exhaustive search of a suitable entropic index, which we call TAC with Curricular (TAC^2). We demonstrate that the proposed TAC^2 outperforms existing methods and even achieves a comparable performance compared to TAC with the optimal entropic index found from the exhaustive search

in terms of both utility and sample-efficiency. This superior sample-efficiency allows us to successfully learn the controller of the soft mobile robot within moderate time.

II. Background

A. Markov Decision Processes

A Markov decision process (MDP) is defined as a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, d, P, \gamma, \mathbf{r}\}$, where $\mathcal{S}$ is the state space, $\mathcal{F}$ is the corresponding feature space, $\mathcal{A}$ is the action space, $d(s)$ is the distribution of an initial state, $P(s'|s,a)$ is the transition probability from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ by taking $a \in \mathcal{A}$, $\gamma \in (0,1)$ is a discount factor, and $\mathbf{r}$ is the reward function defined as $\mathbf{r}(s,a,s') \triangleq \mathbb{E}[\mathbf{R}|s,a,s']$ with a random reward $\mathbf{R}$. In our paper, we assume that $\mathbf{r}$ is bounded. Then, the MDP problem can be formulated as: $\max_{\pi\in\Pi} \mathbb{E}_{\tau\sim P,\pi}\left[\sum_t^\infty \gamma^t \mathbf{R}_t\right]$, where $\sum_t^\infty \gamma^t \mathbf{R}_t$ is a discounted sum of rewards, also called a return, $\Pi = \{\pi | \forall s, a \in \mathcal{S}\times\mathcal{A}, \pi(a|s) \geq 0, \sum_a \pi(a|s) = 1\}$ is a set of policies, and τ is a sequence of state-action pairs sampled from the transition probability and policy, i.e., $s_{t+1} \sim P(\cdot|s_t, a_t), a_t \sim \pi(\cdot|s_t)$ for $t \in [0,\infty]$ and $s_0 \sim d$. For a given π, we can define the state value and state-action (or action) value as $V^\pi(s) \triangleq \mathbb{E}_{\tau\sim P,\pi}\left[\sum_{t=0}^\infty \gamma^t \mathbf{R}_t | s_0 = s\right]$ and $Q^\pi(s,a) \triangleq \mathbb{E}_{\tau\sim P,\pi}\left[\sum_{t=0}^\infty \gamma^t \mathbf{R}_t | s_0 = s, a_0 = a\right]$, respectively. The solution of an MDP is called the optimal policy $\pi^\star$. The optimal value $V^\star = V^{\pi^\star}$ and action-value $Q^\star = Q^{\pi^\star}$ satisfy the Bellman optimality equation as follows: For $\forall s, a$,

$$\begin{aligned} Q^\star(s,a) &= \mathop{\mathbb{E}}_{s'\sim P}\left[\mathbf{r}(s,a,s') + \gamma V^\star(s')\right], \\ V^\star(s) &= \max_{a'} Q^\star(s,a'), \pi^\star \in \arg\max_{a'} Q^\star(s,a'), \end{aligned} \tag{1}$$

where $\arg\max_{a'} Q^\star(s,a')$ indicates a set of the policy π satisfying $\mathbb{E}_{a\sim\pi}[Q^\star(s,a)] = \max_{a'} Q^\star(s,a')$ and $a \sim \pi^\star$ indicates $a \sim \pi^\star(\cdot|s)$. Note that there may exist multiple optimal policies if the optimal action value has multiple maxima with respect to actions.

B. Related Work

Recently, regularization on a policy function has been widely investigated in RL [3, 22, 29, 23, 13, 24, 14, 9, 7, 19, 6, 11, 4]. The main purpose of regularizing a policy is to encourage exploration by inducing a stochastic policy from regularization. If a policy converges to a greedy policy before collecting enough information about an environment, its behavior can be sub-optimal. This issue can be efficiently handled by a stochastic policy induced from a regularization.

The SG entropy has been widely used as a policy regularization. It has been empirically shown that maximizing the SG entropy of a policy along with reward maximization encourages exploration since the entropy maximization penalizes a greedy behavior [22]. In [9], it was also demonstrated that maximizing the SG entropy helps to learn diverse and useful behaviors. This penalty from the SG entropy also helps to capture the multi-modal behavior where the resulting policy is robust against unexpected changes in the environment [13]. Theoretically, [29, 23, 13, 14] have shown that the optimal solution of maximum entropy RL has a softmax distribution of state-action values, not a greedy policy. Haarnoja et al. [14] showed that the SG entropy has the benefits over exploring a continuous action space, however, the performance of SAC is sensitive to a regularization coefficient. Furthermore, the maximum SG entropy in RL provides the connection between policy gradient and value-based learning [29, 25]. Dai et al. [7] have also shown that maximum entropy induces a smoothed Bellman operator and it helps stable convergence of value function estimation.

While the SG entropy in RL provides better exploration, numerical stability, and capturing multiple optimal actions, it is known that the maximum SG entropy causes a performance loss since it hinders exploiting the best action to maximize the reward [19, 6]. Such drawback is often handled by scheduling a coefficient of the SG entropy to progressively vanish [5]. However, designing a proper decaying schedule is still a demanding task in that it often requires an additional validation step in practice. Grau-Moya et al. [12] handled the same issue by automatically manipulating the importance of actions using mutual information. On the other hand, Lee et al. [19] and Chow et al. [6] have proposed an alternative way to handle the exploitation issue of the SG entropy using a sparse Tsallis (ST) entropy, which is a special case of the Tsallis entropy [32]. The ST entropy encourages exploration while penalizing less on a greedy policy, compared to the SG entropy. However, unlike the SG entropy, the ST entropy may discover a sub-optimal policy since it enforces the algorithm to explore the environment less [19, 6].

Recently, an analysis of general concave regularization of a policy function has been investigated [3, 24, 11]. Azar et al. [3] proposes dynamics programming for regularized MDPs and provides theoretical guarantees for finite state-action spaces. While the theory was derived for general concave regularizer, only SG entropy-based algorithm is demonstrated on a simple grid world example [3]. Neu et al. [24] also applied an SG entropy-based algorithm to a simple discrete action space. In contrast to prior work [3, 24, 11], we focus on analyzing the Tsallis entropy in MDPs and RL[1]. We derive unique properties of the Tsallis entropy such as performance bounds. We also propose two dynamic programming algorithms and extend it to a continuous actor-critic method and empirically show that the proposed method outperforms the SG entropy-based method.

