Bridging the Gap Between Value and Policy Based Reinforcement Learning

Ofir Nachum[1] **Mohammad Norouzi** **Kelvin Xu**[1] **Dale Schuurmans**
{ofirnachum,mnorouzi,kelvinxx}@google.com, daes@ualberta.ca
Google Brain

Abstract

We formulate a new notion of *softmax* temporal consistency that generalizes the standard hard-max Bellman consistency usually considered in value based reinforcement learning (RL). In particular, we show how *softmax consistent* action values correspond to optimal policies that maximize entropy regularized expected reward. More importantly, we establish that softmax consistent action values and the optimal policy must satisfy a mutual compatibility property that holds across *any* state-action subsequence. Based on this observation, we develop a new RL algorithm, *Path Consistency Learning (PCL)*, that minimizes the total inconsistency measured along multi-step subsequences extracted from both on and off policy traces. An experimental evaluation demonstrates that PCL significantly outperforms strong actor-critic and Q-learning baselines across several benchmark tasks.

1. Introduction

Model-free RL aims to acquire an effective behavior policy through trial and error interaction with a black box environment. The goal is to optimize the quality of an agent's behavior policy, in terms of the total expected discounted reward. Model-free RL has a myriad of applications in games (Tesauro, 1995; Mnih et al., 2015), robotics (Kober et al., 2013; Levine et al., 2016), and marketing (Li et al., 2010; Theocharous et al., 2015), to name a few.

The two dominant approaches to developing model-free RL algorithms have been *value based* and *policy based*. Although these approaches have typically been considered distinct, recent work (Norouzi et al., 2016; O'Donoghue et al., 2017) has begun to relate the two by posing *entropy regularized* expected reward as the target objective (Williams & Peng, 1991). Given entropy regularization, the optimal policy π^* assigns a probability to an action sequence that is proportional to the exponentiated total reward received. In particular, given an episode of states, actions and rewards, $(s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T)$, where s_T is a terminal state, the optimal policy π^* satisfies,

$$\prod_{i=0}^{T-1} \pi^*(a_i \mid s_i) \propto \exp\left(\sum_{i=0}^{T-1} r_i/\tau\right), \qquad (1)$$

where $\tau \geq 0$ is an entropy regularization coefficient. Unfortunately, such results have only been established thus far for undiscounted finite horizon tasks.

This paper further develops a connection between value and policy based RL by introducing a new notion of temporal difference consistency in the presence of *discounted* entropy regularization, which not only allows extension to the infinite horizon case, but also reveals surprisingly strong new properties. Consider a transition where a state action pair (s, a) is followed by[2] a state s'. The literature currently considers two primary notions of temporal consistency, which we refer to as *average* (2) and *hard-max* (3) consistency respectively:

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{\pi(a'|s')} Q^\pi(s',a'), \qquad (2)$$
$$Q^\circ(s,a) = r(s,a) + \gamma \max_{a'} Q^\circ(s',a'). \qquad (3)$$

Here $0 \leq \gamma \leq 1$ is the reward *discount* factor. The constraint (2) applies to any policy π and serves as the basis of the *on-policy* one-step expected SARSA (Rummery & Niranjan, 1994; Singh & Sutton, 1996) and actor-critic (Mnih et al., 2016) algorithms; while the constraint (3) only applies to a maximum reward policy and is the basis of the *off-policy* Q-learning algorithm (Watkins, 1989).

Given γ-discounted entropy regularization, we formulate a new notion of *softmax* temporal consistency for optimal Q-values as:

$$Q^*(s,a) = r(s,a) + \gamma\tau \log \sum_{a'} \exp(Q^*(s',a')/\tau). \qquad (4)$$

Note that as $\tau \to 0$ the standard hard-max consistency (3) is recovered as a special case. For finite action spaces and

[1] Work done as a member of the Google Brain Residency program (g.co/brainresidency)

[2] To simplify matters, we focus on environments with a *deterministic* transition from (s, a) to s'. Such transitions may depend on a shared random seed, applicable to a wide range of tasks.

discrete state spaces, given a deterministic state transition function, we prove that a *unique* solution Q^* to the softmax consistency constraint (4) must exist. Further, we show that the policy $\pi^*(a \mid s) \propto \exp(Q^*(s,a)/\tau)$ defined from Q^* maximizes entropy regularized expected discounted reward. Crucially, this notion of softmax consistency can be extended beyond single step backups to derive multi-step path-wise consistency constraints *on arbitrary trajectories*, in contrast to the standard hard-max consistency (3).

