Mutual Information State Intrinsic Control With Tsallis Entropy Regularization

Jongkook Kim[1], Songhwai Oh[1]

Abstract— **RL researches on exploration has been centered around Atari-games and control problems within the Mujoco environment. Mutual Information State Intrinsic Control(MUSIC) is one of the first RL algorithm with intrinsically motivated exploration strategy that proved successful in robotic manipulation tasks. MUSIC utilizes agent-surrounding state separation to compute mutual information between the two and use it as an intrinsic motivation(or a proxy for prioritization coefficient). While Soft-Actor-Critic version of the algorithm is better performing than its Deep Deterministic Policy Gradient version, having a "dense" stochastic policy(as is used in SAC) seems to hinder its performance in certain task. Considering that too much stochasticity in its policy may hinder its applicability in real situation performance of robotic manipulation task, this project utilizes Tsallis Entropy Regularization to the MUSIC algorithm and shows improvements in its performance in FetchPush-v1, FetchSlide-v1, FetchPickandPlace-v1 tasks.**

I. INTRODUCTION

Reinforcement learning(RL) combined with a powerful function approximators like neural network ha shown success on challenging sequential decision making problems. Final objective of RL is to maximize the expected reward given a policy. Model-free RL algorithms aims to learn a policy that effectively performs a given task given the current state and the reward that is resulted from certain actions without any model formation of the environment, whereas model-based RL constructs its own dynamics model to evaluate its policy and plan accordingly. Reinforcement learning requires a meticulate control of the exploration-exploitation trade-off - an agent may choose to take actions that the agent already knows to be rewarding in light of the past action-reward transitions or to take actions that the agent is unsure which result it will bring about. New actions that were unexplored before carries a potential of bigger results, and thus a better trajectory and outcome, but the agent would have to pay the opportunity cost of taking the 'already known good' action.

Various benchmarks and experimental environments were providing including the Arcade Learning Environment(ALE)[8], Mujoco environment[43], D4RL[16], Raisim, and was used widely to measure the extent to which agents can task various techniques to maximize the expected reward. Deep Q-Network was the first to effectively deal with raw Pixel Images and still learn a powerful policy in the ALE. [29], [39], [40], [18] accelerated the dimensions in which reinforcement learning could be applied - in terms of continuous control, faster convergence, higher entropy, higher state dimensions, etc. [18] introduced and entropy term $\mathcal{H}(\pi(a|s))$ into the loss function, encouraging the policy to take diverse actions when given a state. Moreover, Prioritized Experience Replay [38], Hindsight Experience Replay[2] added colors to the use of replay buffer in accelerating learning and stimulating convergence.

Intrinsic motivation has seen significant success as a guide for exploration in sparse reward environments. Since external rewards rarely do appear and given to the agent to learn about the environment, there were increasing need to develop a internal mechanism that would drive the agent to go to the unexplored areas (or states). Count-based methods were among the first attempts to [9], [31] record the number certain state was visited. With raw pixel inputs, density models were used to group states together. Such methodology was improved in [42] where hashing was used. Curiosity based models[33], [10] used forward dynamics models based on [19]. Self-supervised learning[34] and ensembles also improved its performance.

Recent literature on unsupervised representation learning generally focuses on extracting latent representation maximizing an approximate lower bound on the mutual information between the code and the data. [7] aim at maximizing the approximation of mutual information between latent code and the raw data, estimating the mutual information with neural networks Donsker and Varadhan estimation to learn better generative models. [20] builds on the idea and trains a decoder-free encoding representation maximizing the mutual information between the input image and the representation. However, mutual information itself has not been applied to estimate the change itself in the state that is caused by agent's interaction with the environment. Furthermore, mutual information proved useful in [46] with robotic manipulation tasks.

Robots were originally designed to assist or replace humans by performing repetitive or dangerous tasks that humans would not normally prefer or could not perform due to physical limitations imposed by extreme environments. These include the limited accessibility of long, narrow pipes underground, the anatomical location of certain minimally invasive surgical procedures, and objects on the ocean floor, for example. With the continuous development of machines, sensing technology, intelligent control and other modern technologies, robots have improved their autonomy capabilities and become more agile. Today, commercial and industrial robots are widely used in both fields such as manufacturing, assembly, packaging, transportation, surgery, and Earth and space exploration due to their low long-term cost and high accuracy and reliability.

