Path Consistency Learning in Tsallis Entropy Regularized MDPs

Ofir Nachum [*1] **Yinlam Chow** [*2] **Mohamamd Ghavamzadeh** [*2]

Abstract

We study the sparse entropy-regularized reinforcement learning (ERL) problem in which the entropy term is a special form of the *Tsallis* entropy. The optimal policy of this formulation is sparse, i.e., at each state, it has non-zero probability for only a small number of actions. This addresses the main drawback of the standard Shannon entropy-regularized RL (soft ERL) formulation, in which the optimal policy is *softmax*, and thus, may assign a non-negligible probability mass to non-optimal actions. This problem is aggravated as the number of actions is increased. In this paper, we follow the work of Nachum et al. (2017) in the soft ERL setting, and propose a class of novel path consistency learning (PCL) algorithms, called *sparse PCL*, for the sparse ERL problem that can work with both on-policy and off-policy data. We first derive a *sparse consistency* equation that specifies a relationship between the optimal value function and policy of the sparse ERL along any system trajectory. Crucially, a weak form of the converse is also true, and we quantify the sub-optimality of a policy which satisfies sparse consistency, and show that as we increase the number of actions, this sub-optimality is better than that of the soft ERL optimal policy. We then use this result to derive the sparse PCL algorithms. We empirically compare sparse PCL with its soft counterpart, and show its advantage, especially in problems with a large number of actions.

1. Introduction

In reinforcement learning (RL), the goal is to find a policy with maximum long-term performance, defined as the sum of discounted rewards generated by following the policy (Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998). In case the number of states and actions are small, and the model is known, the optimal policy is the solution of the *non-linear* Bellman optimality equations (Bellman, 1957). When the system is large or the model is unknown, greedily solving the Bellman equations often results in policies that are far from optimal. A principled way of dealing with this issue is *regularization*. Among different forms of regularization, such as ℓ_2 (e.g., Farahmand et al. 2008; 2009) and ℓ_1 (e.g., Kolter & Ng 2009; Johns et al. 2010; Ghavamzadeh et al. 2011), *entropy regularization* is among the most studied in both value-based (e.g., Kappen 2005; Todorov 2006; Ziebart 2010; Azar et al. 2012; Fox et al. 2016; O'Donoghue et al. 2017; Asadi & Littman 2017) and policy-based (e.g., Peters et al. 2010; Todorov 2010) RL formulations. In particular, two of the most popular deep RL algorithms, TRPO (Schulman et al., 2015) and A3C (Mnih et al., 2016), are based on entropy-regularized policy search. We refer the interested readers to Neu et al. (2017), for an insightful discussion on entropy-regularized RL algorithms and their connection to online learning.

In entropy-regularized RL (ERL), an entropy term is added to the Bellman equation. This formulation has four main advantages: **1)** it softens the non-linearity of the Bellman equations and makes it possible to solve them more easily, **2)** the solution of the softened problem is quantifiably not much worse than the optimal solution in terms of accumulated return, **3)** the addition of the entropy term brings nice properties, such as encouraging exploration (Shannon entropy) (e.g., Fox et al. 2016; Nachum et al. 2017) and maintaining a close distance to a baseline policy (relative entropy) (e.g., Schulman et al. 2015; Belousov & Peters 2017; Nachum et al. 2018), and **4)** unlike the original problem that has a deterministic solution, the solution to the softened problem is stochastic, which is preferable in problems in which exploration or dealing with unexpected situations is important. However, in the most common form of ERL, in which a Shannon (or relative) entropy term is added to the Bellman equations, the optimal policy is of the form of *softmax*. Despite the advantages of a softmax policy in terms of exploration, its main drawback is that at each step, it assigns a non-negligible probability mass to non-optimal actions, a problem that is aggravated as the number of actions is increased. This may result in policies that may not be safe to execute. To address this issue, Lee et al. (2018) proposed to add a special form of a general no-

[*]Equal contribution [1]Google Brain [2]DeepMind. Correspondence to: Yinlam Chow <yinlamchow@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

tion of entropy, called Tsallis entropy (Tsallis, 1988), to the Bellman equations. This formulation has the property that its solution has sparse distributions, i.e., at each state, only a small number of actions have non-zero probability. Lee et al. (2018) studied the properties of this ERL formulation, proposed value-based algorithms (fitted Q-iteration and Q-learning) to solve it, and showed that although it is harder to solve than its soft counterpart, it potentially has a solution closer to that of the original problem.

In this paper, we propose novel path consistency learning (PCL) algorithms for the Tsallis ERL problem, called *sparse PCL*. PCL is a class of actor-critic type algorithms developed by Nachum et al. (2017) for the soft (Shannon entropy) ERL problem. It uses a nice property of soft ERL, namely the equivalence of consistency and optimality, and learns parameterized policy and value functions by minimizing a loss that is based on the consistency equation of soft ERL. The most notable feature of soft PCL is that it can work with both on-policy (sub-trajectories generated by the current policy) and off-policy (sub-trajectories generated by a policy different than the current one, including any sub-trajectory from the replay buffer) data. We first derive a multi-step consistency equation for the Tsallis ERL problem, called *sparse consistency*. We then prove that in this setting, while optimality implies consistency (similar to the soft case), unlike the soft case, consistency only implies sub-optimality. We then use the sparse consistency equation and derive PCL algorithms that use both on-policy and off-policy data to solve the Tsallis ERL problem. We empirically compare sparse PCL with its soft counterpart. As expected, we gain from using the sparse formulation when the number of actions is large, both in algorithmic tasks and in discretized continuous control problems.

2. Markov Decision Processes (MDPs)

We consider the reinforcement learning (RL) problem in which the agent's interaction with the system is modeled as a MDP. A MDP is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, r, P, P_0, \gamma)$, where $\mathcal{X}$ and $\mathcal{A}$ are state and action spaces; $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and $P : \mathcal{X} \times \mathcal{A} \to \Delta_{\mathcal{X}}$ are the reward function and transition probability distribution, with $r(x, a) \in [0, R_{\max}]$ and $P(\cdot|x, a)$ being the reward and the next state probability of taking action a in state x; $P_0 : \mathcal{X} \to \Delta_{\mathcal{X}}$ is the initial state distribution; and $\gamma \in [0, 1)$ is a discounting factor. In this paper, we assume that the action space is finite, but can be large. The goal in RL is to find a stationary Markovian policy, i.e., a mapping from state and action spaces to a simplex over the actions $\mu : \mathcal{X} \times \mathcal{A} \to \Delta_{\mathcal{A}}$, that maximizes the expected discounted sum of rewards, i.e.,

$$\max_{\mu} \ \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)\big] \tag{1}$$
$$\text{s.t.} \quad \forall x \ \sum_{a \in \mathcal{A}} \mu(a|x) = 1, \quad \forall x, a \ \ \mu(a|x) \geq 0,$$

where $x_0 \sim P_0$, $a_t \sim \mu(\cdot|x_t)$, and $x_{t+1} \sim P(\cdot|x_t, a_t)$. For a given policy μ, we define its value and action-value functions as

$$V^{\mu}(x) = \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)|x_0 = x, \mu, P\big],$$
$$Q^{\mu}(x, a) = \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)|x_0 = x, a_0 = a, \mu, P\big].$$

Any solution of the optimization problem (1) is called an *optimal* policy and is denoted by μ^*. Note that while a MDP may have several optimal policies, it only has a single optimal value function $V^* = V^{\mu^*}$. It has been proven that (1) has a solution in the space of *deterministic* policies, i.e., $\Pi_d = \{\mu : \mu : \mathcal{X} \to \mathcal{A}\}$, which can be obtained as the *greedy* action w.r.t. the optimal action-value function, i.e., $\mu^*(x) \in \arg\max_a Q^*(x, a)$ (Puterman, 1994; Bertsekas & Tsitsiklis, 1996). The optimal action-value function Q^* is the *unique* solution of the non-linear Bellman optimality equations, i.e., for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$,

$$Q(x, a) = r(x, a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, a) \max_{a' \in \mathcal{A}} Q(x', a'). \tag{2}$$

Any optimal policy μ^* and the optimal state and state-action value functions, V^* and Q^*, satisfy the following equations for all states and action,

$$Q^*(x, a) = r(x, a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, a) V^*(x'),$$
$$V^*(x) = \max_{a \in \mathcal{A}} Q^*(x, a), \qquad \mu^*(x) \in \arg\max_{a \in \mathcal{A}} Q^*(x, a).$$

3. Entropy Regularized MDPs

As discussed in Section 2, finding an optimal policy for a MDP involves solving a non-linear system of equations (see Eq. 2), which is often complicated. Moreover, the optimal policy may be deterministic, always selecting the same optimal action at a state even when there are several optimal actions in that state. This is undesirable when it is important to explore and to deal with unexpected situations. In such cases, one might be interested in multimodal policies that still have good performance. This is why many researchers have proposed to add a regularizer in the form of an *entropy* term to the objective function (1) and solve the following *entropy-regularized* optimization problem

$$\max_{\mu} \ \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \big(r(x_t, a_t) + \alpha H^{\mu}(x_t, a_t)\big)\Big] \tag{3}$$
$$\text{s.t.} \quad \forall x \ \sum_{a \in \mathcal{A}} \mu(a|x) = 1, \quad \forall x, a \ \ \mu(a|x) \geq 0,$$

where $H^{\mu}(x, a)$ is an entropy-related term and α is the regularization parameter. The entropy term smoothens the objective function (1) such that the resulting problem (3) is often easier to solve than the original one (1). This is another reason for the popularity of entropy-regularized MDPs.