C. q-Exponential, q-Logarithm, and Tsallis Entropy

Before defining the Tsallis entropy, let us first introduce variants of exponential and logarithm functions, which are called q-exponential and q-logarithm, respectively. They are used to define the Tsallis entropy and defined as follows[2]:

$$\exp_q(x) \triangleq [1 + (q-1)x]_+^{\frac{1}{q-1}}, \ \ln_q(x) \triangleq (x^{q-1} - 1)/(q-1), \tag{2}$$

where $[x]_+ = \max(x, 0)$ and q is a real number. Note that, for $q = 1$, q-logarithm and q-exponential are defined as

[1]Note that the Tsallis entropy also provides concave regularization.

[2]Note that the definition of $\exp_q, \ln_q$, and the Tsallis entropy are different from the original one [2] but those settings can be recovered by setting $q = 2 - q'$, where q' is the entropic index used in [2].

their limitations, i.e., $\ln_1(x) \triangleq \lim_{q\to 1} \ln_q(x) = \ln(x)$ and $\exp_1(x) \triangleq \lim_{q\to 1} \exp_q(x) = \exp(x)$. Furthermore, when $q = 2$, $\exp_2(x)$ and $\ln_2(x)$ become a linear function. This property gives some clues that the entropy defined using $\ln_q(x)$ will generalize the SG (or ST) entropy and, furthermore, the proposed method can generalize an actor critic method using SG entropy [14] and ST entropy [19, 6].

Now, we define the Tsallis entropy using $\ln_q(x)$.

Definition 1 (Tsallis Entropy [2])**.** *The Tsallis entropy of a random variable X with the distribution P is defined as $S_q(P) \triangleq \mathbb{E}_{X\sim P}\left[-\ln_q(P(X))\right]$. q is called an entropic-index.*

The Tsallis entropy can represent various types of entropy by varying the *entropic index*. For example, when $q \to 1$, $S_1(P)$ becomes the Shannon-Gibbs entropy and when $q = 2$, $S_2(P)$ becomes the sparse Tsallis entropy [19]. Furthermore, when $q \to \infty$, $S_q(P)$ converges to zero. We would like to emphasize that, for $q > 0$, the Tsallis entropy is a concave function with respect to the density function, but, for $q \leq 0$, the Tsallis entropy is a convex function. Detail proofs are included in the supplementary material [20]. In this paper, we only consider the case when $q > 0$ since our purpose of using the Tsallis entropy is to give a bonus reward to a stochastic policy.

III. Maximum Tsallis Entropy in MDPs

In this section, we formulate MDPs with Tsallis entropy maximization, which will be named Tsallis MDPs. We mainly focus on deriving the optimality conditions and algorithms generalized for the entropic index so that a wide range of q values can be used for a learning agent. First, we extend the definition of the Tsallis entropy so that it can be applicable for a policy distribution in MDPs. The Tsallis entropy of a policy distribution π is defined by $S_q^\infty(\pi) \triangleq \mathbb{E}_{\tau\sim P,\pi}\left[\sum_{t=0}^\infty \gamma^t S_q(\pi(\cdot|s_t))\right]$. Using S_q^∞, the original MDPs can be converted into Tsallis MDPs by adding $S_q^\infty(\pi)$ to the objective function as follows:

$$\max_{\pi\in\Pi} \mathop{\mathbb{E}}_{\tau\sim P,\pi}\left[\sum_t^\infty \gamma^t \mathbf{R}_t\right] + \alpha S_q^\infty(\pi), \tag{3}$$

where $\alpha > 0$ is a coefficient. A state value and state-action value are redefined for Tsallis MDPs as follows: $V_q^\pi(s) \triangleq \mathbb{E}_{\tau\sim P,\pi}\left[\sum_{t=0}^\infty \gamma^t \left(\mathbf{R}_t + \alpha S_q(\pi(\cdot|s_t))|s_0 = s\right]\right.$ and $Q_q^\pi(s,a) \triangleq \mathbb{E}_{\tau\sim P,\pi}\left[\mathbf{R}_0 + \sum_{t=1}^\infty \gamma^t \left(\mathbf{R}_t + \alpha S_q(\pi(\cdot|s_t))|s_0 = s, a_0 = a\right]\right.$, where q is the entropic index. The goal of a Tsallis MDP is to find an optimal policy distribution that maximizes both the sum of rewards and the Tsallis entropy whose importance is determined by α. The solution of the problem (3) is denoted as $\pi_q^\star$ and its value functions are denoted as $V_q^\star = V_q^{\pi_q^\star}$ and $Q_q^\star = Q_q^{\pi_q^\star}$, respectively. In our analysis, α is set to one, however one can easiliy generalize the case of $\alpha \neq 1$ by replacing $\mathbf{r}, V$, and Q with $\mathbf{r}/\alpha, V/\alpha$, and Q/α, respectively. In the following sections, we first derive the optimality condition of (3), which will be called the Tsallis-Bellman optimality (TBO) equation. Second, dynamic programming to solve Tsallis MDPs is proposed with convergence and optimality guarantees. Finally, we provide the performance error bound of the optimal policy of the Tsallis MDP, where the error is caused by the additional entropy regularization term. The theoretical results derived in this section are extended to a practical actor-critic algorithm in Section V.

A. q-Maximum Operator

Before analyzing an MDP with the Tsallis entropy, we define an operator, which is called q-maximum. A q-maximum operator is a bounded approximation of the maximum operator. For a function $f(x)$, q-maximum is defined as follows:

$$q\text{-}\max_x\left(f(x)\right) \triangleq \max_{P\in\Delta}\left[\mathop{\mathbb{E}}_{X\sim P}\left[f(X)\right] + S_q(P)\right], \tag{4}$$

where Δ is a probability simplex whose element is a probability. The following theorem shows the relationship between q-maximum and maximum operators.

Theorem 1. *For any function $f(x)$ defined on a finite input space $\mathcal{X}$, the q-maximum satisfies the following inequalities.*

$$q\text{-}\max_x\left(f(x)\right) + \ln_q\left(1/|\mathcal{X}|\right) \leq \max_x(f(x)) \leq q\text{-}\max_x\left(f(x)\right), \tag{5}$$

where $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$.