Based on these developments, we propose a new RL algorithm, *Path Consistency Learning (PCL)*, that optimizes a policy by minimizing the path-wise temporal inconsistency on a set of on and off policy traces. PCL iteratively fits both a parameterized policy and a state value function (implicitly Q-values), bridging the gap between value and policy based methods. Unlike algorithms using Q^π-values, the proposed approach seamlessly combines on-policy and off-policy traces. Unlike algorithms based on hard-max consistency and Q°-values, the proposed formulation easily generalizes to multi-step backups on arbitrary paths, while maintaining proper regularization and consistent mathematical justification. We assess the effectiveness of PCL on a suite of RL benchmarks, and we find that it significantly outperforms a strong actor-critic and a strong Q-learning baseline.

In summary, the main contributions of this paper include:

- A complete characterization of *softmax* temporal consistency,[3] which generalizes the commonly used hard-max Bellman consistency.
- A proof that Q-values satisfying softmax temporal consistency directly determine the optimal policy that maximizes entropy regularized expected discounted reward.
- Identification of a new multi-step *path-wise* softmax consistency property that relates the optimal Q-values at the end points of *any path* to the log-probabilities of the optimal policy along actions of that path.
- An effective RL algorithm, Path Consistency Learning, that exploits multi-step path-wise consistency and combines elements of value and policy based RL.
- Strong experimental results versus current actor-critic and Q-learning baselines.

2. Notation and Background

We model an agent's policy by a parametric distribution π_θ over a finite set of actions. At iteration t, the agent encounters a state s_t and performs an action a_t sampled from $\pi_\theta(a \mid s_t)$. The environment then returns a scalar reward r_t and transitions to the next state s_{t+1}. To simplify matters, we restrict attention to deterministic environments, where the per-step reward r_t and the next state s_{t+1} are given by deterministic functions conditioned on a latent variable h (*e.g.,* a random seed, unknown to the agent), *i.e.,* $r_t = r(s_t, a_t \mid h)$, and $s_{t+1} = f(s_t, a_t \mid h)$ for functions r and f. Note that most RL benchmarks including Atari games fall into this category. Thus the latent variable h determines the set of states $S(h)$ including the initial state $s_0(h)$ and the set of transitions $E(h) = \{(s, a, s') \mid s' = f(s, a \mid h)\}$. For clarity of exposition, we focus on finite horizon environments in the main body of the paper, whereby $S(h)$ and $E(h)$ constitute vertices and edges in a directed acyclic graph. That said, all theorems and corollaries presented in the paper hold in the infinite horizon case, as proved in the Appendix. We denote $L(h) = \{s \in S(h) \mid \forall a, \forall s' \in S(h), (s, a, s') \notin E(h)\}$ as the set of terminal states in the graph.

For brevity, we henceforth drop h from our notation and assume that it is implicitly included. We will denote a trajectory in the graph as $s_{1:t} \equiv (s_1, a_1, \ldots, a_{t-1}, s_t)$. We denote a trajectory of length $t-1$ sampled from π_θ starting at s_1 (not necessarily the initial state s_0) as $s_{1:t} \sim \pi_\theta(s_{1:t})$. Note that $\pi_\theta(s_{1:t}) = \prod_{i=1}^{t-1} \pi_\theta(a_i \mid s_i)$. A trajectory sampled from π_θ at s_1, continuing until termination at $s_T \in L$, is denoted $s_{1:T} \sim \pi_\theta(s_{1:})$. When used in an expectation over paths, we omit the sampling distribution $\pi_\theta(\cdot)$ for brevity, but unless otherwise specified, all expectations are with respect to trajectories sampled from π_θ.

The two common definitions of action values, Q^π and Q°, have already been established in (2) and (3). However, it is also convenient to introduce state value functions, defined for the expected and hard-max cases respectively as

$$V^\pi(s) = \mathbb{E}_{\pi(a|s)} Q^\pi(s,a)\,, \tag{5}$$

$$V^\circ(s) = \max_a Q^\circ(s,a)\,. \tag{6}$$

The expected value function V^π is used in SARSA and actor-critic algorithms, while the hard-max value function V° serves as the basis of Q-learning and its many variants.