3.1. Entropy Regularized MDP with Shannon Entropy

It is common to use $H^{\mu}_{\text{sf}}(x_t, a_t) \triangleq -\log \mu(a_t|x_t)$ in entropy-regularized MDPs (e.g., Fox et al. 2016; Nachum et al. 2017). Note that $H_{\text{sf}}(\mu) = \mathbb{E}_{\mu}\big[H^{\mu}_{\text{sf}}(x,a)\big]$ is the *Shannon entropy*. Problem (3) with $H^{\mu}_{\text{sf}}(x,a)$ can be seen as a RL problem in which the reward function is the sum of the original reward function $r(x,a)$ and a term that encourages *exploration*.[1] Unlike (1), the optimization problem (3) with H^{μ}_{sf} has a unique optimal policy μ^*_{sf} and a unique optimal value V^*_{sf} (action-value Q^*_{sf}) function that satisfy the following equations:

$$\begin{aligned} Q^*_{\text{sf}}(x,a) &= r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) V^*_{\text{sf}}(x'), \\ V^*_{\text{sf}}(x) &= \alpha \cdot \text{sfmax}\big(Q^*_{\text{sf}}(x,\cdot)/\alpha\big), \\ \mu^*_{\text{sf}}(a|x) &= \frac{\exp\big(Q^*_{\text{sf}}(x,a)/\alpha\big)}{\sum_{a' \in \mathcal{A}} \exp\big(Q^*_{\text{sf}}(x,a')/\alpha\big)}, \end{aligned} \tag{4}$$

where for any function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, the sfmax operator is defined as $\text{sfmax}\big(f(x,\cdot)\big) = \log\big(\sum_a \exp\big(f(x,a)\big)\big)$. Note that the equations in (4) are derived from the KKT conditions of (3) with H^{μ}_{sf}. In this case, the optimal policy is *soft-max*, with the regularization parameter α playing the role of its temperature (see Eq. 4). This is why (3) with H^{μ}_{sf} is called the *soft MDP* problem. In soft MDPs, the optimal value function V^*_{sf} is the unique solution of the *soft Bellman optimality* equations, i.e., $\forall x \in \mathcal{X}, \forall a \in \mathcal{A}$,

$$V(x) = \alpha \cdot \text{sfmax}\Big(\big[r(x,\cdot) + \gamma \sum_{x'} P(x'|x,\cdot) V(x')\big]/\alpha\Big). \tag{5}$$

Note that the sfmax operator is a smoother function of its inputs than the $\max$ operator associated with the Bellman optimality equation (2). This means that solving the soft MDP problem is easier than the original one, with the cost that its optimal policy μ^*_{sf} performs worse than the optimal policy of the original MDP μ^*. This difference can be quantified as

$$\forall x \in \mathcal{X} \quad V^*(x) - \frac{\alpha}{1-\gamma}\log(|\mathcal{A}|) \le V^{\mu^*_{\text{sf}}}(x) \le V^*(x), \tag{6}$$

where we discriminate between the value function of a policy μ in the soft V^{μ}_{sf} and original V^{μ} MDPs. Note that the sub-optimality of μ^*_{sf} is unbounded as $|\mathcal{A}| \to \infty$. This is the main drawback of using softmax policies; in large action space problems, at each step, the policy assigns a non-negligible probability mass to non-optimal actions, which in aggregate can be detrimental to its reward performance.

[1] Another entropy term that has been studied in the literature is $H^{\mu}_{\text{rel}}(x_t, a_t) \triangleq -\log \frac{\mu(a_t|x_t)}{\mu_b(a_t|x_t)}$, where π_b is a baseline policy. Note that $H_{\text{rel}}(\mu) = \mathbb{E}_{\mu}\big[H^{\mu}_{\text{rel}}(x,a)\big]$ is the *relative entropy*. Problem (3) with $H^{\mu}_{\text{rel}}(x,a)$ can be seen as a RL problem in which the reward function is the sum of the original reward function $r(x,a)$ and a term that penalizes deviation from the baseline policy π_b.

3.2. Entropy Regularized MDP with Tsallis Entropy

To address the issues with the softmax policy, Lee et al. (2018) proposed to use $H^{\mu}_{\text{sp}}(x_t, a_t) \triangleq \frac{1}{2}\big(1 - \mu(a_t|x_t)\big)$ in entropy-regularized MDPs. Note that $H_{\text{sp}}(\mu) = \mathbb{E}_{\mu}\big[H^{\mu}_{\text{sp}}(x,a)\big]$ is a special case of a general notion of entropy, called *Tsallis entropy* (Tsallis, 1988), i.e., $S_{q,k}(p) = \frac{k}{q-1}(1 - \sum_i p_i^q)$, for the parameters $q = 2$ and $k = \frac{1}{2}$.[2] Similar to the soft MDP problem, the optimization problem (3) with H^{μ}_{sp} has a unique optimal policy μ^*_{sp} and a unique optimal value V^*_{sp} (action-value Q^*_{sp}) function that satisfy the following equations (Lee et al., 2018):

$$\begin{aligned} Q^*_{\text{sp}}(x,a) &= r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) V^*_{\text{sp}}(x'), \\ V^*_{\text{sp}}(x) &= \alpha \cdot \text{spmax}\big(Q^*_{\text{sp}}(x,\cdot)/\alpha\big), \\ \mu^*_{\text{sp}}(a|x) &= \Big(Q^*_{\text{sp}}(x,a)/\alpha - \mathcal{G}\big(Q^*_{\text{sp}}(x,\cdot)/\alpha\big)\Big)^+, \end{aligned} \tag{7}$$

where $(\cdot)^+ = \max(\cdot, 0)$, and for any function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, the spmax operator is defined as

$$\text{spmax}\big(f(x,\cdot)\big) = \frac{1}{2}\Big[1 + \sum_{a \in \mathcal{S}(x)} \Big(\Big(\frac{f(x,a)}{\alpha}\Big)^2 - \mathcal{G}\Big(\frac{f(x,\cdot)}{\alpha}\Big)^2\Big)\Big],$$

in which $\mathcal{G}\big(f(x,\cdot)\big) = \frac{\sum_{a \in \mathcal{S}(x)} f(x,a) - 1}{|\mathcal{S}(x)|}$ and $\mathcal{S}(x)$ is the set of actions satisfying $1 + i\frac{f(x,a_{(i)})}{\alpha} > \sum_{j=0}^{i} \frac{f(x,a_{(j)})}{\alpha}$, where $a_{(i)}$ indicates the action with the i'th largest value of $f(x,a)$. Note that the equations in (7) are derived from the KKT conditions of (3) with H^{μ}_{sp}. In this case, the optimal policy may have zero probability for several actions (see Eq. 7). This is why (3) with H^{μ}_{sp} is called the *sparse MDP* problem. The regularization parameter α controls the sparsity of the resulted policy. The policy would be more sparse for smaller values of α. In sparse MDPs, the optimal value function V^*_{sp} is the unique fixed-point of the *sparse Bellman optimality* operator $\mathcal{T}_{\text{sp}}$ (Lee et al., 2018) that for any function $f : \mathcal{X} \to \mathbb{R}$ is defined as

$$(\mathcal{T}_{\text{sp}} f)(x) = \alpha \cdot \text{spmax}\Big(\big[r(x,\cdot) + \gamma \sum_{x'} P(x'|x,\cdot) f(x')\big]/\alpha\Big). \tag{8}$$

Similar to (5), the spmax operator is a smoother function of its inputs than the $\max$, and thus, solving the sparse MDP problem is easier than the original one, with the cost that its optimal policy μ^*_{sp} performs worse than the optimal policy of the original MDP μ^*. This difference can be quantified as (Lee et al., 2018),

$$\forall x \in \mathcal{X} \quad V^*(x) - \frac{\alpha}{1-\gamma} \cdot \frac{|\mathcal{A}| - 1}{2|\mathcal{A}|} \le V^{\mu^*_{\text{sp}}}(x) \le V^*(x). \tag{9}$$

[2] Note that the Shannon entropy is a special case of the Tsallis entropy for the parameters $q = k = 1$ (Tsallis, 1988).

On the other hand, the spmax operator is more complex than sfmax, and thus, it is slightly more complicated to solve the sparse MDP problem than its soft counterpart. However, as can be seen from Eqs. 6 and 9, the optimal policy of the sparse MDP, μ^*_{sp}, can have a better performance than its soft counterpart, μ^*_{sf}, and this difference becomes more apparent as the number of actions $|\mathcal{A}|$ grows. For large action size, the term $(|\mathcal{A}|-1)/(2|\mathcal{A}|)$ in (9) turns to a constant, while $\log|\mathcal{A}|$ in (6) grows unbounded.[3]

4. Path Consistency Learning in Soft MDPs

A nice property of soft MDPs that was elegantly used by Nachum et al. (2017) is that any policy μ and function $V:\mathcal{X}\rightarrow\mathbb{R}$ that satisfy the (one-step) *consistency* equation, i.e., for all $x\in\mathcal{X}$ and for all $a\in\mathcal{A}$,

$$V(x)=r(x,a)-\alpha\log\mu(a|x)+\gamma\sum_{x'\in\mathcal{X}}P(x'|x,a)V(x'), \quad (10)$$

are optimal, i.e., $\mu=\mu^*_{\text{sf}}$ and $V=V^*_{\text{sf}}$ (*consistency implies optimality*). Due to the uniqueness of the optimal policy in soft MDPs, the reverse is also true, i.e., the optimal policy μ^*_{sp} and the value function V^*_{sp} satisfy the *consistency* equation (*optimality implies consistency*).

As shown in Nachum et al. (2017), the (one-step) consistency equation (10) can be easily extended to multi-step, i.e., any policy μ and function $V:\mathcal{X}\rightarrow\mathbb{R}$ that for any state x_0 and sequence of actions $a_0,\ldots,a_{d-1}$, satisfy the *multi-step consistency* equation

$$V(x_0)=\mathbb{E}_{x_{1:d}|x_0,a_{0:d-1}}\Big[\gamma^d V(x_d) + \sum_{t=0}^{d-1}\gamma^t\big(r(x_t,a_t)-\alpha\log\mu(a_t|x_t)\big)\Big] \quad (11)$$

are optimal, i.e., $\mu=\mu^*_{\text{sf}}$ and $V=V^*_{\text{sf}}$.

The property that both single and multiple step consistency equations imply optimality (Eqs. 10 and 11) was the motivation of a RL algorithm by Nachum et al. (2017), *path consistency learning* (PCL). The main idea of (soft) PCL is to learn a parameterized policy and value function by minimizing the following objective function: $\mathcal{J}(\theta,\phi)=\frac{1}{2}\sum_{\xi_i}J(\xi_i,\theta,\phi)^2$, where $\xi=(x_0,a_0,r_0,\ldots,x_{d-1},a_{d-1},r_{d-1},x_d)$ is any d-length sub-trajectory, θ and ϕ are the policy and value function parameters, respectively, and

$$J(\xi,\theta,\phi)=-V_\phi(x_0)+\gamma^d V_\phi(x_d) + \sum_{t=0}^{d-1}\gamma^t\big(r(x_t,a_t)-\alpha\log\mu_\theta(a_t|x_t)\big). \quad (12)$$

[3]When $\alpha=O(1/|A|)$ in soft MDP, theoretically it has a bounded performance when $|A|>>1$, but this would potentially lead to unstable learning as the magnitude of $Q(\cdot,\cdot)/\alpha$ blows up.

An important property of the soft PCL algorithm is that since the multi-step consistency (11) holds for any d-length sub-trajectory, it can use both on-policy (ξ's generated by the current policy μ_θ) and off-policy data, i.e., ξ's generated by a policy different than the current one, including any d-length sub-trajectory from the replay buffer.

Note that since both optimal policy μ^*_{sf} and value function V^*_{sf} can be written based on the optimal action-value function Q^*_{sf} (see Eq. 4), we may write the objective function (12) based on Q_ψ, and optimize only one set of parameters ψ, instead of separate θ and ϕ.

5. Consistency between Optimal Value & Policy in Sparse MDPs

This section begins the main contributions of our work. We first identify a (one-step) consistency equation for the sparse MDPs defined by (3). We then prove the relationship between the *sparse consistency* equation and the optimal policy and value function of the sparse MDP, and highlight its similarities and differences with that in soft MDPs, discussed in Section 4. We then extend the sparse consistency equation to multiple steps and prove results that allow us to use the *multi-step sparse consistency* equation to derive on-policy and off-policy algorithms to solve sparse MDPs, which we fully describe in Section 6. The significance of the sparse consistency equation is in providing an efficient tool for computing a *near-optimal* policy for sparse MDPs, which only involves solving a set of linear equations and linear complementary constraints, as opposed to (iteratively) solving the fixed-point of the non-linear sparse Bellman operator (8). We report the proofs of all the theorems of this section in Appendix A.