The proof can be found in the supplementary material [20]. The proof of Theorem 1 utilizes the definition of q-maximum. This boundedness property will be used to analyze the performance bound of an MDP with the maximum Tsallis entropy. The solution of q-maximum is obtained as $P(x) = \exp_q\left(f(x)/q - \psi_q\left(f/q\right)\right)$, where $\psi_q(\cdot)$ is called a q-potential function [2], which is uniquely determined by the normalization condition:

$$\sum_{x\in X} P(x) = \sum_{x\in X} \exp_q\left(f(x)/q - \psi_q\left(f/q\right)\right) = 1. \tag{6}$$

A detail derivation can be found in the supplementary material [20]. The property of q-maximum and the solution of q-maximum plays an important role in the optimality condition of Tsallis MDPs.

B. Tsallis Bellman Optimality Equation

Using the q-maximum operator, the optimality condition of a Tsallis MDP can be obtained as follows.

Theorem 2. *For $q > 0$, an optimal policy $\pi_q^\star$ and optimal value $V_q^\star$ sufficiently and necessarily satisfy the following Tsallis-Bellman optimality (TBO) equations:*

$$\begin{aligned} Q_q^\star(s,a) &= \mathop{\mathbb{E}}_{s'\sim P}[\mathbf{r}(s,a,s') + \gamma V_q^\star(s')|s,a], \\ V_q^\star(s) &= q\text{-}\max_a(Q_q^\star(s,a)), \\ \pi_q^\star(a|s) &= \exp_q\left(Q_q^\star(s,a)/q - \psi_q\left(Q_q^\star(s,\cdot)/q\right)\right), \end{aligned} \tag{7}$$

where ψ_q is a q-potential function.

The proof can be found in the supplementary material [20]. The TBO equation differs from the original Bellman equation in that the maximum operator is replaced by the q-maximum operator. The optimal state value $V_q^\star$ is the q-maximum of the optimal state-action value $Q_q^\star$ and the optimal policy $\pi_q^\star$

is the solution of q-maximum (4). Thus, as q changes, $\pi_q^\star$ can represent various types of q-exponential distributions. We would like to emphasize that the TBO equation becomes the original Bellman equation as q diverges into infinity. This is a reasonable tendency since, as $q \to \infty$, S_∞ tends zero and the Tsallis MDP becomes the original MDP. Furthermore, when $q \to 1$, q-maximum becomes the log-sum-exponential operator and the Bellman equation of maximum SG entropy RL, (a.k.a. soft Bellman equation) [13] is recovered. When $q = 2$, the Bellman equation of maximum ST entropy RL, (a.k.a. sparse Bellman equation) [19] is also recovered. Moreover, our result guarantees that the TBO equation holds for all $q > 0$.

IV. Dynamic Programming for Tsallis MDPs

In this section, we develop dynamic programming algorithms for a Tsallis MDP: Tsallis policy iteration (TPI) and Tsallis value iteration (TVI). These algorithms can compute an optimal value and policy. TPI is a policy iteration method which consists of policy evaluation and policy improvement. TVI is a value iteration method that computes the optimal value directly. In the dynamic programming of the original MDPs, the convergence is derived from the maximum operator. Similarly, in the MDP with the SG entropy, log-sum-exponential plays a crucial role for the convergence. In TPI and TVI, we generalize such maximum or log-sum-exponential operators by the q-max operator, which is a more abstract notion and available for all $q > 0$. Note that proofs of all theorems in this section are provided in the supplementary material [20].

A. Tsallis Policy Iteration

We first discuss the policy evaluation method in a Tsallis MDP, which computes V_q^π and Q_q^π for fixed policy π. Similar to the original MDP, a value function of a Tsallis MDP can be computed using the expectation equation defined by

$$\begin{aligned} Q_q^\pi(s,a) &= \mathop{\mathbb{E}}_{s'\sim P}[\mathbf{r}(s,a,s') + \gamma V_q^\pi(s')|s,a], \\ V_q^\pi(s) &= \mathop{\mathbb{E}}_{a\sim\pi}[Q_q^\pi(s,a) - \ln_q(\pi(a|s))], \end{aligned} \tag{8}$$

where $s' \sim P$ indicates $s' \sim P(\cdot|s,a)$ and $a \sim \pi$ indicates $a \sim \pi(\cdot|s)$. Equation (8) will be called the Tsallis Bellman expectation (TBE) equation and it is derived from the definition of V_q^π and Q_q^π. Based on the TBE equation, we can define the operator for an arbitrary function $F(s,a)$ over $\mathcal{S}\times\mathcal{A}$, which is called the TBE operator,

$$\begin{aligned} \left[\mathcal{T}_q^\pi F\right](s,a) &\triangleq \mathop{\mathbb{E}}_{s'\sim P}[\mathbf{r}(s,a,s') + \gamma V_F(s')|s,a], \\ V_F(s) &\triangleq \mathop{\mathbb{E}}_{a\sim\pi}[F(s,a) - \ln_q(\pi(a|s))]. \end{aligned} \tag{9}$$

Then, the policy evaluation method for a Tsallis MDP can be simply defined as repeatedly applying the TBE operator to an initial function F_0,i.e., $F_{k+1} = \mathcal{T}_q^\pi F_k$.

Theorem 3 (Tsallis Policy Evaluation)**.** *For fixed π and $q > 0$, consider the TBE operator $\mathcal{T}_q^\pi$, and define Tsallis policy evaluation as $F_{k+1} = \mathcal{T}_q^\pi F_k$ for an arbitrary initial function F_0 over $\mathcal{S}\times\mathcal{A}$. Then, F_k converges to Q_q^π and satisfies the TBE equation (8).*

The proof of Theorem 3 relies on the contraction property of $\mathcal{T}_q^\pi$. The contraction property guarantees the sequence of F_k converges to a fixed point F_* of $\mathcal{T}_q^\pi$, i.e., $F_* = \mathcal{T}_q^\pi F_*$ and the fixed point F_* is the same as Q_q^π. The value function evaluated from Tsallis policy evaluation can be employed to update the policy distribution. In the policy improvement step, the policy is updated to maximize

$$\forall s,\ \pi_{k+1}(\cdot|s) = \arg\max_{\pi(\cdot|s)} \mathop{\mathbb{E}}_{a\sim\pi}[Q_q^{\pi_k}(s,a) - \ln_q(\pi(a|s))|s]. \tag{10}$$

Theorem 4 (Tsallis Policy Improvement)**.** *For $q > 0$, let π_{k+1} be the updated policy from (10) using $Q_q^{\pi_k}$. For all $(s,a) \in \mathcal{S}\times\mathcal{A}$, $Q_q^{\pi_{k+1}}(s,a)$ is greater than or equal to $Q_q^{\pi_k}(s,a)$.*

Theorem 4 tells us that the policy obtained by the maximization (10) has performance no worse than the previous policy. From Theorem 3 and 4, it is guaranteed that the Tsallis policy iteration gradually improves its policy as the number of iterations increases and it converges to the optimal solution.