In particular, the actor-critic algorithm (Mnih et al., 2016) maintains an estimate of V^π to estimate the future expected rewards beyond a certain state. This allows for estimating the value of one or several actions without rolling out an entire trajectory until termination. In addition, V^π is used as a baseline to reduce the variance of gradients.

The Q-learning algorithm (Watkins, 1989) exploits the fact that the hard-max consistent Q-values, denoted Q°, induce a policy with maximal expected reward, where at each state s one chooses an action with largest $Q^\circ(s,a)$. The Q-learning algorithm then minimizes one-step hard-max inconsistencies induced by a Q-value approximator $Q_\phi(s,a)$.

[3] We use the term *softmax* to refer to the log-sum-exp function, and *soft indmax* to refer to the normalized exponential function corresponding to the (regrettably misnamed) "softmax activation function" that produces a vector of multinomial probabilities.

3. Softmax Temporal Consistency

We begin our formulation of softmax temporal consistency with a simple motivating example. Suppose an agent is at some state s_0 and faces n possible actions $\{a_1, \ldots, a_n\}$, each yielding an immediate reward in $\{r_1, \ldots, r_n\}$ and leading to a successor state in $\{s_1, \ldots, s_n\}$, where each successor has an associated estimate of future value (*i.e.,* state values) $\{v_1, \ldots, v_n\}$. Consider the problem of inferring the current state value v_0 assuming that a policy π has been locally optimized for the next action choice. In particular, we face a problem of optimizing π subject to $0 \leq \pi(a_i) \leq 1$ and $\sum_i \pi(a_i) = 1$ to maximize some objective. As we will see, different choices of objective lead to different forms of temporal consistency defining v_0.

First, consider the standard expected reward objective:

$$O_{\text{MR}}(\pi) = \sum_{i=1}^{n} \pi(a_i)(r_i + \gamma v_i^\circ), \tag{7}$$

where we suppose we have access to the O_{MR}-optimal state values at the successors $\{v_1^\circ, \ldots, v_n^\circ\}$. In this context, the optimal policy π° is a one-hot distribution with a probability of $\pi^\circ(a_m) = 1$ at an action a_m with maximal return, *i.e.,* $m = \text{argmax}_i(r_i + \gamma v_i^\circ)$, and zero elsewhere. Accordingly, the O_{MR}-optimal state value of s_0 based on $\{v_1^\circ, \ldots, v_n^\circ\}$ is given by

$$v_0^\circ = O_{\text{MR}}(\pi^\circ) = \max_i (r_i + \gamma v_i^\circ). \tag{8}$$

Alternatively, one can consider an *entropy regularized* expected reward objective:

$$O_{\text{ENT}}(\pi) = \sum_{i=1}^{n} \pi(a_i)(r_i + \gamma v_i^* - \tau \log \pi(a_i)), \tag{9}$$

where we have access to the O_{ENT}-optimal state values $\{v^*, \ldots, v_n^*\}$. It follows[4] that $\pi^*(a_i) \propto \exp\{(r_i + \gamma v_i^*)/\tau\}$, which, substituting back into the objective, yields

$$v_0^* = O_{\text{ENT}}(\pi^*) = \tau \log \sum_{i=1}^{n} \exp\{(r_i + \gamma v_i^*)/\tau\}. \tag{10}$$

This gives an intuitive definition of state values based on a softmax function that generalizes the hard-max state values defined above. The connection between this formulation of v_0^* and softmax Q-values defined in (4) is straightforward.

Crucially, $\exp\{v_0^*/\tau\}$ also serves as the normalization constant for π^* at s_0:

$$\pi^*(a_i) = \frac{\exp\{(r_i + \gamma v_i^*)/\tau\}}{\exp\{v_0^*/\tau\}} \,. \tag{11}$$

[4] The $O_{\text{ENT}}(\pi)$ objective is simply a τ-scaled, constant-shifted KL-divergence between π and $\pi^*(a_i) \propto \exp\{(r_i + \gamma v_i^*)/\tau\}$.