For any policy μ and value function $V:\mathcal{X}\rightarrow\mathbb{R}$, we define the (one-step) consistency equation of the sparse MDPs as, for all state $x\in\mathcal{X}$ and for all actions $a\in\mathcal{A}$,

$$V(x)=r(x,a)+\frac{\alpha}{2}-\alpha\mu(a|x)+\lambda(a|x)-\Lambda(x) + \gamma\sum_{x'}P(x'|x,a)V(x'), \quad (13)$$

where $\lambda:\mathcal{X}\times\mathcal{A}\rightarrow\mathbb{R}_+$ and $\Lambda:\mathcal{X}\rightarrow\mathbb{R}_-$ are Lagrange multipliers, such that $\lambda(a|x)\cdot\mu(a|x)=0$ and $-\frac{\alpha}{2}\leq\Lambda(x)\leq 0$. We call this the *one-step sparse consistency* equation and is equivalent to Eq. 10 in soft MDPs.

We now present a theorem which states that, similar to soft MDPs, optimality in sparse MDPs is a necessary condition for consistency, i.e., *optimality implies consistency*.

Theorem 1. *The optimal policy μ^*_{sp} and value function V^*_{sp} of the sparse MDP (3) satisfy the consistency equation (13).*

Theorem 2 shows that in the sparse MDPs, *consistency only implies near-optimality*, as opposed to optimality in the case of soft MDPs.

Theorem 2. *Any policy μ that satisfies the consistency equation (13) is $\alpha/(1-\gamma)$-optimal in the sparse MDP (3), i.e., for each state $x \in \mathcal{X}$, we have $V_{sp}^{\mu}(x) \geq V_{sp}^{*}(x) - \frac{\alpha}{1-\gamma}$.*

This result indicates that for a fixed γ, as α decreases, a policy μ satisfying the one-step consistency equations approaches the true optimal μ_{sp}^{*}. To connect the performance of μ to the original goal of maximizing expected return, we present the following corollary, which is a direct consequence of Theorem 2 and the results reported in Section 3.2 on the performance of μ_{sp}^{*} in the original MDP.

Corollary 1. *Any policy μ that satisfies the consistency equation (13) is $(\frac{3}{2} - \frac{1}{2|\mathcal{A}|}) \cdot \frac{\alpha}{1-\gamma}$-optimal in the original MDP (1), i.e., for each state $x \in \mathcal{X}$, we have $V^{*}(x) - \left(\frac{3}{2} - \frac{1}{2|\mathcal{A}|}\right) \cdot \frac{\alpha}{1-\gamma} \leq V^{\mu}(x) \leq V^{*}(x)$.*

We now extend the (one-step) sparse consistency equation (13) to multiple steps. For any state $x_0 \in \mathcal{X}$ and sequence of actions $a_0, \ldots, a_{d-1}$, define the multi-step consistency equation for sparse MDPs as

$$V(x_0) = \mathbb{E}_{x_{1:d}|x_0,a_{0:d-1}}\Big[\gamma^d V(x_d) + \sum_{t=0}^{d-1}\gamma^t\big(r(x_t,a_t) + \frac{\alpha}{2} - \alpha\mu(a_t|x_t) + \lambda(a_t|x_t) - \Lambda(x_t)\big)\Big], \tag{14}$$

where $\lambda : \mathcal{X} \times \mathcal{A} \to \mathbb{R}_{+}$ and $\Lambda : \mathcal{X} \to \mathbb{R}_{-}$ are Lagrange multipliers, such that $\lambda(a|x) \cdot \mu(a|x) = 0$ and $-\frac{\alpha}{2} \leq \Lambda(x) \leq 0$. We call this *multi-step sparse consistency* equation, the equivalent of Eq. 11 in soft MDPs.

From Theorem 1, we can immediately show that multi-step sparse consistency is a necessary condition of optimality.

Corollary 2. *The optimal policy μ_{sp}^{*} and value function V_{sp}^{*} of the sparse MDP (3) satisfy the multi-step consistency equation (14).*

Conversely, followed from Theorem 2, we prove the following result on the performance of any policy satisfying the mutli-step consistency equation. This is a novel result showing that solving the multi-step consistency equation is indeed *sufficient* to guarantee near-optimality.

Corollary 3. *Any policy μ that satisfies the consistency equation (14) is $\alpha/(1-\gamma)$-optimal in sparse MDP (3).*

Equipped with the above results on the relationship between (near)-optimality and multi-step consistency in sparse MDPs, we are now ready to present our off-policy RL algorithms to solve the sparse MDP (3).

6. Path Consistency Learning in Sparse MDPs

Similar to the PCL algorithm for soft MDPs, in sparse MDPs the multi-step consistency equation (14) naturally leads to a path-wise algorithm for training a policy μ_θ and value function V_ϕ parameterized by θ and ϕ, as well as Lagrange multipliers Λ_ρ and $\lambda_{\theta,\rho}$ parameterized by the auxiliary parameter ρ. To characterize the objective function of this algorithm, we first define the *soft consistency error* for the d-step sub-trajectory ξ as a function of θ, ρ, and ϕ,

$$J(\xi;\theta,\rho,\phi) = -V_\phi(x_0) + \gamma^d V_\phi(x_d) + \sum_{t=0}^{d-1}\gamma^j\big(r(x_t,a_t) + \frac{\alpha}{2} - \alpha\mu_\theta(a_t|x_t) + \lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big).$$

The goal of our algorithm is to learn V_ϕ, μ_θ, $\lambda_{\theta,\rho}$, and Λ_ρ, such that the expectation of $J(\xi;\theta,\rho,\phi)$ for any initial state x_0 and action sequence $a_{0:d-1}$ is as close to 0 as possibles. Our sparse PCL algorithm minimizes the empirical objective function $\mathcal{J}_n(\theta,\rho,\phi) = \frac{1}{2}\sum_{\xi_i} J(\xi_i;\theta,\rho,\phi)^2$, which converges to $\mathcal{J}(\theta,\rho,\phi) = \mathbb{E}_{x_0,a_{0:d-1}}\left[\mathbb{E}[J(\xi;\theta,\rho,\phi)^2 \mid x_0, a_{0:d-1}]\right]$ as $i \to \infty$. By the Cauchy-Schwarz inequality, $\mathcal{J}(\theta,\rho,\phi)$ is a conservative surrogate of $\mathbb{E}\left[J(\xi;\theta,\rho,\phi)\right]^2$, which represents the error of the multi-step consistency equation. This relationship justifies that the solution policy of the sparse PCL algorithm is near-optimal (see Corollary 5). Moreover, the gradient of $J(\xi)$ w.r.t. the parameters is as follows:

$$\frac{\partial J(\xi)}{\partial\theta} = J(\xi;\theta,\rho,\phi)\sum_{t=0}^{d-1}\gamma^t\nabla_\theta\big(\lambda_{\theta,\rho}(a_t|x_t) - \alpha\mu_\theta(a_t|x_t)\big),$$

$$\frac{\partial J(\xi)}{\partial\rho} = J(\xi;\theta,\rho,\phi)\sum_{t=0}^{d-1}\gamma^t\nabla_\rho\big(\lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big),$$

$$\frac{\partial J(\xi)}{\partial\phi} = J(\xi;\theta,\rho,\phi)\nabla_\phi\big(V_\phi(x_0) - \gamma^d V_\phi(x_d)\big).$$

We may relate the sparse PCL algorithm to the standard actor-critic (AC) algorithm (Konda & Tsitsiklis, 2000; Sutton et al., 2000), where $\partial J(\xi)/\partial\theta$ and $\partial J(\xi)/\partial\phi$ correspond to the actor and critic updates, respectively. An advantage of sparse PCL over the standard AC is that it does not need the multi-time-scale update for convergence.

While optimizing $J(\theta,\rho,\phi)$ minimizes the mean square of the soft consistency error, in order to satisfy the multi-step consistency in (14), one still needs to impose the following constraints on Lagrange multipliers into the optimization problem: **(i)** $-\frac{\alpha}{2} \leq \Lambda_\rho(x) \leq 0$, and **(ii)** $\lambda_{\theta,\rho}(a|x) \cdot \mu_\theta(a|x) = 0$, $\forall x \in \mathcal{X}$, $\forall a \in \mathcal{A}$. One standard approach is to replace the above constraints with adding penalty functions (Bertsekas, 1999) to the original objective function $\mathcal{J}_n$. Note that each penalty function is associated with a penalty parameter and there are $|\mathcal{X}|\cdot|\mathcal{A}|+2|\mathcal{X}|$ constraints. When $|\mathcal{X}|$ and $|\mathcal{A}|$ are large, tuning all the parameters becomes computationally expensive. Another approach is to update the penalty parameters using gradient ascent methods (Bertsekas, 2014). This is equivalent to finding the saddle point of the Lagrangian function in the constrained optimization problem. However, the challenge is to balance the primal and dual updates in practice.

We hereby describe an alternative and a much simpler methodology to parameterize the Lagrange multipliers $\lambda_{\theta,\rho}(a|x)$ and $\Lambda_\rho(x)$, such that the aforementioned constraints are immediately satisfied. Although this method may impose extra restrictions to the representations of their function approximations, it avoids the difficulties of directly solving a constrained optimization problem. Specifically, to satisfy the constraint **(i)**, one can parameterize Λ_ρ with a multilayer perceptron network that has either an activation function of $-\alpha/2 \cdot \sigma(\cdot)$ or $-\alpha/2 \cdot (1 + \tanh(\cdot))/2$ at its last layer. To satisfy constraint **(ii)**, we consider the case when μ_θ is written in form of $(f_\theta(x, a))^+$ for some function approximator f_θ. This parameterization of μ_θ is justified by the closed-form solution policy of the Tsallis entropy-regularized MDP problem in (7). Specifically, (7) uses $f^*_{\text{sp}}(x, a) = Q^*_{\text{sp}}(x, a)/\alpha - \mathcal{G}(Q^*_{\text{sp}}(x, \cdot)/\alpha)$. Now suppose that $\lambda_{\theta,\rho}$ is parameterized as follows: $\lambda_{\theta,\rho}(a|x) = \big(-f_\theta(x, a)\big)^+ \cdot F_\rho(x, a)$, where $F_\rho : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^+$ is an auxiliary function approximator. Then by the property $(x)^+ \cdot (-x)^+ = 0$, constraint **(ii)** is immediately satisfied. A pseudo-code of our sparse PCL algorithm can be found in Algorithm 1 in the Appendix A.