Theorem 5 (Optimality of TPI)**.** *When $q > 0$, define the Tsallis policy iteration as alternatively applying (9) and (10), then π_k converges to the optimal policy.*

The proof is done by checking if the converged policy satisfies the TBO equation. In the next section, Tsallis policy iteration is extended to a Tsallis actor-critic method which is a practical algorithm to handle continuous state and action spaces in complex environments.

B. Tsallis Value Iteration

Tsallis value iteration is derived from the optimality condition. From (7), the TBO operator is defined by

$$\begin{aligned} \left[\mathcal{T}_q F\right](s,a) &\triangleq \mathop{\mathbb{E}}_{s'\sim P}\left[\mathbf{r}(s,a,s') + \gamma V_F(s)\middle|s,a\right], \\ V_F(s) &\triangleq \mathop{q\text{-max}}_{a'}\left(F(s,a')\right). \end{aligned} \tag{11}$$

Then, Tsallis value iteration (TVI) is defined by repeatedly applying the TBO operator,i.e., $F_{k+1} = \mathcal{T}_q F_k$.

Theorem 6. *For $q > 0$, consider the TBO operator $\mathcal{T}_q$, and define Tsallis value iteration as $F_{k+1} = \mathcal{T}_q F_k$ for an arbitrary initial function F_0 over $\mathcal{S}\times\mathcal{A}$. Then, F_k converges to $Q_q^\star$.*

Similar to Tsallis policy evaluation, the convergence of Tsallis value iteration depends on the contraction property of $\mathcal{T}_q$, which makes F_k converges to a fixed point of $\mathcal{T}_q$. Then, the fixed point can be shown to satisfy the TBO equation.

C. Performance Error Bounds and q-Scheduling

We provide the performance error bounds of the optimal policy of a Tsallis MDP which can be obtained by TPI or TVI. The error is caused by the regularization term used in Tsallis entropy maximization. We compare the performance between the optimal policy of a Tsallis MDP and that of the original MDP. The performance error bounds are derived as follows.

Theorem 7. *Let $J(\pi)$ be the expected sum of rewards of a given policy π, $\pi^\star$ be the optimal policy of an original MDP, and $\pi_q^\star$ be the optimal policy of a Tsallis MDP with an*

Algorithm 1 Tsallis Actor Critic (TAC)

Input: Total time steps $t_{\max}$, Max episode length $l_{\max}$, Memory size N, Entropy coefficient α, Entropic index q (or schedule), Moving average ratio τ, Environment *env*
Initialize: $\psi, \psi^-, \theta_1, \theta_2, \phi$, $\mathcal{D}$: Queue with size N, $t = 0$, $t_e = 0$
while $t \leq t_{\max}$ **do**
  $a_t \sim \pi_\phi$ and $\mathbf{r}_{t+1}, s_{t+1}, d_{t+1} \sim env$ where d_{t+1} is a terminal signal.
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, \mathbf{r}_{t+1}, s_{t+1}, d_{t+1})\}$
  $t_e = t_e + 1$, $t = t + 1$
  if d_{t+1} = True or $t_e = l_{\max}$ **then**
    for $i = 1$ to t_e **do**
      Randomly sample a mini-batch $\mathcal{B}$ from $\mathcal{D}$
      Minimize $J_\psi, J_{\theta_1}, J_{\theta_2}$, and J_ϕ using a stochastic gradient descent
      $\psi^- \leftarrow (1-\tau)\psi^- + \tau\psi$
    end for
    Reset env, $t_e = 0$
    if Schedule of q exists **then**
      Update q_t
    end if
  end if
end while

entropic index q. Then, the following inequality holds: $J(\pi^\star) + (1-\gamma)^{-1} \ln_q(1/|\mathcal{A}|) \leq J(\pi_q^\star) \leq J(\pi^\star)$, *where* $|\mathcal{A}|$ *is the cardinality of* $\mathcal{A}$ *and* $q > 0$.

The proof of Theorem 7 is included in the supplementary material [20]. Here, we can observe that the performance gap shows a similar property of the TBO equation. We further verify Theorem 7 on a simple grid world problem. We compute the expected sum of rewards of $\pi_q^\star$ obtained from TVI by varying q values and compare them to the bounds in Theorem 7. Notice that $\ln_q(1/|\mathcal{A}|) \propto 1/|\mathcal{A}|^{q-1}$ converges to zero as $q \to \infty$. This fact supports that $\pi_q^\star$ converges to the greedy optimal policy in the original Bellman equation when $q \to \infty$. Inspired by Theorem 7, we develop a scheduled TPI by linearly increasing q_k from zero to infinity during Tsallis policy iteration. From the following theorem, we can guarantee that it converges to the optimal policy of the original MDP.

Theorem 8 (Scheduled TPI)**.** *Let* $\mathcal{TPI}_q$ *be the Tsallis policy iteration operator with an entropic index q. Assume that a diverging sequence* q_k *is given, such that* $\lim_{k\to\infty} q_k = \infty$. *For given* q_k*, scheduled TPI is defined as* $\mathcal{TPI}_{q_k}$*, i.e.,* $\pi_{k+1} = \mathcal{TPI}_{q_k}(\pi_k)$. *Then,* $\pi_k \to \pi^\star$ *as* $k \to \infty$.

V. Tsallis Actor Critic for Model-Free RL

We extend Tsallis policy iteration to a Tsallis actor-critic (TAC) method, which can be applied to a continuous action space. From our theoretical results, existing SG entropy-based methods can be freely extended to utilize a Tsallis entropy by replacing the SG entropy term. In order to verify the pure effect of the Tsallis entropy, we modified the soft actor critic (SAC) method by employing $\ln_q(\pi(a|s))$ instead of $\ln(\pi(a|s))$ and compare to the SAC method.