Manipulation of (11) by taking log of both sides, results in the following relationship between the optimal state value v_0^*, the optimal state values for each successor state v_i^*, and the log-probabilities of the optimal policy:

$$v_0^* = -\tau \log \pi^*(a_i) + r_i + \gamma v_i^* \,. \tag{12}$$

This relationship between optimal state values and the optimal policy can in fact be extended beyond a single step to a multi-step consistency. We will use the multi-step path-wise consistency to propose an RL algorithm that fits both policy and value estimates simultaneously.

We will develop these preliminary findings more generally below. In particular, we will define V^* in terms of Q^* to match the definition of v_0^* above. We will then express O_{ENT} for the more general sequential decision making case. Subsequently we will state the relationships between V^* and the optimal policy of O_{ENT} and note that the simple single-step identity stated in (12) is easily extended to a multi-step path-wise identity. Finally, we present an RL algorithm that fits a parameterized policy and value estimate to satisfy path-wise consistencies.

Note that the preceding logic can provide a basis for an inductive proof of the claims that follow in a finite horizon setting, although in the Appendix our proofs hold for the general infinite horizon setting.

3.1. Softmax Values

The softmax Q-value generalization of the hard-max Q° has already been presented in (4). Analogous to the fact that the expected and hard-max Q-values both have natural definitions of state value, so does the softmax Q-value. We define the softmax state value function as

$$V^*(s) \;=\; \tau \log \sum_a \exp\{Q^*(s,a)/\tau\} \,. \tag{13}$$

Note that this definition is equivalent to the one presented more simply in (10), since $Q^*(s,a) = r(s,a) + \gamma V^*(s')$. That is, one may recursively define V^* as

$$V^*(s) \;=\; \tau \log \sum_{a,s'} \exp\{(r(s,a) + \gamma V^*(s'))/\tau\} \,. \tag{14}$$

3.2. Discounted Entropy Regularization

We next present O_{ENT} in full generality. To account for discounting, we define a γ-discounted entropy

$$H^\gamma(s_1, \pi_\theta) = -\mathbb{E}_{s_{1:T}} \left[\sum_{i=1}^{T-1} \gamma^{i-1} \log \pi_\theta(a_i|s_i) \right].$$

We propose the following objective for optimizing π_θ:

$$O_{\text{ENT}}(s_0, \theta) = \mathbb{E}_{s_{0:T}}[R(s_{0:T})] + \tau H^\gamma(s_0, \pi_\theta), \tag{15}$$

where $R(s_{m:n}) = \sum_{i=0}^{n-m-1} \gamma^i r(s_{m+i}, a_{m+i})$ and τ is a user-specified temperature parameter. Optimizing this objective for s_0 is equivalent to optimizing $O_{\text{ENT}}(s_1, \theta)$ for all $s_1 \in S$. Rather than only maximizing the expected sum of future rewards, this objective maximizes the expected sum of future discounted rewards and τ-weighted log-probabilities.

In the case of $\gamma = 1$ this is equivalent to the entropy regularizer proposed in Williams & Peng (1991). While this objective is only a small modification of the maximum-reward objective, it alters the optimal policy π^* from a one-hot distribution on the maximal-reward path to a smoother distribution that assigns probabilities to all trajectories commensurate with their reward.

3.3. Consistency of Optimal Values and Policy

This section presents a general characterization of the connection between the optimal policy of O_{ENT} with the softmax consistent Q-values. The first observation is that the optimal policy π^* satisfies the same strong property given in (12) but now in the more general sequential decision making case. In particular, the optimal state value and optimal policy satisfy a strong form of local consistency for *every* state-action-successor tuple.

Theorem 1. *The optimal policy π^* for $O_{ENT}(s_0, \theta)$ and the state values V^* defined in (13) satisfy*

$$V^*(s) = -\tau \log \pi^*(a|s) + r(s, a) + \gamma V^*(s'), \qquad (16)$$

for any $(s, a, s') \in E$.

Proof. All theorems and corollaries are proved in the Appendix. □

This result also allows us to characterize π^* in terms of Q^*.

Corollary 2. *The optimal policy π^* may be given in terms of the softmax Q-values Q^* as*

$$\pi^*(a|s) = \exp\{(Q^*(s, a) - V^*(s))/\tau\}.$$

Next, we make the key observation that the form of one-step temporal consistency identified in Theorem 1 can in fact be extended to any *sequence* of state-action-successor tuples. That is, the softmax state values at the start and end state of any trajectory can be related to the rewards and optimal log-probabilities along that trajectory.