Unified Sparse PCL Note that the closed-form optimal policy μ^*_{sp} and value function V^*_{sp} are both functions of the optimal state-action value function Q^*_{sp}. As in soft PCL, based on this observation one can also parameterize both policy and value function in sparse PCL (see Eq. 7) with a single function approximator $Q_\psi(x, a)$. Although consistency does not imply optimality in sparse MDPs (as opposed to the case of soft MDPs), the justification of this parameterization comes from Corollary 4, where the unique optimal value function and optimal policy satisfy the consistency equation (14). From an actor-critic perspective, the significance of this is that both policy *(actor)* and value function *(critic)* can be updated simultaneously without affecting the convergence. Accordingly, the update rule for the model parameter ψ takes the form

$$\frac{\partial J(\xi)}{\partial \psi} = J(\xi; \psi, \rho)\Big(\sum_{t=0}^{d-1} \gamma^t \nabla_\psi \big(\lambda_{\psi,\rho}(a_t|x_t) - \alpha\mu_\psi(a_t|x_t)\big) + \nabla_\psi V_\psi(x_0) - \gamma^d \nabla_\psi V_\psi(x_d)\Big).$$

The Saddle Point Reformulation While sparse PCL presents an effective policy learning methodology, unless the immediate reward and transition function are deterministic, in general the gradient derived by minimizing the empirical mean-square consistency error $\mathcal{J}_n(\theta, \rho, \phi)$ is biased w.r.t. the original consistency error $\mathbb{E}\big[J(\xi; \theta, \rho, \phi)\big]^2$. If the variance of $\mathcal{J}_n$ is large, solving for a policy by minimizing this function could lead to an inaccurate solution. To resolve this issue, inspired by Dai et al. (2017); Liu et al. (2015), we propose to reformulate the PCL objective into an equivalent saddle-point problem by exploiting the Fenchel conjugate of the square function, i.e., $x^2 = \max_\nu(2\nu x - \nu^2)$ and by applying the interchangeability principle (Proposition 6.37 in Shapiro et al. 2009). Specifically, using the above properties one can easily show that the problem of minimizing $\mathbb{E}\big[J(\xi; \theta, \rho, \phi)\big]^2$ is equivalent to the saddle-point problem $\min_{\theta,\rho,\phi} \max_{\nu \in \mathcal{N}} \mathbb{E}_\xi[2\nu(x_0, a_{0:d-1}) \cdot J(\xi; \theta, \rho, \phi) - \nu(x_0, a_{0:d-1})^2]$, where $\mathcal{N}$ is the set of functions defined on the space $\mathcal{X} \times \mathcal{A}^d \to \mathbb{R}$. This means that the corresponding stochastic gradient is *unbiased*. Furthermore, using completion of squares and defining $\tilde{\nu}(x_0, a_{0:d-1}) = \nu(x_0, a_{0:d-1}) + V_\phi(x_0)$, the objective function of the saddle-point problem can be rewritten as $\min_{\theta,\rho,\phi} \mathbb{E}_\xi[J(\xi; \theta, \rho, \phi)^2] - \min_{\tilde{\nu}} \mathbb{E}_\xi[(\tilde{\nu}(x_0, a_{0:d-1}) - \tilde{J}(\xi; \theta, \rho, \phi))^2]$, where $\tilde{J}(\xi; \theta, \rho, \phi) = \gamma^d V_\phi(x_d) + \sum_{t=0}^{d-1} \gamma^j \big(r(x_t, a_t) + \frac{\alpha}{2} - \alpha\mu_\theta(a_t|x_t) + \lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big)$ is a d-step auxiliary cumulative return. In this formulation, the first term is the same as the original objective function of sparse PCL, while the second term cancels extra variance coming from $\tilde{J}(\xi; \theta, \rho, \phi)$. Leveraging this property, we can propose an alternative saddle point algorithm for sparse PCL, which we call sparse saddle point PCL (Algorithm 2 in Appendix B). Notice that the above problem is *not* a standard convex-concave saddle point problem, i.e., the objective is concave in ν for fixed (θ, ρ, ϕ), but it is not necessarily always convex in (θ, ρ, ϕ) for fixed ν. Yet, utilizing recent results from saddle point optimization, one can still derive convergence analysis for this algorithm (see Appendix B for more details).

7. Experimental Results

We demonstrate the effectiveness of the sparse PCL algorithm by comparing its performance with that of the soft PCL algorithm on a number of RL environments available in the OpenAI Gym environment (Brockman et al., 2016).

7.1. Discrete Control

Here we compare the performance of these two algorithms on the following standard algorithmic tasks: **1)** Copy, **2)** DuplicatedInput, **3)** RepeatCopy, **4)** Reverse, and **5)** ReversedAddition (see Appendix C for more details). Each task can be viewed as a grid environment, where each cell stores a single character from a finite vocabulary $\mathcal{V}$. An agent moves on the grid of the environment and writes to output. At each time step the agent observes the character of the single cell in which it is located. After observing the character, the agent must take an action of the form (m, w, c), where m determines the agent's move to an adjacent cell, (in 1D environments, $m \in \{\text{left}, \text{right}\}$; in 2D environments, $m \in \{\text{left}, \text{right}, \text{up}, \text{down}\}$), $w \in \{0, 1\}$ determines whether the agent writes to output or not, and $c \in \mathcal{V}$ determines the character that the agent writes if $w = 1$ (otherwise $c = \emptyset$). Based on this problem setting,

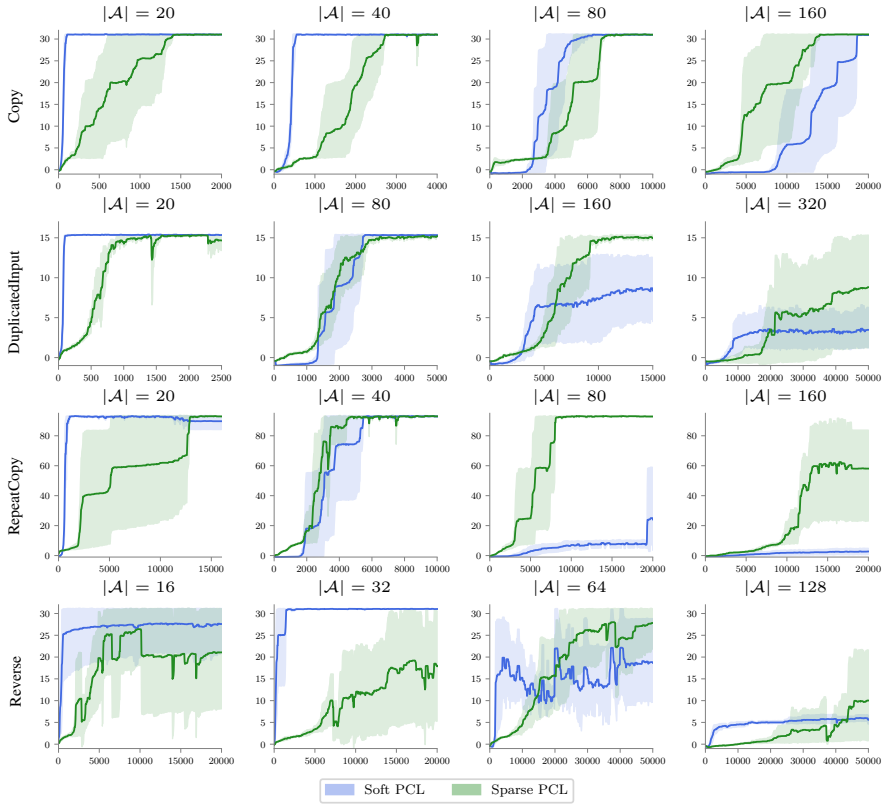

Figure 1. Results of the average reward from sparse PCL and standard soft PCL during training. Each row corresponds to a specific algorithmic task. For each particular task, the action space is increased from left to right across the rows, corresponding to an increase in difficulty. We observe that soft PCL returns a better solution when the action space is small, but its performance degrades quickly as the size of the action space grows. On the other hand, sparse PCL is not only able to learn good policies in tasks with small actions, but unlike soft PCL, also successfully learns high-reward policies in the higher-dimension variants (see Appendix C for additional results).

the action space $\mathcal{A}$ has size $|\mathcal{A}| = \Theta(|\mathcal{V}|)$. Accordingly, the difficulty of these tasks grows with the size of the vocabulary. To illustrate the effectiveness of Tsallis entropy-regularized MDPs in problems with large action space, we evaluate these two PCL algorithms on 4 different choices of $|\mathcal{V}|$.

In each task, the agent has a different goal. In Copy, the environment is a 1D sequence of characters and the agent aims to copy the sequence to output. In DuplicatedInput, the environment is a 1D sequence of duplicated characters and the agent needs to write the de-duplicated sequence to output. In RepeatCopy, the environment is a 1D sequence of characters in which the agent must copy in forward order, reverse order, and finally forward order again. In Reverse, the environment is a 1D sequence of characters in which the agent must copy to output in reverse order. Finally, in ReversedAddition, the environment is a $2n$ grid of digits representing two numbers in base-$|\mathcal{V}|$ that the agent needs to sum. In each task the agent receives a reward of 1 for each correctly output character. The episode is terminated either when the task is completed or when the agent outputs an incorrect character.

We follow a similar experimental procedure as in Nachum et al. (2017), where the functions V, μ, λ, and Λ in the consistency equations are parameterized with a recurrent neural network with multiple heads. For each task and each PCL algorithm, we perform a hyper-parameter search to find the optimal regularization weight α, and the corresponding training curves for average reward are shown in Figure 1. To increase the statistical significance of these experiments, we also train these policies on 5 different Monte Carlo trials (Notice that these environments are inherently deterministic, and thus, no additional Monte Carlo evaluation is needed.). Details of the experimental setup and extra numerical results are included in Appendix C.

For each task we evaluated sparse PCL compared to the original soft PCL on a suite of variants which successively increase the vocabulary size. For low vocabulary sizes soft PCL achieves better results. This suggests that Shannon entropy encourages better exploration in small action spaces. Indeed, in such regimes, a greater proportion of the total ac-

tions are useful to explore and exploration is not as costly. Therefore, the decreased exploration of the Tsallis entropy may outweigh its asymptotic benefits. The sub-optimality bounds presented in this paper support this behavior: when $|\mathcal{A}|$ is small, $\alpha_{\text{soft PCL}} \log(|\mathcal{A}|) \leq \frac{3}{2}\alpha_{\text{sparse PCL}}$.

As we increase the vocabulary size and as a result the action space, the picture changes. We see that the advantage of soft PCL over sparse PCL decreases until eventually the order is reversed and sparse PCL begins to show a significant improvement over the standard soft PCL. This supports our original hypothesis. In large action spaces, the tendency of soft PCL to assign a non-zero probability to many sub-optimal actions over-emphasizes exploration and is detrimental to the final reward performance. On the other hand, sparse PCL is able to handle exploration in large action spaces properly. These empirical results provide evidence for this unique advantage of sparse PCL.