Similarly to SAC, our algorithm maintains five networks to model a policy π_ϕ, state value V_ψ, target state value V_{ψ^-}, two state action values Q_{θ_1} and Q_{θ_2}. We also utilize a replay buffer $\mathcal{D}$ which stores every interaction data $(s_t, a_t, r_{t+1}, s_{t+1})$. To update state value network V_ψ, we minimize the following loss,

$$J_\psi = \mathop{\mathbb{E}}_{s_t, a_t \sim \mathcal{B}} \left[\left(y_t - V_\psi(s_t) \right)^2 / 2 \right] \tag{12}$$

where $\mathcal{B} \subset \mathcal{D}$ is a mini-batch and y_t is a target value defined as $y_t = Q_{\min}(s_t, a_t) - \alpha \ln_q(\pi_\phi(a_t|s_t)$, and, $Q_{\min}(s_t, a_t) = \min[Q_{\theta_1}(s_t, a_t), Q_{\theta_2}(s_t, a_t)]$. The technique using the minimum state action value between two approximations of Q^π is known to prevent overestimation problem [10] and makes the learning process numerically stable. After updating ψ, ψ^- is updated by an exponential moving average with a ratio τ. For both θ_1 and θ_2, we minimize the following loss function,

$$J_\theta = \mathop{\mathbb{E}}_{b_t \sim \mathcal{B}} \left[\left(Q_\theta(s_t, a_t) - r_{t+1} - \gamma V_{\psi^-}(s_{t+1}) \right)^2 / 2 \right], \tag{13}$$

where b_t is $(s_t, a_t, s_{t+1}, r_{t+1})$. This loss function is induced by the Tsallis policy evaluation step.

When updating an actor network, we minimize a policy improvement objective defined as

$$J_\phi = \mathop{\mathbb{E}}_{s_t \sim \mathcal{B}} \left[\mathop{\mathbb{E}}_{a \sim \pi_\phi} \left[\alpha \ln_q(\pi_\phi(a|s_t)) - Q_\theta(s_t, a) \right] \right]. \tag{14}$$

Note that a is sampled from π_ϕ not a replay buffer. Since updating J_ϕ requires to compute a stochastic gradient, we use a reparameterization trick similar to Haarnoja et al. [14] instead of a score function estimation. In our implementation, a policy function is defined as a Gaussian distribution defined by a mean μ_ϕ and variance σ_ϕ^2. Consequently, we can rewrite J_ϕ with a reparameterized action and a stochastic gradient is computed as

$$\nabla_\phi J_\phi = \mathop{\mathbb{E}}_{s_t \sim \mathcal{B}} \left[\mathop{\mathbb{E}}_{\epsilon \sim P_\epsilon} \left[\alpha \nabla_\phi \ln_q(\pi_\phi(a|s_t)) - \nabla_\phi Q_\theta(s_t, a) \right] \right],$$

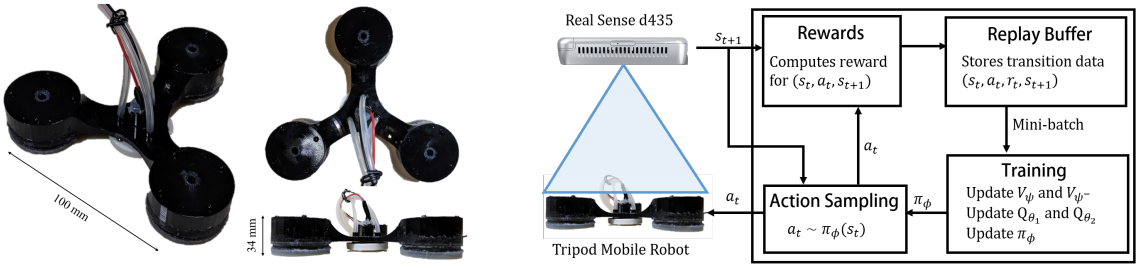
where $a = \mu_\phi + \sigma_\phi \epsilon$ and ϵ is a unit normal noise. Furthermore, we present TAC with Curricular (TAC^2) that gradually increase the entropic index q based on Theorem 8. While it is optimal to search the proper entropic index given an RL problem, the exhaustive search is often impractical due to prohibitive high sample complexity. The entire TAC and TAC^2 algorithms are summarized in Algorithm 1.

VI. Experiments Setup

A. Simulation Setup

To verify the characteristics and efficiency of our algorithm, we prepare four simulation tests on continuous control problems using the MuJoCo simulator: HalfCheetah-v2, Ant-v2, Pusher-v2, and Humanoid-v2. For each task, a robot with multiple actuated joints is given where the number of joints is different from each task. Then, a state is defined as sensor measurements of actuators and an action is defined as torques. The goal of each task is to control a robot with multiple actuated joints to move forward as fast as possible. More detailed definition can be found in [8].

In the first simulation, to verify the effect of the entropic index q, we conduct experiments with a wide range of q values from 0.5 to 5.0 and measure the total average returns during



(a) Tripod Mobile Robot (b) Training System

Fig. 1. (a) A soft mobile robot used in experiment. (b) A diagram for training system. The position of robot is measured by using blob detection from a RGB image ofr RealSense d435.

the training phase. We only change the entropic index and fix an entropy coefficient α to 0.05 for Humanoid-v2 and 0.2 for other problems. We run entire algorithms with ten different random seeds. Second, to verify the effect of α, we run TAC with different q values (including SAC) for three α values: $0.2, 0.02$, and 0.002 on the Ant-v2 problem. Third, we test the variant of TAC by linearly scheduling the entropic index. From the results of the first simulations, we observe that there exists a numerically stable region of $1 < q < 2$, which will be explained in Section VII-A. We schedule q to linearly increase from 1 to 2 for every 5000 steps and we run TAC with q schedule for three α values: $0.2, 0.02$, and 0.002 on the Ant-v2 problem. Finally, we conduct a compare our algorithm to the existing state-of-the-art on-policy and off-policy actor-critic methods. For on-policy methods, trust region policy optimization (TRPO) [27] and proximal policy optimization (PPO) [30] are compared where a value network is employed for generalized advantage estimation [28]. For off-policy methods, deep deterministic policy gradient (DDPG) [21] and twin delayed DDPG which is called TD3 [10] are compared. We also compare with the soft actor-critic (SAC) method [14] which employs the SG entropy for exploration. Since TAC can be reduced to SAC with $q = 1$ and algorithmic details are the same, we denote TAC with $q = 1$ as SAC. We utilize OpenAI's implementations [1] and extend the SAC algorithm to TAC. To obtain consistent results, we run all algorithms with ten different random seeds. While we compare various existing methods, results of TRPO, PPO, and DDPG are omitted here due to their poor performance and the entire results can be found in the supplementary material [20]. The source code is publicly available[3].