Corollary 3. *Given the optimal π^* for $O_{ENT}(s_0, \theta)$ and the corresponding value mapping V^*, then every trajectory $s_{1:t}$ satisfies*

$$-V^*(s_1) + \gamma^{t-1} V^*(s_t) + R(s_{1:t}) - \tau G(s_{1:t}, \pi^*) = 0,$$

where

$$G(s_{m:n}, \pi_\theta) = \sum_{i=0}^{n-m-1} \gamma^i \log \pi_\theta(a_{m+i}|s_{m+i}).$$

Importantly, the converse of Theorem 1 also holds, which opens the door to using path-wise consistency as the foundation for an objective for learning parameterized policy and value estimates.

Theorem 4. *If a policy π_θ and a value function V_ϕ satisfy the consistency property (16) for all tuples $(s, a, s') \in E$, then $\pi_\theta = \pi^*$ and $V_\phi = V^*$.*

3.4. A Path-wise Objective

These properties of the optimal policy and softmax state values lead us to propose a natural path-wise objective for training a parameterized π_θ and V_ϕ. We define the consistency error for a trajectory $s_{1:t}$ under π_θ, V_ϕ as

$$C_{\theta,\phi}(s_{1:t}) = -V_\phi(s_1) + \gamma^{t-1} V_\phi(s_t) + R(s_{1:t}) - \tau G(s_{1:t}, \pi_\theta). \qquad (17)$$

This definition may be extended to trajectories that terminate before step t by considering all rewards and log-probabilities after a terminal state as 0.

Then we may define a minimizing objective for every trajectory as

$$O_{\text{PCL}}(s_{1:t}, \theta, \phi) = \frac{1}{2} C_{\theta,\phi}(s_{1:t})^2. \qquad (18)$$

The gradients of this objective lead us to the following updates for θ and ϕ:

$$\Delta\theta \propto C_{\theta,\phi}(s_{1:t}) \nabla_\theta G(s_{1:t}, \pi_\theta), \qquad (19)$$

$$\Delta\phi \propto C_{\theta,\phi}(s_{1:t}) \left(\nabla_\phi V_\phi(s_1) - \gamma^{t-1} \nabla_\phi V_\phi(s_t)\right). \qquad (20)$$

3.5. Path Consistency Learning

Given the path-wise objectives proposed above and their resulting parameter updates, we propose a new RL algorithm we call *Path Consistency Learning (PCL)*. Unlike actor-critic variants, PCL utilizes both on-policy and off-policy trajectory samples and uses the two sampling methods to optimize a path-wise consistency. Unlike Q-learning variants, PCL optimizes towards a consistency that holds path-wise and not only on single steps. The pseudocode of PCL is presented in Algorithm 1. Given a rollout parameter d, at each iteration, PCL samples a batch of on-policy trajectories and computes the corresponding parameter updates for

Algorithm 1 Path Consistency Learning

Input: Environment ENV, learning rates η_π, η_ϕ, discount factor γ, rollout d, number of steps N, replay buffer capacity B, prioritized replay hyperparameter α.
function Gradients($s_{0:T}$)
// Recall $G(s_{t:t+d}, \pi_\theta)$ is a discounted sum of log-probabilities from s_t to s_{t+d}.
Compute $\Delta\theta = \sum_{t=0}^{T-1} C_{\theta,\phi}(s_{t:t+d}) \nabla_\theta G(s_{t:t+d}, \pi_\theta)$.
Compute $\Delta\phi = \sum_{t=0}^{T-1} C_{\theta,\phi}(s_{t:t+d}) \left(\nabla_\phi V_\phi(s_t) - \gamma^d \nabla_\phi V_\phi(s_{t+d})\right)$.
Return $\Delta\theta, \Delta\phi$
end function
Initialize θ, ϕ.
Initialize empty replay buffer $RB(\alpha)$.
for $i = 0$ **to** $N - 1$ **do**
Sample $s_{0:T} \sim \pi_\theta(s_{0:})$ on ENV.
$\Delta\theta, \Delta\phi =$ Gradients($s_{0:T}$).
Update $\theta \leftarrow \theta + \eta_\pi \Delta\theta$.
Update $\phi \leftarrow \phi + \eta_\phi \Delta\phi$.
Input $s_{0:T}$ into RB with priority $R^1(s_{0:T})$.
If $|RB| > B$, remove episodes uniformly at random.
Sample $s_{0:T}$ from RB.
$\Delta\theta, \Delta\phi =$ Gradients($s_{0:T}$).
Update $\theta \leftarrow \theta + \eta_\pi \Delta\theta$.
Update $\phi \leftarrow \phi + \eta_\phi \Delta\phi$.
end for