7.2. Continuous Control

We further evaluate the two PCL algorithms on HalfCheetah, a continuous control problem in the OpenAI gym. The environment consists of a $6-$dimensional action space, where each dimension corresponds to a torque of $[-1, 1]$. Here we discretize each continuous action with either one of the following grids: $\{-1, 0, 1\}$ and $\{-1, -0.5, 0, 0.5, 1\}$. Even though the resolution of these discretization grids is coarse, the corresponding action spaces are quite large, with sizes of $3^6 = 729$ and $5^6 = 15625$, respectively.

We present the results of sparse PCL and soft PCL on these discretized problems in Figure 2. Similar to the observations in the algorithmic tasks, here the policy learned by sparse PCL performs much better than that of soft PCL. Specifically sparse PCL achieves higher average reward and is able to learn much faster. To better visualize the learning progress of these two PCL algorithms in these problems, at each training step we also compare the average probability of the most-likely actions across all time-steps from the on-policy trajectory. Specifically in each iteration we collect a single on-policy trajectory of 1000 steps. Therefore, this metric is an average over 1000 samples of (greedy) action probabilities. Clearly, sparse PCL quickly converges to a near-deterministic policy, while the policy generated by soft PCL still allocates significant probability masses to non-optimal actions (as the average probability of most-likely actions barely ever exceeds 0.75). In environments like HalfCheetah, where the trajectory has a long horizon (1000 steps), the soft-max policy will in general suffer because it chooses a large number of sub-optimal actions in each episode for exploration.

Comparing with the performance of other continuous RL algorithms, such as deterministic policy gradient (DPG) (Silver et al., 2014), we found that the policy generated by sparse PCL is sub-optimal. This is mainly due to the coarse discretization of the action space. Our main purpose here is to demonstrate the fast and improved convergence to deterministic policies in sparse PCL, compared to soft PCL. Further evaluation of sparse PCL will be left to future work.

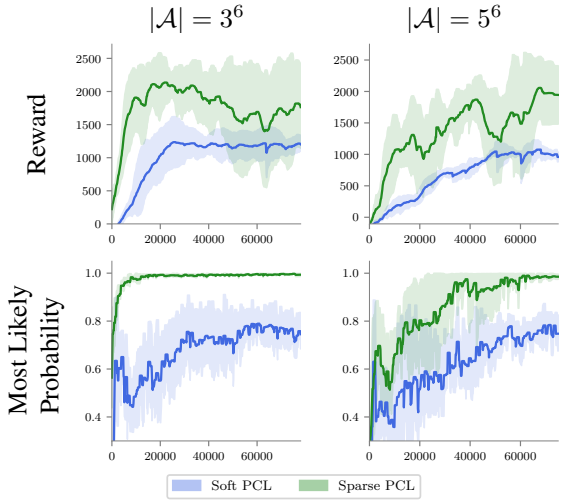

Figure 2. Results of sparse PCL and soft PCL in HalfCheetah with discretized actions. The top figure shows the average reward over 5 random runs during training, with best hyper-parameters. On the bottom we plot the average probability of the most-likely actions during training. The bottom figure illustrates the fast convergence of sparse PCL to a near-deterministic policy.

8. Conclusions

In this work we studied the sparse entropy-regularized problem in RL, whose optimal policy has non-zero probability for only a small number of actions. Similar to the work by Nachum et al. (2017), we derived a relationship between (near-)optimality and consistency for this problem. Furthermore, by leveraging the properties of the consistency equation, we proposed a class of sparse path consistency learning (sparse PCL) algorithms that are applicable to both on-policy and off-policy data and can learn from multi-step trajectories. We found that the theoretical advantages of sparse PCL correspond to empirical advantages as well. For tasks with a large number of actions, we find significant improvement in final performance and amount of time needed to reach that performance by using sparse PCL compared to the original soft PCL.

Future work includes **1)** extending the sparse PCL algorithm to the more general class of Tsallis entropy, **2)** investigating the possibility of combining sparse PCL and path following algorithms such as TRPO (Schulman et al., 2015), and **3)** comparing the performance of sparse PCL with other deterministic policy gradient algorithms, such as DPG (Silver et al., 2014) in the continuous domain.

References

Asadi, K. and Littman, M. An alternative softmax operator for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 243–252, 2017.

Azar, M., Gómez, V., and Kappen, H. Dynamic policy programming. *Journal of Machine Learning Research*, 13:3207–3245, 2012.

Bellman, R. *Dynamic Programming*. Princeton University Press, 1957.

Belousov, B. and Peters, J. F-divergence constrained policy improvement. *arXiv preprint arXiv:1801.00056*, 2017.

Bertsekas, D. *Nonlinear programming*. Athena scientific Belmont, 1999.

Bertsekas, D. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

Bertsekas, D. and Tsitsiklis, J. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym. arXiv:1606.01540, 2016.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Chen, J., and Song, L. Smoothed dual embedding control. *arXiv preprint arXiv:1712.10285*, 2017.

Farahmand, A. M., Ghavamzadeh, M., Szepesvári, Cs., and Mannor, S. Regularized policy iteration. In *Proceedings of Advances in Neural Information Processing Systems 21*, pp. 441–448. MIT Press, 2008.

Farahmand, A. M., Ghavamzadeh, M., Szepesvári, Cs., and Mannor, S. Regularized fitted Q-iteration for planning in continuous-space Markovian decision problems. In *Proceedings of the American Control Conference*, pp. 725–730, 2009.

Fox, R., Pakman, A., and Tishby, N. G-learning: Taming the noise in reinforcement learning via soft update. In *Proceedings of the 32nd International Conference on Uncertainty in Artificial Intelligence*, pp. 202–211, 2016.

Ghadimi, S. and Lan, G. Stochastic first and zeroth order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Ghavamzadeh, M., Lazaric, A., Munos, R., and Hoffman, M. Finite-sample analysis of lasso-TD. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, pp. 1177–1184, 2011.

Johns, J., Painter-Wakefield, C., and Parr, R. Linear complementarity for regularized policy evaluation and improvement. In *Proceedings of Advances in Neural Information Processing Systems 23*, pp. 1009–1017. MIT Press, 2010.

Kappen, H. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics*, 11, 2005.

Kolter, Z. and Ng, A. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the Twenty-Sixth International Conference on Machine Learning*, pp. 521–528, 2009.

Konda, V. and Tsitsiklis, J. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014, 2000.

Lee, K., Choi, S., and Oh, S. Sparse Markov decision processes with causal sparse Tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 2018.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Finite-sample analysis of proximal gradient td algorithms. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 504–513, 2015.

Milgrom, P. and Segal, I. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

Mnih, V., Badia, A., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1928–1937, 2016.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. In *NIPS*, pp. 2772–2782, 2017.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Trust-PCL: An off-policy trust region method for continuous control. In *Proceedings of the 5th International Conference on Learning Representations*, 2018.

Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized Markov decision processes. arXiv:1705.07798, 2017.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., and Mnih, V. PGQ: Combining policy gradient and Q-learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Peters, J., Müling, K., and Altun, Y. Relative entropy policy search. In *Proceedings of the 24th Conference on Artificial Intelligence*, pp. 1607–1612, 2010.

Puterman, M. *Markov Decision Processes*. Wiley Interscience, 1994.

Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1889–1897, 2015.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML*, 2014.

Sutton, R. and Barto, A. *An Introduction to Reinforcement Learning*. MIT Press, 1998.

Sutton, R., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 2000.

Todorov, E. Linearly-solvable Markov decision problems. In *Proceedings of the 19th Advances in Neural Information Processing*, pp. 1369–1376, 2006.

Todorov, E. Policy gradients in linearly-solvable MDPs. In *Proceedings of the 23rd Advances in Neural Information Processing*, pp. 2298–2306, 2010.

Tsallis, C. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52(1):479–487, 1988.

Ziebart, B. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

A. Proofs of Section 5

Consider the Bellman operator for the entropy-regularized MDP with Tsallis entropy:

$$(\mathcal{T}_{\text{sp}} f)(x) = \alpha \cdot \text{spmax}\Big(\big[r(x,\cdot) + \gamma \sum_{x'} P(x'|x,\cdot) f(x') \big] / \alpha \Big).$$

We first have the following technical result about its properties.

Proposition 1. *The sparse-max Bellman operator $\mathcal{T}_{sp}$ has the following properties: (i) Translation: $(\mathcal{T}_{sp}(V+\beta))(x) = (\mathcal{T}_{sp}V)(x) + \gamma\beta$; (ii) γ-contraction: $\|(\mathcal{T}_{sp}V_1) - (\mathcal{T}_{sp}V_2)\|_\infty \le \gamma \|V_1 - V_2\|_\infty$; (iii) Monotonicity: $(\mathcal{T}_{sp}V_1)(x) \le (\mathcal{T}_{sp}V_2)(x)$ for any value functions $V_1, V_2 : \mathcal{X} \to \mathbb{R}$ such that $V_1(x) \le V_2(x)$, $\forall x \in \mathcal{X}$.*

The detailed proof of this proposition can be found in Lee et al. (2018). Using these results, the Banach fixed point theorem shows that there exists a unique solution for the following fixed point equation: $V(x) = (\mathcal{T}_{\text{sp}}V)(x)$, $\forall x \in \mathcal{X}$, and this solution is equal to the optimal value function $V^*_{\text{sp}}(x)$. Analogous to the arguments in standard MDPs, in this case the optimal value function can also be computed using dynamic programming methods such as value iteration.

Before proving the main results, notice that by using analogous arguments of the complementary-slackness property in KKT conditions, the second and the third consistency equation in (13) is equivalent to the following condition:

$$\begin{aligned} r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) V(x') + \frac{\alpha}{2} - \alpha\mu(a|x) - V(x) &= \Lambda(x), \ \forall x \in \mathcal{X}, \ \forall a \in \mathcal{A}_\mu(x), \\ r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) V(x') + \frac{\alpha}{2} - \alpha\mu(a|x) - V(x) &\le \Lambda(x), \ \forall x \in \mathcal{X}, \ \forall a \notin \mathcal{A}_\mu(x), \end{aligned} \tag{15}$$

where $\mathcal{A}_\mu(x) = \{a \in \mathcal{A} : \mu(a|x) > 0\}$ represents the set of actions that have non-zero probabilities w.r.t policy μ.