B. Hardware Platform Setup

To test our algorithm on a soft mobile robot, we use a tripod mobile robot that consisted of three pneumatic soft vibration actuators, a direct current (DC) motor, and an equilateral triangle body plate as shown in Figure 1(a). Each actuator can independently vibrate continuously and robustly regardless of contact with external objects by using the nonlinear stiffness characteristic of hyperelastic material (Eco-flex 30). In addition, the vibration frequency of the actuator can be controlled by the input pressure. In order to control the direction of rotation of the robot, a direct current (DC) motor was installed at the center of the robot combined with a rotating plate. As a result, the mobile robot is capable of making various motions, such as translation and rotation, with a combination of the three vibration modes of the actuator and the rotation of the rotating plate.

[3]https://github.com/rllab-snu/tsallis_actor_critic_mujoco

C. Real Robot Experiment Setup

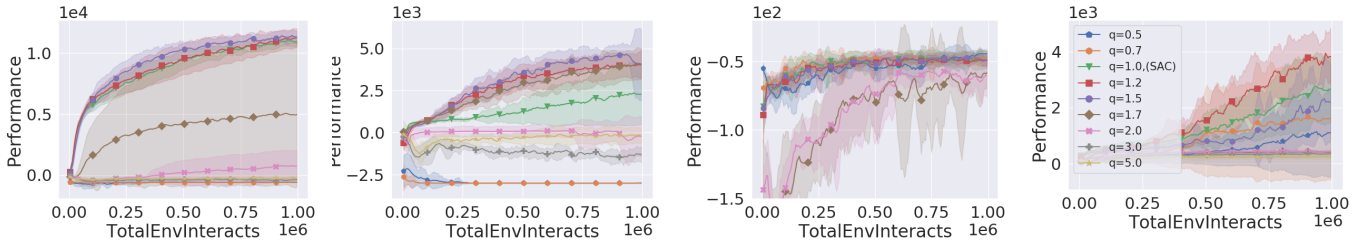
We apply the proposed algorithm to a soft mobile robot and compare the proposed method to SAC with $\alpha = 0.01$ and SAC with automatic entropy adjustment (SAC-AEA) [15] which automatically adjusts α to maintain the entropy to be greater than a predefined threshold δ. In experiment, we heuristically set δ to $-\ln(d)$ as proposed in [15] where d is a dimension of the action space. In [15], since SAC-AEA shows efficient performances for learning quadrupedal locomotion, we try to check whether SAC-AEA can be applied to a soft mobile robot while comparing their performance to the proposed method. We would like to note that TAC[2] only schedule q with fixed α and SAC-AEA only changes α with fixed $q = 1.0$. From this comparison, we can demonstrate which factor is more important to achieve efficient exploration.

In this task, our goal is to train a feedback controller of a soft mobile robot where a controller makes a robot move in a straight line towards a goal position (x_g, y_g) with a heading $\theta_g := \arctan(y_g - y_t, x_g - x_t)$ where x_t, y_t is a current position of the robot. Note that if a robot's heading is aligned to its moving direction, then, $\theta_g = \theta_t$.

The robot has three soft membrane vibration actuators and one motor for controlling the angular momentum of the robot. Hence, an action is defined as a four dimensional vector as $a_t = (p_1, p_2, p_3, \delta\Omega)_t$, where p_i is an input pressure of each vibration actuator and $\delta\Omega$ is the change in the motor speed. Note that, if we directly change the motor signal, it may generate unstable motion and inconsistent movements due to the delay of the motor. Hence, by controlling a difference of the motor signal, we can generate a smooth change of motor speed.

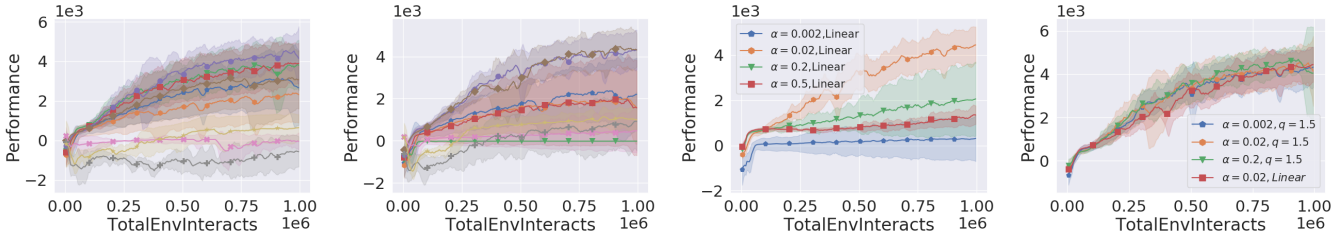
Then, a state of a robot is defined as $s_t := (\Delta\theta_t, d_t, \Omega_t)$, where $\Delta\theta_t := \theta_g - \theta_t$ is a difference between heading and goal direction, $d_t := \sqrt{(x_t - x_g)^2 + (y_t - y_g)^2}$, is the Euclidean distance to the desired position, and Ω_t is the current motor speed.

A reward function $r(s_t)$ assigns a higher score as a control minimizes the gap between robot's current state and desired



(a) HalfCheetah-v2 (b) Ant-v2 (c) Pusher-v2 (d) Humanoid-v2

Fig. 2. Average training returns of TAC with different q values on four MuJoCo tasks. A solid line is the average return over ten trials and the shade area shows one variance.



(a) Ant-v2, $\alpha = 0.02$ (b) Ant-v2, $\alpha = 0.002$ (c) Ant-v2, Scheduled q_k (d) Ant-v2, Comparison

Fig. 3. (a), (b) Average returns of different $\alpha = \{0.02, 0.002\}$ and different q. (a) and (b) share the legend with Figure 2(d). (c) Average returns of scheduling q_k with different α. Linear indicates linear curriculum of q_k. (d) Comparison of all variants of TAC.

state: $r(s_t) := -d_t - |\Delta\theta_t| + 2$, which is a decreasing function of d_t and $\Delta\theta_t$ where 2 is added to give a positive reward near the goal position. γ is set to 0.99. The entire training system is illustrated in Figure 1(b).

For a fair comparison, we evaluate each algorithm every 500 steps. In evaluation, we control a robot using only the mean value of the trained policy without sampling. We run all algorithms with 2500 steps for five trials.