There are different types of robots available, which can

[1] Electrical and Computer Engineering Department, Seoul National University `jkkim123@gmail.com`

be grouped into several categories depending on their movement, Degrees of Freedom (DoF), and function. Articulated robots, are among the most common robots used today. They look like a human arm and that is why they are also called robotic arm or manipulator arm. In some contexts, a robotic arm may also refer to a part of a more complex robot. A robotic arm can be described as a chain of links that are moved by joints which are actuated by motors. We will start from a brief explanation of these mechanical components of a typical robotic manipulator [3,4]. Figure 1 shows the schematic diagram of a simple two-joint robotic arm mounted on a stationary base on the floor. Robotic manipulation task[26] has been tackled from various aspects, including object and environment representation, compositional and hierarchical task structures, characterizing skills by preconditions and effects[4], skill policies[1], and transition models[37].

Tsallis entropy regularization is a generalized form of Shannon-Gibbs entropy and was successful in creating sparsely stochastic policies rather than "dense" policies. In many application, the policy need not become dense, since there may be some optimal paths to accomplish the task. Tsallis reinforcement learning has been presented as a unified framework in [27] and [28]. The proposed framework is formulated as a new class of Markov decision process using Tsallis entropy maximization called Tsallis MDP. Tsallis entropy essentially generalizes classes of entropy, including standard Shannon-Gibbs (SG) entropy, by controlling a parameter called the entropy index, and Tsallis MDP introduces a unified view of the various uses of entropy in RL. Tsallis MDP's different entropy indices provide a comprehensive analysis of how to generate different types of optimal policies and Bellman optimal equations. The theoretical results allow us to interpret the results of various types of entropy normalization in RL. In particular, different optimal policies due to entropy indices provide different exploration-exploitation trade-off behavior because the entropy indices affect the probability of that optimal policy. This feature is actually highly desirable as the sample complexity is highly affected by the exploration-exploitation trade-off and can provide systematic control over the trade-off by controlling the entropy index.

MUSIC was the first fully intrisically motivated exploration method on robotic manipulation task, however did perform well only on FetchPush-v1 with 95% success rate, but not on FetchPickAndPlace-v1(53%) or FetchSlide-v1(28%). The main results have been produced with using DDPG[29] and SAC[18]. The paper tries to encourage exploration via having the additional entropic term from [18]. However although exploration is necessary and "dense" policies do leave space for exploration, the stochasticity of its policy need not be dense and be open to all modes of actions in the real world. This paper attempts to reduce such dense entropy into a sparse one and the results have shown faster convergence and/or better success rate.

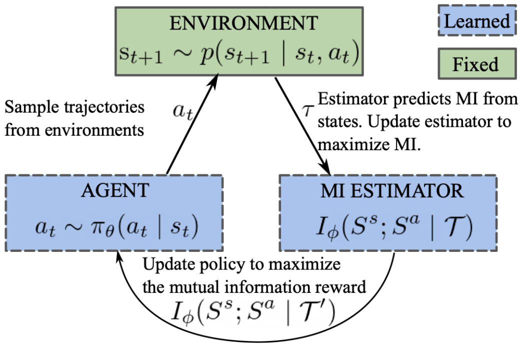
ENVIRONMENT
$s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$
Learned
Fixed
Sample trajectories from environments
a_t
τ
Estimator predicts MI from states. Update estimator to maximize MI.
AGENT
$a_t \sim \pi_\theta(a_t \mid s_t)$
MI ESTIMATOR
$I_\phi(S^s; S^a \mid \mathcal{T})$
Update policy to maximize the mutual information reward
$I_\phi(S^s; S^a \mid \mathcal{T}')$

Fig. 1. Diagram of the MUSIC algorithm. Agent learns the policy(SAC, DDPG) and the Mutual Information Estimator together. The estimator utilizes the division of surrounding state and the agent state. Unsupervised MUSIC version is first trained in advance for 50 epochs to learn the MI estimator and then used for exploration afterwards.