Theorem 3. *The pair of optimal value function and optimal policy (V^*_{sp}, μ^*_{sp}) of the MDP problem in (3) satisfies the consistency equation in (13).*

Proof. Recall that the optimal state-action value function is given by

$$Q^*_{\text{sp}}(x,a) = r(x,a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x,a) V^*_{\text{sp}}(x').$$

According to Bellman's optimality, the optimal value function satisfies the following equality:

$$V^*_{\text{sp}}(x) = \max_{\mu \in \Delta_x} \sum_{a \in \mathcal{A}} \mu(a|x) \left[Q^*_{\text{sp}}(x,a) + \frac{\alpha}{2}(1 - \mu(a|x)) \right], \tag{16}$$

at any state $x \in \mathcal{X}$, where μ^*_{sp} is the corresponding maximizer. By the KKT condition, we have that

$$Q^*_{\text{sp}}(x,a) + \frac{\alpha}{2}(1 - \mu^*_{\text{sp}}(a|x)) + \lambda^*_{\text{sp}}(a|x) = \Lambda^*_{\text{sp}}(x) + \frac{\alpha}{2}\mu^*_{\text{sp}}(a|x),$$

for any $x \in \mathcal{X}$ and any $a \in \mathcal{A}$, where Λ^*_{sp} is the Lagrange multiplier that corresponds to equality constraint $\sum_{a \in \mathcal{A}} \mu(a|x) = 1$, and $\lambda^*_{\text{sp}} \ge 0$ is the Lagrange multiplier that corresponds to inequality constraint $\mu(a|x) \ge 0$ such that

$$\lambda^*_{\text{sp}}(a|x) \cdot \mu^*_{\text{sp}}(a|x) = 0, \ \forall x \in \mathcal{X}, \ \forall a \in \mathcal{A}.$$

Recall from the definition of optimal state-action value function Q^*_{sp} and the definition of the optimal policy μ^*_{sp}, one has that $\mathcal{A}_{\mu^*_{\text{sp}}}(x) = \mathcal{S}(Q^*_{\text{sp}}(x,\cdot))$. This condition further implies

$$\Lambda^*_{\text{sp}}(x) = Q^*_{\text{sp}}(x,a) + \frac{\alpha}{2}(1 - 2\mu^*_{\text{sp}}(a|x)), \ \forall x \in \mathcal{X}, a \in \mathcal{S}(Q^*_{\text{sp}}(x,\cdot)).$$

Substituting the equality in (16) to this KKT condition, and noticing that $0 \le \sum_{a \in \mathcal{A}} \mu^*_{\text{sp}}(a|x))^2 \le 1$, the KKT condition implies that

$$\Lambda^*_{\text{sp}}(x) + \frac{\alpha}{2} \ge V^*_{\text{sp}}(x) = \Lambda^*_{\text{sp}}(x) + \frac{\alpha}{2} \sum_{a \in \mathcal{A}} \mu^*_{\text{sp}}(a|x) \mu^*_{\text{sp}}(a|x) \ge \Lambda^*_{\text{sp}}(x),$$

which further implies that

$$-\frac{\alpha}{2} \leq \Lambda^*_{\text{sp}}(x) - V^*_{\text{sp}}(x) \leq 0, \ \forall x \in \mathcal{X}.$$

Therefore, by defining $\Lambda(x) = \Lambda^*_{\text{sp}}(x) - V^*_{\text{sp}}(x)$, and $\lambda(a|x) = \lambda^*_{\text{sp}}(a|x)$, one immediately has that $-\frac{\alpha}{2} \leq \Lambda(x) \leq 0$, $\forall x \in \mathcal{X}$. Using this construction, one further has the following expression for any $x \in \mathcal{X}$ and any $a \in \mathcal{S}(Q^*_{\text{sp}}(x, \cdot))$:

$$\Lambda(x) = Q^*_{\text{sp}}(x, a) + \frac{\alpha}{2} - \alpha\mu^*_{\text{sp}}(a|x) - V^*_{\text{sp}}(x),$$

which proves consistency, based on the equivalence condition in (15). □

Theorem 4. *The solution policy μ of the consistency equation in (13) is $\alpha/(1-\gamma)$-optimal w.r.t. the sparse MDP problem in (3). That is,*

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left[R_t + \frac{\alpha}{2}\left(1 - \mu(a_t|x_t)\right)\right] \mid x_0 = x, \mu, P\right] \geq V^*_{sp}(x) - \frac{\alpha}{(1-\gamma)}. \tag{17}$$

Proof. To proof the sub-optimality performance bound given in this theorem, we first study the expression of $(\mathcal{T}_{\text{sp}}V)$, where $\mathcal{T}_{\text{sp}}$ is the Bellman operator of the Tsallis entropy-regularized MDP problem in (3). Let

$$\bar{Q}(x, a) = r(x, a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, a)V(x')$$

be the corresponding state-action value function. Using the definition from (3), one has the following expression:

$$\begin{aligned}(\mathcal{T}_{\text{sp}}V)(x) &= \alpha \cdot \text{spmax}\left(\frac{1}{\alpha} \cdot \left\{r(x, a) + \gamma \sum_{x' \in \mathcal{X}} P(x'|x, a)V(x')\right\}_{a \in \mathcal{A}}\right) \\ &= \alpha \cdot \text{spmax}\left(\frac{\bar{Q}(x, \cdot)}{\alpha}\right).\end{aligned}$$

Furthermore, by exploiting the structure of the sparse-max formulation of an arbitrary value function, one also has the following chain of equalities/inequalities:

$$\begin{aligned}\alpha \cdot \text{spmax}\left(\frac{\bar{Q}(x, \cdot)}{\alpha}\right) &= \max_{\mu \in \Delta_x} \sum_{a \in \mathcal{A}} \mu(a|x) \cdot \left(\bar{Q}(x, a) + \frac{\alpha}{2}(1 - \mu(a|x))\right) \\ &= \sum_{a \in \mathcal{A}} \mu(a|x) \cdot \left(\bar{Q}(x, a) + \frac{\alpha}{2}(1 - \mu(a|x))\right) \\ &= \sum_{a \in \mathcal{A}} \mu(a|x) \left(\bar{Q}(x) + \frac{\alpha}{2} - \alpha\mu(a|x)\right) + \frac{\alpha}{2} \sum_{a \in \mathcal{A}} \mu(a|x)^2 \\ &\leq V(x) + \frac{\alpha}{2}.\end{aligned}$$

The first equality follows from the fact that $\alpha \cdot \text{spmax}(\bar{Q}(x, \cdot)/\alpha)$ is a closed form solution of the optimization problem

$$\max_{\mu \in \Delta_x} \sum_{a \in \mathcal{A}} \mu(a|x) \left(\bar{Q}(x, a) - \alpha\mathbb{H}_\mu(x, a)\right),$$

when $\mathbb{H}_\mu$ is the Tsallis entropy. The second equality follows from the fact that if (V, μ) satisfies the consistency equation, then there exists a Lagrange multiplier $\Lambda^*(x) = \Lambda(x) + V(x)$, $\forall x \in \mathcal{X}$ such that the following KKT conditions hold:

$$\begin{aligned}&\bar{Q}(x, a) + \frac{\alpha}{2} - \alpha\mu(a|x) = \Lambda^*(x), \ \forall x \in \mathcal{X}, \ \forall a \in \mathcal{A}_\mu(x), \\ &\bar{Q}(x, a) + \frac{\alpha}{2} - \alpha\mu(a|x) \leq \Lambda^*(x), \ \forall x \in \mathcal{X}, \ \forall a \notin \mathcal{A}_\mu(x), \\ &\sum_a \mu(a|x) = 1, \ \mu(a|x) \geq 0, \ \forall x \in \mathcal{X}, \ \forall a \in \mathcal{A},\end{aligned}$$

which further implies that μ is the maximizer of the inner optimization problem. The third equality follows from arithmetic manipulations, and the first inequality follows from the consistency equation in (15), i.e., for any $x \in \mathcal{X}$ and any $a \in \mathcal{A}_\mu(x)$, there exists $\Lambda(x) \in [-\frac{\alpha}{2}, 0]$ such that:

$$0 \geq \Lambda(x) = \bar{Q}(x,a) + \frac{\alpha}{2} - \alpha\mu(a|x) - V(x) \iff \bar{Q}(x,a) + \frac{\alpha}{2} - \alpha\mu(a|x) \leq V(x).$$

Therefore combining all these arguments, one concludes that the following Bellman inequality holds:

$$(\mathcal{T}_{\text{sp}} V)(x) \leq V(x) + \frac{\alpha}{2}, \ \forall x \in \mathcal{X}. \tag{18}$$

Now recall that the $\gamma-$contraction property (w.r.t. the $\infty-$norm) of the Bellman operator $\mathcal{T}_{\text{sp}}$. By the Banach fixed-point theorem, this property implies that there exists a unique fixed point solution V^*_{sp} to equation $V(x) = (\mathcal{T}_{\text{sp}} V)(x)$, for all $x \in \mathcal{X}$, and it is the limit point (over all $x \in \mathcal{X}$) of the converging iterative sequence $\lim_{n\to\infty}(\mathcal{T}^n_{\text{sp}} V_0)(x)$ for any initial value function V_0. Also recall that the translation property of this Bellman operator, i.e., for any constant K, $(\mathcal{T}_{\text{sp}}(V + K)) = (\mathcal{T}_{\text{sp}} V) + \gamma K$. Therefore, by repeatedly applying the Bellman operator to both sides of the inequality in (18), and by using the above properties of a Bellman operator, one can show that

$$V^*_{\text{sp}}(x) = \lim_{n\to\infty}(\mathcal{T}^n_{\text{sp}} V)(x) \leq \sum_{t=0}^{\infty} \gamma^t \cdot \frac{\alpha}{2} + V(x) = \frac{\alpha}{2} \cdot \frac{1}{1-\gamma} + V(x), \ \forall x \in \mathcal{X}. \tag{19}$$

Furthermore, consider the consistency equation in (13), i.e., there exists a function $\Lambda(x) \in [0, \frac{\alpha}{2}]$ such that for any $x \in \mathcal{X}$ and any $a \in \mathcal{A}_\mu(x)$,

$$-\frac{\alpha}{2} \leq \Lambda(x) = \bar{Q}(x,a) + \frac{\alpha}{2} - \alpha\mu(a|x) - V(x) \iff V(x) \leq \bar{Q}(x,a) + \alpha - \alpha\mu(a|x).$$

By multiplying $\mu(a|x)$ on both sides of this inequality and summing over $a \in \mathcal{A}$, the above expression implies

$$\begin{aligned} V(x) &\leq \sum_{a\in\mathcal{A}} \mu(a|x) \left(\bar{Q}(x,a) + \alpha - \alpha\mu(a|x)\right) \\ &\leq \sum_{a\in\mathcal{A}} \mu(a|x) \left(\bar{Q}(x,a) + \frac{\alpha}{2}\left(1 - \mu(a|x)\right)\right) + \frac{\alpha}{2}\sum_{a\in\mathcal{A}} \mu(a|x)\left(1 - \mu(a|x)\right) \\ &\leq \sum_{a\in\mathcal{A}} \mu(a|x) \left(r(x,a) + \gamma \sum_{x'\in\mathcal{X}} P(x'|x,a)V(x') + \frac{\alpha}{2}\left(1 - \mu(a|x)\right)\right) + \frac{\alpha}{2}. \end{aligned} \tag{20}$$

Therefore, equipped with the $\gamma-$contraction property of the following Bellmen operator:

$$(\mathcal{T}_\mu V)(x) = \sum_{a\in\mathcal{A}} \mu(a|x) \left(r(x,a) + \frac{\alpha}{2}\left(1 - \mu(a|x)\right) + \gamma \sum_{x'\in\mathcal{X}} P(x'|x,a)V(x')\right)$$

and the Banach fixed-point theorem, for any initial value function V_0, one can deduce the following expression:

$$\lim_{n\to\infty} \mathbf{T}_\mu[V_0]^n(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left[r(x_t, a_t) + \frac{\alpha}{2}\left(1 - \mu(a_t|x_t)\right)\right] \mid \mu, x_0 = x\right].$$