VII. Result

A. Effect of Entropic Index q

The results are shown in Figure 2. We realize that the proposed method performs better when $1 \leq q < 2$ than when $0 < q < 1$ and $q \geq 2$, in terms of stable convergence and final total average returns. Using $0 < q < 1$ generally shows poor performance since it hinders exploitation more strongly than the SG entropy. For $1 \leq q < 2$, the Tsallis entropy penalizes less the greediness of a policy compared to the SG entropy (or $q = 1$). From a reparameterization trick, the gradient of the Tsallis entropy becomes $\mathbb{E}_{a\sim\pi_\phi}[\pi_\phi(a|s)^{q-2}\nabla_\phi\pi_\phi(a|s)]$. For $q \geq 2$, the gradient is proportional to $\pi_\phi(a|s)$, thus, if $\pi_\phi(a|s)$ is small, then, the gradient becomes smaller and it leads to early convergence to a locally optimal policy. For $0 < q < 2$, the gradient is proportional to $1/\pi_\phi(a|s)$, thus, if $\pi_\phi(a|s)$ is small, the gradient becomes larger, which encourages exploration of the action with a small probability. For $0 < q < 1$, since $\pi_\phi(a|s)^{q-2}$ is more amplified than when $1 \leq q < 2$, the penalty of greediness is stronger than when $1 \leq q < 2$. Thus, when $0 < q < 1$, it penalizes the exploitation of TAC more and hinders the convergence to an optimal policy. In this regard, we can see TAC with $1 \leq q < 2$ outperforms TAC with $q \geq 2$. Furthermore, in HalfCheetah-v2 and Ant-v2, TAC with $q = 1.5$ shows the best performance and, in Humanoid-v2, TAC with $q = 1.2$ shows the best performance. Furthermore, in Pusher-v2, the final total average returns of all settings are similar, but TAC with $q = 1.2$ shows slightly faster convergence. We believe that these results empirically show that there exists an appropriate q value between one and two depending on the environment while $q \geq 2$ has a negative effect on exploration.

B. Effect of Coefficient α

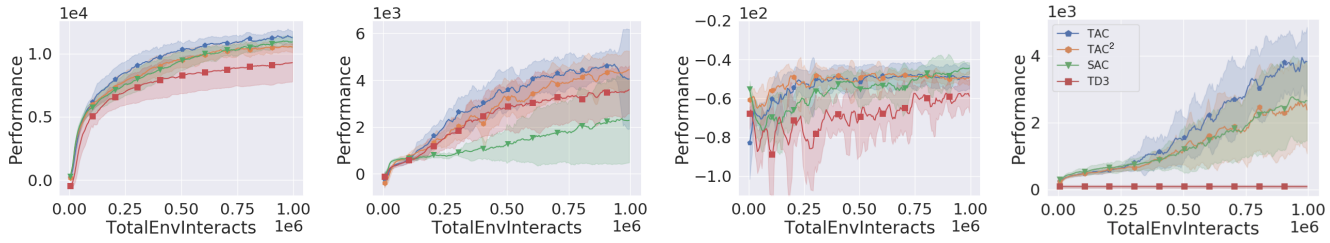
As shown in Figure 2(b),3(a) and 3(b). For all α values, $q = 1.5$ (purple circle line) always shows the fastest convergence and achieves the best performance among tested q values This result tells us that TAC with the best q value is robust to change α. For $q = 1.2$ (or $q = 1.7$), the average return of TAC with $q = 1.2$ (or $q = 1.7$) is sensitive to α, respectively, where $q = 1.7$ has the best average return at $\alpha = 0.002$, and $q = 1.2$ has the best value at $\alpha = 0.02$. However, TAC with $q = 1.5$ consistently outperforms other entropic indices while α is changed.

C. Curriculum on Entropic Index q

Figure 3(c) shows the performance of TAC^2 with different α and Figure 3(d) illustrates the comparison to TAC with fixed q. From this observation, it is shown that TAC^2 achieves a similar performance of the best q value without using a brute force search.

D. Comparative Evaluation

Figure 4 shows the total average returns of TAC and other compared methods. We use the best combination of q and α from the previous experiments for TAC with $q \neq 1$ and SAC (TAC with $q = 1$). SAC and TAC use the same architectures for actor and critic networks. TAC and TAC^2 indicates TAC with



(a) HalfCheetah-v2 (b) Ant-v2 (c) Pusher-v2 (d) Humanoid-v2

Fig. 4. Comparison to existing actor-critic methods on four MuJoCo tasks. SAC (red square line) is the same as TAC with $q = 1$, TAC and TAC^2 indicates TAC with fixed $q \neq 1$ and scheduled q, respectively.

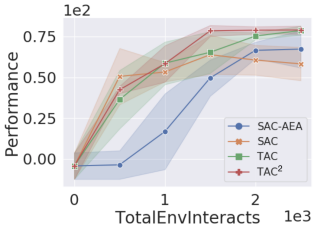
the fixed best q and linearly scheduled q, respectively. First, TAC with a proper q outperforms all existing methods in all environments. Furthermore, TAC achieves better performance with a smaller number of samples than SAC and TD3 in all problems. Especially, in Ant-v2, TAC improves the performance from SAC by changing $q = 1.5$. Furthermore, in Humanoid-v2 which has the largest action space (17D), TAC with $q = 1.2$ outperforms all the other methods. Finally, TAC^2 consistently shows similar performances to TAC, except Humanoid-v2.

E. Real Robot Experiment

Figure 5 shows the results of compared algorithms including the proposed method. TAC^2 shows the best performance in terms of the convergence speed and the sum of rewards compared to other algorithms. In particular, the policy trained by TAC^2 could reach any goal point with only about 1500 steps (≈30 minutes) of training. Furthermore, TAC with $q = 1.5$ shows the second-best performance.

For SAC and SAC-AEA, while SAC-AEA shows slower convergence than SAC due to the constraint to keep the entropy of the policy above the threshold, it achieves higher performance than SAC at the end of the training. This result demonstrates that maintaining the entropy of the policy helps exploration and leads to better final performance, however, it hampers the exploitation.