II. Preliminaries

A. Reinforcement Learning Setting

A Markov decision process(MDP) is defined as a tuple $M = S, A, d, P, \gamma, r$, where S is the state space, F is the corresponding feature space, A is the action space, d(s) is the distribution of an initial state, $P(s'|s,a)$ is the transition probability from $s \in S$ by taking $a \in A, \gamma \in (0,1)$ is a discount factor, and r is the reward function defined $r(s,a,s') \triangleq \mathbb{E}[\mathbf{R}|s,a,s']$ with random reward $\mathbf{R}$. In Tsallis MDP, this r is assumed to be bounded. Then, the MDP problem can be formulated as :

$$max_{\pi \in \Pi} \mathbb{E}_\pi[\Sigma_t^\infty \gamma^t \mathbf{R}_t], \tag{1}$$

where $\Sigma_t^\infty \gamma^t \mathbf{R}_t$ is a discounted sum of rewards, also called a return, $\Pi = \{\pi | \forall s, a \in \mathcal{S} \times \mathcal{A}, \pi(a|s) \geq 0, \Sigma_a \pi(a|s) = 1\}$ 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 $s_o \sim d$. For a given π, we can define the state value and state-action (or action) value as $V^\pi(s) \triangleq \mathbb{E}_{\tau \sim P,\pi}[\Sigma_t^\infty \gamma^t \mathbf{R}_t | s_o = s]$ and $Q^\pi(s,a) \triangleq \mathbb{E}_{\tau \sim P,\pi}[\Sigma_t^\infty \gamma^t \mathbf{R}_t | s_o = s, a_o = a]$, respectively. The solution of an MDP is called the optimal policy π^*. The optimal value $V^* = V^{\pi^*}$ and the action-value $Q^* = Q^{\pi^*}$ satisfy the Bellman optimality equation as follows: For $\forall s, a$,

$$Q^*(s,a) = \mathbb{E}_{s' \sim P}[\mathbf{r}(s,a,s') + \gamma V^*(s')] \tag{2}$$

$$V^*(s) = max_{a'} Q^*(s,a'), \pi^* \in argmax_{a'} Q^*(s,a') \tag{3}$$

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

B. Agent State, Surrounding state and Mutual Information Reward Function

In MUSIC[46], the state is division into surrounding state s^s and agent state s^a. Agent state refers to the state variable for the agent, whereas the surrounding state means the state variable that described the surroundings of the agent, such as the state variable of an object as is shown in Figure 1. For multi-goal environments, the assumption from [35] and [3] is applied that goals can be represented as states. Goal variable is denoted as g. For example, in the manipulation task, a goal is a particular desired position of the object in the episode. These desired positions are sample from the environment. The division between the agent and the surrounding state is naturally defined by the agent surrounding separation concept derived from psychology.

C. Tsallis Entropy Regularization and Tsallis Actor Critic

Before defining Tsallis entropy, let us first introduce variants of exponential and logarithmic functions, called q-exponential and q-logarithmic respectively. It is used to define the Tsallis entropy and is defined as:

$$exp_q(x) \triangleq [1+(q-1)x]_+^{\frac{1}{q-1}}, \ln_q(x) \triangleq (x^{q-1}-1)/(q-1) \tag{4}$$

where $[x]_+ = max(x, 0)$ and q is a real number. Note that, for q=1, q-logarithm and q-exponential are defined as 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 and $\ln_2$ 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, Tsallis entropy regularization method can generalize an actor critic method using SG entropy and ST entropy.

The definition of Tsallis entropy of a random variable X with the distribution P is defined as follow:

$$S_q(S) \triangleq \mathbb{E}_{X\sim P}[-ln_q(P(X))], \tag{5}$$

where 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. Furthermore, when $q \to \infty$, $S_q(P)$ converges to zero. 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.