Using the translation property of the Bellman operator $(\mathcal{T}_\mu V)$ and repeatedly applying this Bellman operator to both sides of (20), one obtains the following inequality for any $x \in \mathcal{X}$:

$$\begin{aligned} V(x) &\leq \lim_{n\to\infty} (\mathcal{T}_\mu V)^n(x) + \sum_{t=0}^{\infty} \frac{\alpha}{2}\gamma^t \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left[r(x_t, a_t) + \frac{\alpha}{2}\left(1 - \mu(a_t|x_t)\right)\right] \mid \mu, x_0 = x\right] + \frac{\alpha}{2} \cdot \frac{1}{1-\gamma}. \end{aligned} \tag{21}$$

Therefore, by combining the results in (19) and in (21), one completes the proof of this theorem. □

Corollary 4. *The optimal policy μ^*_{sp} and value function V^*_{sp} of the sparse MDP (3) satisfy the multi-step consistency equation (14).*

Proof. The proof follows from Theorem 1, by repeatedly applying the expression in (13) over trajectory ξ, taking the expectation over the trajectory, and using telescopic cancellation of the value function of intermediate states. □

Corollary 5. *Any policy μ that satisfies the consistency equation (14) is $\alpha/(1-\gamma)$-optimal in sparse MDP (3).*

Proof. Consider the multi-step consistency equation (14). Since it is true for any initial state x_0 and sequence of actions $a_{0:d-1}$, unrolling it for another d steps starting at state x_d and using the action sequence $a_{d:2d-1}$ yields $V(x_0) = \mathbb{E}_{x_{1:2d}|x_0,a_{0:2d-1}}[\gamma^{2d}V(x_{2d}) + \sum_{t=0}^{2d-1}\gamma^t(r(x_t,a_t) + \frac{\alpha}{2} - \alpha\mu(a_t|x_t) + \lambda(a_t|x_t) - \Lambda(x_t))]$. Note that this process can be repeated for an arbitrary number of times (say k times), and also note that as V is a bounded function, one has $\lim_{k\to\infty}\gamma^{kd}V(x_{kd}) = 0$. Therefore, by further unrolling, we obtain $V(x_0) = \mathbb{E}_{x_{1:\infty}|x_0,a_{0:\infty}}[\sum_{t=0}^{\infty}\gamma^t(r(x_t,a_t) + \frac{\alpha}{2} - \alpha\mu(a_t|x_t) + \lambda(a_t|x_t) - \Lambda(x_t))]$.

Followed from the Banach fixed-point theorem (Bertsekas & Tsitsiklis, 1996), one can show that the solution pair (V,μ) is also a solution to the one-step consistency condition in (13), i.e., $V(x) = r(x,a) + \frac{\alpha}{2} - \alpha\mu(a|x) + \lambda(a|x) - \Lambda(x) + \gamma\sum_{x'\in\mathcal{X}}P(x'|x,a)V(x')$, for any $x \in \mathcal{X}$ and $a \in \mathcal{A}$. Thus the $\alpha/(1-\gamma)$-optimality performance guarantee of μ is implied by Theorem 2. □

Algorithm 1 Sparse Path Consistency Learning

Input: Environment ENV, learning rate η, discount factor γ, regularization α, rollout d, number of steps N, replay buffer capacity B, prioritized replay hyper-parameter α. Parameterizations of Λ and λ follow from the descriptions in Section 6.

function Gradients($x_{0:T}, a_{0:T-1}, R_{0:T-1}$)

Compute $J(t) = -\bar{V}_\phi(x_t) + \gamma^d\bar{V}_\phi(x_{t+d}) + \sum_{j=0}^{d-1}\gamma^j(R_{t+j} + \alpha/2 - \alpha\bar{\mu}_\theta(a_{t+j}|x_{t+j}) + \lambda_\theta(a_{t+j}|x_{t+j}) - \Lambda_\rho(x_{t+j}))$ for $t < T$, padding with zeros as necessary.

Compute $\Delta\theta = \sum_{t=0}^{T-1}J(t)\nabla_\theta J(t)$.

Compute $\Delta\phi = \sum_{t=0}^{T-1}J(t)\nabla_\phi J(t)$.

Compute $\Delta\rho = \sum_{t=0}^{T-1}J(t)\nabla_\rho J(t)$.

Return $\Delta\theta, \Delta\phi, \Delta\rho$

end function

Initialize θ, ϕ, ρ.

Initialize empty replay buffer $RB(\alpha)$.

for $i = 0$ **to** $N-1$ **do**

Sample $x_{0:T}, a_{0:T-1} \sim \bar{\mu}_\theta$ on ENV, yielding reward $R_{0:T-1}$.

$\Delta\theta, \Delta\phi, \Delta\rho$ = Gradients($x_{0:T}, a_{0:T-1}, R_{0:T-1}$).

Update $\theta \leftarrow \theta - \eta\Delta\theta$.

Update $\phi \leftarrow \phi - \eta\Delta\phi$.

Update $\rho \leftarrow \rho - \eta\Delta\rho$.

Input $x_{0:T}, a_{0:T-1}$ into RB with priority $\sum_{j=0}^{T-d}R_j$.

If $|RB| > B$, remove episodes uniformly at random.

Sample $(x_{0:T}, a_{0:T-1}, R_{0:T-1})$ from RB.

$\Delta\theta, \Delta\phi, \Delta\rho$ = Gradients($x_{0:T}, a_{0:T-1}, R_{0:T-1}$).

Update $\theta \leftarrow \theta - \eta\Delta\theta$.

Update $\phi \leftarrow \phi - \eta\Delta\phi$.

Update $\rho \leftarrow \rho - \eta\Delta\rho$.

end for

B. Sparse Saddle Point PCL Algorithm

Algorithm 2 Sparse Saddle Point Path Consistency Learning

Input: Environment ENV, learning rate η, discount factor γ, regularization α, rollout d, number of steps N, replay buffer capacity B, prioritized replay hyper-parameter α. Parameterizations of Λ and λ follow from the descriptions in Section 6.

function Gradients($x_{0:T}, a_{0:T-1}, R_{0:T-1}$)

Compute $J(t) = -\bar{V}_\phi(x_t) + \gamma^d \bar{V}_\phi(x_{t+d}) + \sum_{j=0}^{d-1} \gamma^j (R_{t+j} + \alpha/2 - \alpha\bar{\mu}_\theta(a_{t+j}|x_{t+j}) + \lambda_\theta(a_{t+j}|x_{t+j}) - \Lambda_\rho(x_{t+j}))$ and $\tilde{J}(t) = \gamma^d \bar{V}_\phi(x_{t+d}) + \sum_{j=0}^{d-1} \gamma^j (R_{t+j} + \alpha/2 - \alpha\bar{\mu}_\theta(a_{t+j}|x_{t+j}) + \lambda_\theta(a_{t+j}|x_{t+j}) - \Lambda_\rho(x_{t+j}))$ for $t < T$, padding with zeros as necessary.

Compute $\psi^* \in \arg\min_\psi \sum_{t=0}^{T-1} (\nu_\psi(x_t, a_t) - \tilde{J}(t))^2$

Compute $\Delta\theta = \sum_{t=0}^{T-1} J(t)\nabla_\theta J(t) - (\tilde{J}(t) - \tilde{\nu}_{\psi^*}(x_t, a_t))\nabla_\theta \tilde{J}(t)$.

Compute $\Delta\phi = \sum_{t=0}^{T-1} J(t)\nabla_\phi J(t) - (\tilde{J}(t) - \tilde{\nu}_{\psi^*}(x_t, a_t))\nabla_\phi \tilde{J}(t)$.

Compute $\Delta\rho = \sum_{t=0}^{T-1} J(t)\nabla_\rho J(t) - (\tilde{J}(t) - \tilde{\nu}_{\psi^*}(x_t, a_t))\nabla_\rho \tilde{J}(t)$.

Return $\Delta\theta, \Delta\phi, \Delta\rho$

end function

Initialize θ, ϕ, ρ.

Initialize empty replay buffer $RB(\alpha)$.

for $i = 0$ **to** $N - 1$ **do**

Sample $x_{0:T}, a_{0:T-1} \sim \bar{\mu}_\theta$ on ENV, yielding reward $R_{0:T-1}$.

$\Delta\theta, \Delta\phi, \Delta\rho =$ Gradients$(x_{0:T}, a_{0:T-1}, R_{0:T-1})$.

Update $\theta \leftarrow \theta - \eta\Delta\theta$.

Update $\phi \leftarrow \phi - \eta\Delta\phi$.

Update $\rho \leftarrow \rho - \eta\Delta\rho$.

Input $x_{0:T}, a_{0:T-1}$ into RB with priority $\sum_{j=0}^{T-d} R_j$.

If $|RB| > B$, remove episodes uniformly at random.

Sample $(x_{0:T}, a_{0:T-1}, R_{0:T-1})$ from RB.

$\Delta\theta, \Delta\phi, \Delta\rho =$ Gradients$(x_{0:T}, a_{0:T-1}, R_{0:T-1})$.

Update $\theta \leftarrow \theta - \eta\Delta\theta$.

Update $\phi \leftarrow \phi - \eta\Delta\phi$.

Update $\rho \leftarrow \rho - \eta\Delta\rho$.

end for

Unbiased Gradient Next, we analyze the stochastic gradient used in Algorithm 2 and show that it is unbiased w.r.t. original consistency error if the dual function ν is optimal. Using the Fenchel conjugate function of x^2, first recall that the gradient of the consistency error, which is denoted by $\nabla\mathbb{E}\left[J(\xi;\theta,\rho,\phi)\right]^2$, can be re-written as:

$$\begin{aligned}\nabla\mathbb{E}_\xi\left[J(\xi;\theta,\rho,\phi)\right]^2 =& \nabla \max_{\nu\in F(\mathcal{X}\times\mathcal{A}^d)} \mathbb{E}_\xi\left[2\nu(x_0,a_{0:d-1})\cdot J(\xi;\theta,\rho,\phi) - (\nu(x_0,a_0))^2\right] \\ =& \nabla \max_{\tilde{\nu}\in F(\mathcal{X}\times\mathcal{A}^d)} \mathbb{E}_\xi\left[J(\xi;\theta,\rho,\phi)^2 - (\tilde{\nu}(x_0,a_{0:d-1}) - \tilde{J}(\xi;\theta,\rho,\phi))^2\right].\end{aligned}$$

Since the above objective function is concave in $\tilde{\nu}$, by the Envelope theorem for arbitrary choice sets (Milgrom & Segal, 2002), one can show that the gradient formulation is

$$\nabla\mathbb{E}_\xi\left[J(\xi;\theta,\rho,\phi)\right]^2 = \nabla\mathbb{E}_\xi\left[J(\xi;\theta,\rho,\phi)^2 - (\tilde{\nu}^*(x_0,a_{0:d-1}) - \tilde{J}(\xi;\theta,\rho,\phi))^2\right],$$

where $\tilde{\nu}^*$ is the optimal dual function w.r.t the minimization problem $\mathbb{E}_\xi\left[(\tilde{\nu}(x_0,a_{0:d-1}) - \tilde{J}(\xi;\theta,\rho,\phi))^2\right]$. Therefore, one can easily construct the Monte-Carlo estimate $\frac{1}{N}\sum_{i=1}^N \nabla\left(J(\xi_i;\theta,\rho,\phi)^2 - (\tilde{\nu}^*(x_0,a_{0:d-1}) - \tilde{J}(\xi_i;\theta,\rho,\phi))^2\right)$ of the original gradient w.r.t. minimizing the consistency error, which is also an unbiased estimator.