While both TAC^2 and SAC-AEA control the exploration-exploitation trade-off by scheduling the level of regularization, the empirical result shows that scheduling q instead of adjusting α shows better performance in terms of both convergence speed and final average return. While adjusting α in SAC-AEA only rescales the magnitude of the gradient of the entropy, scheduling q can change both the scale and direction of the gradient of the entropy, similarly to the results in Section VII-A. Specifically, in TAC^2, the regularization effect is gradually reduced as the entropic index q increases while SAC-AEA keeps the level of the Shannon entropy. Hence, scheduling q helps exploitation at the end of the training. Thus, TAC^2 shows not only the highest final average performance but also a much smaller variance than other algorithms, which is a highly preferred feature for training a soft mobile robot. Especially, a low variance of the final performance supports that TAC^2 successfully overcome the unknown stochasticity in the dynamic model of the soft mobile robot. Consequently, we can conclude that TAC^2 efficiently learns a feedback controller of a soft mobile robot and achieves the best performance with the minimum interactions.



(a) Evaluation

Alg.	Fin. Ret.
TAC	75.9 (5.6)
TAC^2	78.7 (2.4)
SAC	67.3 (10.2)
SAC-AEA	58.3 (9.1)

(b) Final Performance

Fig. 5. Comparison to existing actor-critic methods on training a Tripod mobile robot. (a) Average returns over five trials. (b) Final average performance. The number in parentheses is a standard deviation.

VIII. Conclusion

We have proposed a unified framework which allows using a class of different Tsalli entropies in RL problems, which we call Tsallis MDPs, and its application to soft robotics. We first provide the full theoretical analysis about Tsallis MDPs including guarantees of convergence, optimality, and performance error bounds. and have extended it to the Tsallis actor-critic (TAC) method to handle a continuous state-action space. It has been observed that there exists a suitable entropic index for each different RL problem and TAC with the optimal entropic index outperforms existing actor-critic methods. However, since finding an entropic index with the brute force search is a demanding task, we have also present TAC^2 that gradually increases the entropic index and empirically show that it achieves comparable performances with TAC with the optimal entropic index found from an exhaustive search in simulation environments. We have applied TAC^2 on real-world problems of learning a feedback controller for soft mobile robots and demonstrated that TAC^2 shows more efficient exploration tendency than adjusting the regularization coefficient.

Acknowledgements This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning and No.2020-0-01336, Artificial Intelligence Graduate School Program (UNIST)), the National Research Foundation (NRF-2016R1A5A1938472) funded by the Korean Government (MSIT), and the 2020 Research Fund (1.200086.01, 1.200098.01) of UNIST(Ulsan National Institute of Science & Technology).

References

[1] Joshua Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

[2] Shunichi Amari and Atsumi Ohara. Geometry of q-exponential family of probability distributions. *Entropy*, 13(6):1170–1185, 2011.

[3] Mohammad Gheshlaghi Azar, Vicenç Gómez, and Hilbert J. Kappen. Dynamic policy programming. *Journal of Machine Learning Research*, 13:3207–3245, 2012.

[4] Boris Belousov and Jan Peters. Entropic regularization of markov decision processes. *Entropy*, 21(7):674, 2019.

[5] Nicolò Cesa-Bianchi, Claudio Gentile, Gergely Neu, and Gábor Lugosi. Boltzmann exploration done right. In *Proc. of the Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, CA, USA, 2017.

[6] Yinlam Chow, Ofir Nachum, and Mohammad Ghavamzadeh. Path consistency learning in tsallis entropy regularized mdps. In *Proc. of the 35th International Conference on Machine Learning, (ICML)*, Stockholmsmässan, Stockholm, Sweden, 2018.

[7] Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. SBEED: convergent reinforcement learning with nonlinear function approximation. In *Proc. of the 35th International Conference on Machine Learning, (ICML)*, Stockholmsmässan, Stockholm, Sweden, 2018.

[8] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proc. of the 33nd International Conference on Machine Learning, (ICML)*, New York City, NY, USA, 2016.

[9] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

[10] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proc. of the 35th International Conference on Machine Learning, (ICML)*, Stockholmsmässan, Stockholm, Sweden, 2018.

[11] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. *arXiv preprint arXiv:1901.11275*, 2019.

[12] Jordi Grau-Moya, Felix Leibfried, and Peter Vrancx. Soft q-learning with mutual-information regularization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

[13] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proc. of the 34th International Conference on Machine Learning, (ICML)*, Sydney, NSW, Australia, 2017.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. of the 35th International Conference on Machine Learning, (ICML)*, Stockholmsmässan, Stockholm, Sweden, 2018.

[15] Tuomas Haarnoja, Aurick Zhou, Sehoon Ha, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. In *Proc. of the 15th Robotics: Science and Systems (RSS)*, 2019.

[16] Jemin Hwangbo, Inkyu Sa, Roland Siegwart, and Marco Hutter. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103, 2017.

[17] DongWook Kim, Jae In Kim, and Yong-Lae Park. A simple tripod mobile robot using soft membrane vibration actuators. *IEEE Robotics and Automation Letters*, 4(3): 2289–2295, 2019.

[18] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294, 2013.

[19] Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1466–1473, 2018.

[20] Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, Mineui Hong, Jaein Kim, Yong-Lae Park, and Songhwai Oh. Supplementary material for generalized tsallis entropy reinforcement learning and its application to soft mobile robots. Technical report, Department of Electrical and Computer Engineering, Seoul National University, 2020.

[21] 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. *CoRR*, abs/1509.02971, 2015.

[22] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proc. of the 33nd International Conference on Machine Learning, (ICML)*, New York City, NY, USA, 2016.

[23] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Proc. of the Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, CA, USA, 2017.

[24] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[25] Brendan O'Donoghue, Rémi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. PGQ: combining policy gradient and q-learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017.

[26] Daniel J Preston, Haihui Joy Jiang, Vanessa Sanchez, Philipp Rothemund, Jeff Rawson, Markus P Nemitz, Won-Kyu Lee, Zhigang Suo, Conor J Walsh, and George M Whitesides. A soft ring oscillator. *Science Robotics*, 4 (31):eaaw5496, 2019.

[27] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proc. of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, 2015.

[28] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015.

[29] John Schulman, Pieter Abbeel, and Xi Chen. Equivalence between policy gradients and soft q-learning. *CoRR*, abs/1704.06440, 2017.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[31] Thomas George Thuruthel, Egidio Falotico, Federico Renda, and Cecilia Laschi. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 35(1): 124–134, 2018.

[32] Constantino Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of statistical physics*, 52(1-2): 479–487, 1988.

[33] Haochong Zhang, Rongyun Cao, Shlomo Zilberstein, Feng Wu, and Xiaoping Chen. Toward effective soft robot control via reinforcement learning. In *International Conference on Intelligent Robotics and Applications*, pages 173–184. Springer, 2017.