[27] clearly drives the optimality conditions and lays out the algorithms generalized for the entropic index. 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

$$S_q^\infty(\pi) \triangleq \mathbb{E}_{\tau\sim P,\pi}[\Sigma_{t=0}^\infty \gamma^t S_q(\pi(\cdot|s_t))]. \tag{6}$$

Using S_q^∞, the original MDPs can be converted into Tsallis MDPs by adding $S_q^\infty(\pi)$ to the objective function as follows:

$$max_{\pi\in\Pi}\mathbb{E}_{\tau\sim P,\pi}[\Sigma_t^\infty \gamma^t \mathbf{R}_t] + \alpha S_q^\infty(\pi), \tag{7}$$

where $\alpha > 0$ is a coefficient. A state value and state-action value are redefined for Tsallis MDPs as follows: $V_q^\pi(s) \triangleq \mathbb{E}_{\tau\sim P,\pi}[\Sigma_t^\infty \gamma^t(\mathbf{R}_t + \alpha S_q^\infty(\pi(\cdot|s_t))|s_0 = s]$ and $q_q^\pi(s,a) \triangleq \mathbb{E}_{\tau\sim P,\pi}[\mathbf{R}_0 + \Sigma_t^\infty \gamma^t(\mathbf{R}_t + \alpha S_q^\infty(\pi(\cdot|s_t))|s_0 = s, a_0 = a]$, 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.

III. METHOD

A. MUSIC Algorithm

The Mutual Information State Intrinsic Control has four variants as described in [46]. The original MUSIC method is an unsupervised reinforcement learning approach, which is denoted as "MUSIC-u", where "-u" stands for unsupervised learning. Three additional methods are introduced using MUSIC to accelerate learning. The first method is using the MUSIC-u pretrained policy as the parameter initialization and then fine-tuning the agent with the task rewards. This variant is denoted as "MUSIC-f", where "-f" stands for fine-tuning. The second variant is to use the MI intrinsic reward to help the agent to explore more efficiently. Here, the MI reward and the task reward are added together. This method is referred to as "MUSIC-r", where "-r" stands for reward. The third approach is to use the MI quantity from MUSIC to prioritize trajectories for replay. The approach is similar to the TD-error-based prioritized experience replay (PER)[38]. The only difference is that we use the estimated MI instead of the TD-error as the priority for sampling. We name this method as "MUSIC-p", where "-p" stands for prioritization. In this project, experiments will be on MUSIC-r, the intrinsic reward version of the MUSIC algorithm as is shown in Figure 2.

```
Algorithm 1: MUSIC
while not converged do
    Sample an initial state $s_0 \sim p(s_0)$.
    for $t \leftarrow 1$ to steps_per_episode do
        Sample action $a_t \sim \pi_\theta(a_t \mid s_t)$.
        Step environment $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$.
        Sample transitions $\mathcal{T}'$ from the buffer.
        Set intrinsic reward $r = I_\phi(S^s; S^a \mid \mathcal{T}')$.
        Update policy ($\theta$) via DDPG or SAC.
        Update the MI estimator ($\phi$) with SGD.
```

Fig. 2. The MUSIC algorithm

B. Actor-Critic Changes to Tsallis Actor-Critic

Tsallis Actor Critic was implemented using entropic index as a variable. Similarly to SAC, TAC algorithm maintains five networks to model a policy π_ϕ, state value V_ϕ, target state value $V_{\phi-}$, two state action values Q_{θ_1} and Q_{θ_1} . 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,

$$J_\psi = \mathbb{E}_{s_t,a_t\sim\mathcal{B}}[(y_t - V_\psi(s_t))^2/2] \tag{8}$$

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 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,

$$J_\theta = \underset{b_t\sim\mathcal{B}}{\mathbb{E}}[(Q_\theta(s_t, a_t) - r_{t+1} - \gamma V_{\psi^-}(s_{t+1}))^2/2] \tag{9}$$

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

$$J_\phi = \underset{s_t\sim\mathcal{B}}{\mathbb{E}}[\underset{a\sim\pi_\phi}{E}[\alpha \ln_q(\pi_\phi(a|s_t)) - Q_\theta(s_t, a)]] \tag{10}$$

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 [18] 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

$$\nabla_\phi J_\phi = \underset{s_t\sim\mathcal{B}}{\mathbb{E}}[\underset{\epsilon\sim P_\epsilon}{E}[\alpha \nabla_\phi \ln_q(\pi_\phi(a|s_t)) - \nabla_\phi Q_\theta(s_t, a)]] \tag{11}$$

where $a = \mu_\phi + \sigma_{\phi\epsilon}$ and ϵ is a unit normal noise.