On the other hand, the dual solution is updated through solving the subproblem

$$\min_{\tilde{\nu}} \mathbb{E}_\xi\left[\left(\gamma^d\bar{V}_\phi(x_{t+d}) + \sum_{j=0}^{d-1}\gamma^j(R_{t+j} + \alpha/2 - \alpha\bar{\mu}_\theta(a_{t+j}|x_{t+j}) + \lambda_\theta(a_{t+j}|x_{t+j}) - \Lambda_\rho(x_{t+j})) - \tilde{\nu}^*(x_0,a_{0:d-1})\right)^2\right].$$

Furthermore, its optimal solution is equal to

$$\tilde{\nu}^*(x_0, a_{0:d-1}) = \mathbb{E}_\xi \left[\gamma^d \bar{V}_\phi(x_{t+d}) + \sum_{j=0}^{d-1} \gamma^j (R_{t+j} + \alpha/2 - \alpha \bar{\mu}_\theta(a_{t+j}|x_{t+j}) + \lambda_\theta(a_{t+j}|x_{t+j}) - \Lambda_\rho(x_{t+j})) \mid x_0, a_{0:d-1}, \bar{\mu}_\theta \right],$$

i.e., the dual variables can be essentially viewed as a multi-step $Q-$function. Therefore, Algorithm 2 could be interpreted as first fitting a parametrized $Q-$function by dual variables $\tilde{\nu}_{\psi^*}(x, a)$ via mean square loss, and then, applying the stochastic descent w.r.t. θ, ϕ, ρ with gradient estimator $\Delta\theta, \Delta\phi, \Delta\rho$.

Variance Reduction In order to show that the variance of the saddle point objective function is less than that of the original sparse PCL algorithm, we study the second term of the consistency error in the saddle point formulation, i.e., $-\min_{\tilde{\nu}} \mathbb{E}_\xi[(\tilde{\nu}(x_0, a_{0:d-1}) - \tilde{J}(\xi; \theta, \rho, \phi))^2]$, where $\tilde{J}(\xi; \theta, \rho, \phi) = \gamma^d V_\phi(x_d) + \sum_{t=0}^{d-1} \gamma^j \big(r(x_t, a_t) + \frac{\alpha}{2} - \alpha\mu_\theta(a_t|x_t) + \lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big)$. Specifically by expanding this expression, one has that

$$\begin{aligned}
&\min_{\tilde{\nu}} \mathbb{E}_\xi[(\tilde{\nu}(x_0, a_{0:d-1}) - \tilde{J}(\xi; \theta, \rho, \phi))^2] \\
=&\min_{\tilde{\nu}} \mathbb{E}_\xi \left[(\tilde{\nu}(x_0, a_{0:d-1}) - \mathbb{E}_\xi[\tilde{J}(\xi; \theta, \rho, \phi) \mid x_0, a_0] \right] + \mathbb{E}_\xi[\tilde{J}(\xi; \theta, \rho, \phi) \mid x_0, a_0] - \tilde{J}(\xi; \theta, \rho, \phi))^2] \\
=&\min_{\tilde{\nu}} \mathbb{E}_\xi \left[\Big(\tilde{\nu}(x_0, a_{0:d-1}) - \mathbb{E}_\xi[\tilde{J}(\xi; \theta, \rho, \phi) \mid x_0, a_0]\Big)^2 \right] + \mathbb{E}_\xi \left[\Big(\mathbb{E}_\xi[\tilde{J}(\xi; \theta, \rho, \phi) \mid x_0, a_0] - \tilde{J}(\xi; \theta, \rho, \phi)\Big)^2 \right] \\
=&\min_{\tilde{\nu}} \mathbb{E}_\xi \left[\Big(\tilde{\nu}(x_0, a_{0:d-1}) - \mathbb{E}_\xi[\tilde{J}(\xi; \theta, \rho, \phi) \mid x_0, a_0]\Big)^2 \right] \\
&+ \mathbb{V}_\xi \left[\gamma^d V_\phi(x_d) + \sum_{t=1}^{d-1} \gamma^j \big(r(x_t, a_t) + \frac{\alpha}{2} - \alpha\mu_\theta(a_t|x_t) + \lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big) \right] \\
=&\mathbb{V}_\xi \left[\gamma^d V_\phi(x_d) + \sum_{t=1}^{d-1} \gamma^j \big(r(x_t, a_t) + \frac{\alpha}{2} - \alpha\mu_\theta(a_t|x_t) + \lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big) \right],
\end{aligned}$$

where the last equality is due to the fact that $\tilde{\nu}^*(x_0, a_{0:d-1}) = \mathbb{E}_\xi[\tilde{J}(\xi; \theta, \rho, \phi) \mid x_0, a_0]$. This shows that the second term of the objective function is essentially the negative of the variance of $\tilde{J}(\xi; \theta, \rho, \phi) - \big(r(x_0, a_0) + \frac{\alpha}{2} - \alpha\mu_\theta(a_0|x_0) + \lambda_{\theta,\rho}(a_0|x_0) - \Lambda_\rho(x_0)\big)$. On the other hand, the objective function of the original sparse PCL algorithm introduces an extra variance term into the consistency error, i.e.,

$$\mathbb{E}_\xi \left[J(\xi; \theta, \rho, \phi)^2 \right] = \mathbb{E}_\xi \left[J(\xi; \theta, \rho, \phi) \right]^2 + \mathbb{V}_\xi \left[\gamma^d V_\phi(x_d) + \sum_{t=1}^{d-1} \gamma^j \big(r(x_t, a_t) + \frac{\alpha}{2} - \alpha\mu_\theta(a_t|x_t) + \lambda_{\theta,\rho}(a_t|x_t) - \Lambda_\rho(x_t)\big) \right].$$

Therefore, if the variance is large, minimizing the objective function as in the original sparse PCL algorithm could lead to highly inaccurate solution, while such an issue does not exist in our saddle point optimization.

Convergence Analysis It is well-known that for convex-concave saddle point problems, applying stochastic descent ensures global convergence in sub-linearly. However this is not true in our problem, where the convex-concavity assumption fails. On the other hand, if one can *exactly* solve the the dual maximization problem of sparse saddle point PCL at each iteration (which is a concave maximization problem), this algorithm can be viewed as a special case of stochastic descent that is applied to solve the non-convex minimization problem $\min_{\theta,\rho,\phi} \mathbb{E}_\xi[J(\xi; \theta, \rho, \phi)^2] - \mathbb{E}_\xi[(\tilde{\nu}^*(x_0, a_{0:d-1}) - \tilde{J}(\xi; \theta, \rho, \phi))^2]$. Then this problem is proven to converge sub-linearly to the stationary point when the step-size is decaying. Such a convergence property in sparse saddle point PCL is a direct result of Corollary 2.2 in Ghadimi & Lan (2013).

C. Experimental Details

For the algorithmic tasks, we follow a similar experimental setup as described in Nachum et al. (2017). We parameterize all values by a single LSTM recurrent neural network with internal dimension 128 and multiple heads (one for each desired quantity). At each training step, we sample a batch of 400 episodes using the current policy acting on the environment.

We perform a gradient step based on this batch. We then add the experience to the replay buffer and perform a gradient step based on an off-policy batch sampled from the replay buffer. We fix the rollout to $d = 10$. As in Nachum et al. (2017), our replay buffer is prioritized by episode rewards: the probability of sampling an episode from the replay buffer is $0.1 + 0.9 \cdot \exp\{\alpha R\}/Z$ where R is the total reward of the episode, Z is a normalizing factor, and we use $\alpha = 0.5$. We use a replay buffer of capacity $B = 10,000$ episodes. In our experiments we use a learning rate of $\eta = 0.005$ and discount $\gamma = 0.9$.

For HalfCheetah we parameterized the policy and value networks as feed forward networks with two hidden layers of dimension 64 and $\tanh$ non-linearities. At each training step we sampled 100 steps from the environment and input these into a replay buffer. We then sample a batch of 25 sub-episodes of 100 steps from the replay buffer, prioritized by exponentiated recency (with weight 0.01) and perform a single training step. We use rollout $d = 10$, discount $\gamma = 0.99$, and performed a hyperparameter search over learning rate $\eta \in \{0.0005, 0.0001\}$.

In standard Soft PCL, the policy $\bar{\mu}_\theta$ is determined by logits output by the neural network. That is,

$$\bar{\mu}_\theta(-|x) = \text{softmax}\left(\text{NN}(x,\theta)_{0:|\mathcal{A}|-1}\right), \tag{22}$$

where $\text{NN}(x,\theta)_{0:|\mathcal{A}|-1}$ are $|\mathcal{A}|$ output values of the neural network. For Sparse PCL, to induce sparsity, we parameterize the policy using the $\mathcal{G}$ function:

$$\bar{\mu}_\theta(-|x) = \text{relu}\left(\text{NN}(x,\theta)_{0:|\mathcal{A}|-1} - \mathcal{G}(\text{NN}(x,\theta)_{0:|\mathcal{A}|-1})\right). \tag{23}$$

Accordingly, λ_θ is parameterized as

$$\lambda_\theta(-|x) = \text{relu}\left(\mathcal{G}(\text{NN}(x,\theta)_{0:|\mathcal{A}|-1} - \text{NN}(x,\theta)_{0:|\mathcal{A}|-1})\right) \exp\{\text{NN}(x,\theta)_{|\mathcal{A}|}\}. \tag{24}$$

C.1. Experimental Results for ReversedAddition

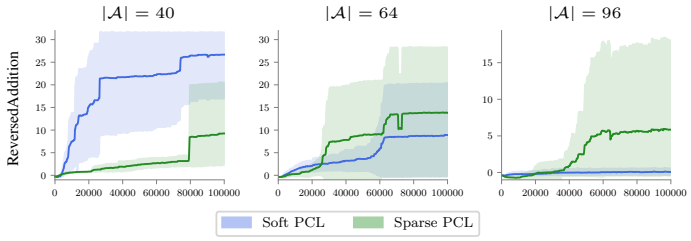

Figure 3. The average reward over training for sparse PCL compared to the standard soft PCL on ReversedAddition. In this task, the environment is a $2 \times n$ grid of digits representing two numbers in base-$|\mathcal{V}|$ that the agent needs to sum. As in the other tasks in the main paper, we see that sparse PCL becomes more advantageous compared to soft PCL as the action space increases in size.