The MUSIC algorithm used TAC in the place of Soft Actor-Critic in the original implementation. This included making q, entropic index as an input as a variable to the graph.

IV. EXPERIMENTS

MUSIC-u was trained for 50 epochs with q=1, and then MUSIC-r(the exploration with Mutual Information as reward) was trained for another 50 epochs with FetchPush-v1, FetchPickAndPlace-v1, FetchSlide-v1. TAC+MUSIC showed faster convergence in FetchPush-v1, exceeded performance of FetchPickAndPlace-v1 and FetchSlide-v1 as shown in Figure 4, 5, 6 respectively. q=1.5 was observed to perform well on these task whereas q=2.0 showed reduced performance than q=1.5 and even q=1.0(SAC) as shown in Figure 7.

Algorithm 1 Tsallis Actor Critic (TAC)

```
Input: Total time steps $t_{\max}$, Max episode length $l_{\max}$, Memory size $N$, Entropy coefficient $\alpha$, Entropic index $q$ (or schedule), Moving average ratio $\tau$, Environment $env$
Initialize: $\psi, \psi^-, \theta_1, \theta_2, \phi, \mathcal{D}$ : Queue with size $N$, $t = 0$, $t_e = 0$
while $t \le t_{\max}$ 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 $J_\phi$ 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
```

Fig. 3. The Tsallis Actor Critic algorithm

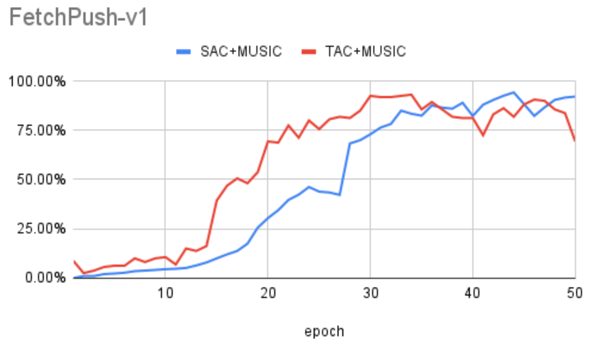

Fig. 4. TAC+MUSIC showed faster convergence to 85% success rate than SAC+MUSIC

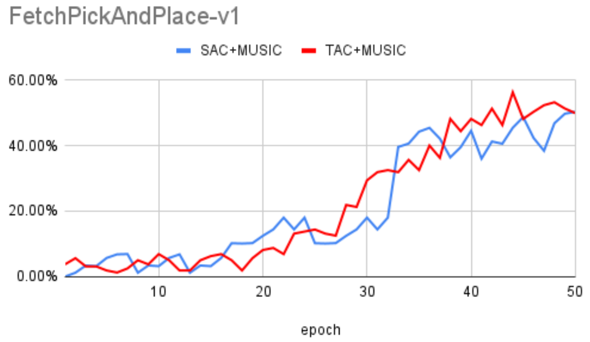

Fig. 5. TAC+MUSIC performance exceeded that of SAC+MUSIC.

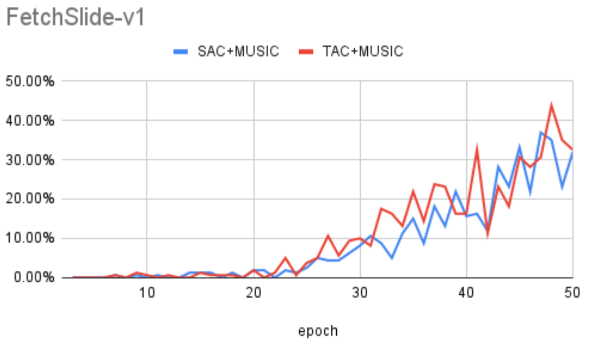

Fig. 6. TAC+MUSIC performance was comparable to that of SAC+MUSIC

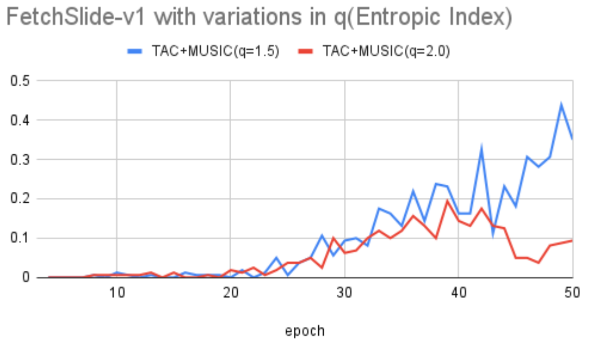

Fig. 7. q=2.0 underperformed q=1.5. This shows that there is an optimal point between q=1.0 and q=2.0 that controls the optimal sparsity of the policy for the robot on robotic manipulation tasks

V. RELATED WORKS

Exploration in reinforcement learning has recently seen huge progress and breakthrough. Exploration-Exploitation has long been searched and studied by RL researchers, but the use of deep learning has boosted its performance and enabled various techniques. Classical exploration strategies include epsilon-greedy where the agent takes a random action with probability of ϵ, upper confidence bounds where the agent selects the greediest action to maximize the upper confidence bound of the q-function and thompson sampling where the agent keeps track of a belief over the distribution of optimal actions and samples from this distribution.

After the introduction of neural network as a function approximators, [18] introduced and entropy term $H(\pi(a|s))$ into the loss function, encouraging the policy to take diverse actions when given a state. [36] and [15] introduced noise into the action, observation and also the parameter space to increase uncertainty and encourage exploration. Along the lines of upper confidence bounds, uncertainty has also been a crucial factor in exploration. [30] uses ensembles network and uses its output to measure uncertainty of q-value for each action and choose the one with the highest uncertainty.

Intrinsic motivation which was first shed light on from psychological viewpoint [32] has provided a great insight into how an agent could learn in a sparse reward environment. [23] proposed the idea of empowerment of the agent as a measure for intrinsic motivation. Started as a thought experiment in [11] the Noisy-TV problem has posed an important question of how to get past noisy unpredictable environment to make more meaningful progress by itself. Using density models for the count, [9] [31] proposed counting the number of states the agent has visited and using it as additional intrinsic motivation. The less it has visited in the past, the more chance the agent has to take action towards such state. [31] used PixelCNN as density models. Hashing after counting was proposed which used locality-sensitive hashing in [42] and autoencoders were used for raw pixel inputs.

Prediction was also widely used to generate intrinsinc motivation. [41] trained a forward dynamics model in the encoding space and the encoding function was learned via an autoencoder. In an attempt to replace autoencoders, [33] proposed Intrinsic Curiosity Module(ICM) which learns the state space encoding with a self-supervised inverse dynamics model. [10] showed that the idea of pure intrinsic motivation could be sufficient to provide compact, sufficient and stable forward dynamics of the environment. Variational Information(Information Gain) of the forward dynamics model was maximized in [22] to provide intrinsic motivation. KL-Divergence was used upon computing the intrinsic reward. Multiple forward dynamics was used in [34] to use disagreements between the models as intrinsic reward, which was differentiable thus allowing gradient descent through the ensembles. [11] uses random network distillation(RND) to capture the error between the predicting neural network $\hat{f}(s_t)$ and the fixed randomly initialized neural network $f(s_t)$. If the discrepancy is high it means the state the agent is currently in is novel and needs to be further explored. Such idea was built upon by [6] and [5] where the former structured the intrinsic reward in two levels: episodic reward utilized random network distillation as lifelong novelty reward and inverse dynamics feature to compute episodic reward. [6] used LSTM[21] and further utilized Recurrent Experience Replay[24] to store episodic memories. [5] added multi-armed bandits to incorporate universal value function approximators[19] for controlling the ratio between lifelong novelty and episodic novelty.

Mutual Information was shed light upon as a tool for exploration by [7], [20], [44]. In MUSIC[46], MI quantity $I(S^s; S^a)$ was approximated using lower bound in the Donsker-Varadhan representation with the compression lemma in the PAC-Bayes literature. [25] uses mutual information as intrinsic motivation and has seen boosts in performance in Atari-games and continuous control tasks.[13] proposed feature control as intrinsic motivation and shows stat-of-the-art results in montezuma's revenge.

unsupervised learning provides a fresh view on learning skills and using goal-conditioned learning. [14] was successful at discovering skills that are task-agnostic, which could be further used for specific tasks. [17], [12] is also succeeded in extracting meaningful skills in a given environment. Intrinsic motivation was also utilized in learning goal-conditioned policies. DISCERN[45] learns a MI objective between states and goals. MUSIC[46] was combined with DISCERN to gather mutual information between the surrounding state and the agent state, and mutual information between the state overall and the goal.

Sparse Markov Decision Process(MDP) was proposed in [27] as a generalized form of Shannon-Gibbs entropy. Sparse MDP generalizes the concept with and index called entropic index, q , which enables the the sparsity of the policy to be manipulated. Tsallis entropy regularization was fully introduced in reinforcement learning with [28], in which was provided dynamic programming methods for Tsallis MDPs and Tsallis Actor Critic, which various experimental results that provided improvements with linear increasing of q, the entropic index.

VI. CONCLUSIONS

Exploration in a reinforcement learning environment is a challenging task. Although many exploration tasks were proposed to learn useful representation or the dynamics model that could guide the agent for a better exploration scheme and succeeded, most exploration research were conducting in Atari-game environments or continuous control problem. Robotic manipulation tasks pose greater challenges than the currently focused exploration environments in that the reward is very sparse in terms of the time horizon, but the entropy involved in the policy is still high, which is not optimal considering the fact that most robotic manipulation tasks do have few - but not one - optimal trajectories that performs the task successfully. We empirically show that changes in the entropic index could lead to higher sparsity in the final policy of the agent, thereby ensuring that the robot take the most efficient and effective actions towards accomplishment of the task. Future work still remains for a further careful manipulation of the entropic index, q, so that the agent could control the pace at which the policy is learning a stochastic policy. Moreover, incorporating the level of uncertainty of actions using mutual information and exploring using sets of skills discovered with MUSIC-p could also aid in exploring the environment.

REFERENCES

[1] Barrett Ames, A. Thackston, and G. Konidaris. Learning symbolic representations for planning with parameterized skills. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 526–533, 2018.

[2] Marcin Andrychowicz, Dwight Crow, Alex Ray, J. Schneider, Rachel Fong, P. Welinder, Bob McGrew, Joshua Tobin, P. Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NIPS*, 2017.

[3] Marcin Andrychowicz, Dwight Crow, Alex Ray, J. Schneider, Rachel Fong, P. Welinder, Bob McGrew, Joshua Tobin, P. Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *NIPS*, 2017.

[4] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics Auton. Syst.*, 57:469–483, 2009.

[5] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, P. Sprechmann, Alex Vitvitskyi, Daniel Guo, and C. Blundell. Agent57: Outperforming the atari human benchmark. *ArXiv*, abs/2003.13350, 2020.

[6] Adrià Puigdomènech Badia, P. Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, O. Tieleman, Martín Arjovsky, A. Pritzel, Andew Bolt, and C. Blundell. Never give up: Learning directed exploration strategies. *ArXiv*, abs/2002.06038, 2020.

[7] Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R. Devon Hjelm, and Aaron C. Courville. MINE: mutual information neural estimation. *CoRR*, abs/1801.04062, 2018.

[8] Marc G. Bellemare, Yavar Naddaf, J. Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents (extended abstract). In *IJCAI*, 2015.

[9] Marc G. Bellemare, S. Srinivasan, Georg Ostrovski, Tom Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016.

[10] Yuri Burda, Harrison Edwards, Deepak Pathak, A. Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. *ArXiv*, abs/1808.04355, 2019.

[11] Yuri Burda, Harrison Edwards, A. Storkey, and Oleg Klimov. Exploration by random network distillation. *ArXiv*, abs/1810.12894, 2019.

[12] Víctor Campos, Alexander Trott, Caiming Xiong, R. Socher, Xavier Giro i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *ICML*, 2020.

[13] Nat Dilokthanakul, Christos Kaplanis, Nick Pawlowski, and M. Shanahan. Feature control as intrinsic motivation for hierarchical reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30:3409–3418, 2019.

[14] Benjamin Eysenbach, A. Gupta, J. Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *ArXiv*, abs/1802.06070, 2019.

[15] Meire Fortunato, M. G. Azar, Bilal Piot, Jacob Menick, Ian Osband, A. Graves, Vlad Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg. Noisy networks for exploration. *ArXiv*, abs/1706.10295, 2018.

[16] Justin Fu, Aviral Kumar, Ofir Nachum, G. Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *ArXiv*, abs/2004.07219, 2020.

[17] K. Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *ArXiv*, abs/1611.07507, 2017.

[18] Tuomas Haarnoja, Aurick Zhou, P. Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

[19] Danijar Hafner, T. Lillicrap, Ian S. Fischer, Ruben Villegas, David R Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *ArXiv*, abs/1811.04551, 2019.

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization, 2019.

[21] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[22] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, F. Turck, and P. Abbeel. Vime: Variational information maximizing exploration. In *NIPS*, 2016.

[23] T. Jung, D. Polani, and P. Stone. Empowerment for continuous agent—environment systems. *Adaptive Behavior*, 19:16 – 39, 2011.

[24] Steven Kapturowski, Georg Ostrovski, John Quan, R. Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *ICLR*, 2019.

[25] HyoungSeok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. Emi: Exploration with mutual information. In *ICML*, 2019.

[26] Oliver Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.*, 22:30:1–30:82, 2021.

[27] 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*, 3:1466–1473, 2018.

[28] Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, and Songhwai Oh. Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning. *ArXiv*, abs/1902.00137, 2019.

[29] T. Lillicrap, Jonathan J. Hunt, A. Pritzel, N. Heess, T. Erez, Yuval Tassa, D. Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016.

[30] Ian Osband, C. Blundell, A. Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *NIPS*, 2016.

[31] Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and R. Munos. Count-based exploration with neural density models. *ArXiv*, abs/1703.01310, 2017.

[32] Pierre-Yves Oudeyer, F. Kaplan, and V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11:265–286, 2007.

[33] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 488–489, 2017.

[34] Deepak Pathak, Dhiraj Gandhi, and A. Gupta. Self-supervised exploration via disagreement. *ArXiv*, abs/1906.04161, 2019.

[35] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, J. Schneider, Joshua Tobin, Maciek Chociej, P. Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *ArXiv*, abs/1802.09464, 2018.

[36] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, S. Sidor, Richard Y. Chen, Xi Chen, T. Asfour, P. Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *ArXiv*, abs/1706.01905, 2018.

[37] Rajesh D. Savatekar and Mr. A. A. Dum. Design of control system for articulated robot using leap motion sensor. 2016.

[38] Tom Schaul, John Quan, Ioannis Antonoglou, and D. Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2016.

[39] John Schulman, Sergey Levine, P. Abbeel, Michael I. Jordan, and P. Moritz. Trust region policy optimization. *ArXiv*, abs/1502.05477, 2015.

[40] John Schulman, F. Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[41] Bradly C. Stadie, Sergey Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *ArXiv*, abs/1507.00814, 2015.

[42] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, F. Turck, and P. Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In *NIPS*, 2017.

[43] E. Todorov, T. Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[44] Petar Velickovic, Rex Ying, Matilde Padovano, R. Hadsell, and C. Blundell. Neural execution of graph algorithms. *ArXiv*, abs/1910.10593, 2020.

[45] David Warde-Farley, T. Wiele, Tejas D. Kulkarni, Catalin Ionescu, S. Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *ArXiv*, abs/1811.11359, 2019.

[46] Rui Zhao, Yang Gao, P. Abbeel, Volker Tresp, and W. Xu. Mutual information state intrinsic control. *ArXiv*, abs/2103.08107, 2021.