each d-length sub-trajectory. PCL also takes advantage of off-policy trajectories by maintaining a replay buffer. We found it beneficial to sample replay episodes using a distribution proportional to exponentiated reward mixed with a uniform distribution, although we did not exhaustively optimize this sampling technique. Specifically, we sample an episode $s_{0:T}$ from the replay buffer of size B with probability $0.1/B + 0.9 \cdot \exp(\alpha R^1(s_{0:T}))/Z$, where we use a discount of 1 on the sum rewards, Z is a normalizing factor, and α is a fixed hyperparameter.

While we focused our experiments on environments with relatively short episodes (length at most 100), in environments with longer episodes the algorithm may be altered to be applied on sub-episodes of manageable length. It may also be beneficial to use multiple rollout lengths d and optimize consistency at multiple scales, although we did not explore this.

3.6. Comparison to Other Algorithms

A reader familiar with advantage-actor-critic (Mnih et al., 2016) (A2C and its asynchronous analogue A3C) should already notice the similarities between our updates and those of actor-critic. Actor-critic takes advantage of the expected value function V^π to reduce the variance of policy gradient updates for maximizing expected reward. As in PCL, two models are trained concurrently: an actor π_θ that determines the policy and a critic V_ϕ that is trained to estimate V^{π_θ}. A fixed rollout parameter d is chosen and the discounted future reward of a trajectory $s_{1:T} \sim \pi_\theta(s_{1:})$ is estimated as $R(s_{1:d+1}) + \gamma^d V_\phi(s_{d+1})$. The advantage of a trajectory $s_{1:T}$ is defined as

$$A_{\theta,\phi}(s_{1:d+1}) = -V_\phi(s_1) + \gamma^d V_\phi(s_{d+1}) + R(s_{1:d+1}),$$

where in the case of the trajectory terminating before step s_{d+1} one considers all rewards after a terminal state as 0. Many variants of actor-critic focus on modifications to this advantage estimate to reduce its variance when $s_{1:d+1}$ is sampled from π_θ (Schulman et al., 2016).

The actor-critic updates for θ and ϕ may be written in terms of the advantage:

$$\Delta\theta \propto \mathbb{E}_{s_{0:T}} \left[\sum_{i=0}^{T-1} A_{\theta,\phi}(s_{i:i+d}) \nabla_\theta \log \pi_\theta(a_i|s_i) \right], \quad (21)$$

$$\Delta\phi \propto \mathbb{E}_{s_{0:T}} \left[\sum_{i=0}^{T-1} A_{\theta,\phi}(s_{i:i+d}) \nabla_\phi V_\phi(s_i) \right]. \quad (22)$$

These updates exhibit a striking similarity to the updates expressed in (19) and (20). Indeed, we may interpret $C_{\theta,\phi}(s_{1:t})$ as a sort of advantage of a trajectory, and the update in (19) pushes the log-probabilities of the actions on that trajectory in the direction of the advantage.

However, one difference is that the actor-critic definition of advantage $A_{\theta,\phi}(s_{1:t})$ is a measure of the advantage of the trajectory $s_{1:t}$ compared to the average trajectory chosen by π_θ starting from s_1 in terms of reward. By contrast, $C_{\theta,\phi}(s_{1:t})$ can be seen as measuring the advantage of the rewards along the trajectory compared to the log-probability of the policy π_θ. At the optimal policy, when the log-probability of the policy is proportional to rewards, this measure of advantage will be 0 on every trajectory, which is not the case for $A_{\theta,\phi}(s_{1:t})$.

It is also important to note that V_ϕ is no longer an estimate of expected future reward under the current policy. In actor-critic, it is essential that V_ϕ track the non-stationary target V^{π_θ} in order to achieve suitable variance reduction. While in PCL $V^*(s)$ is the value of $O_{\text{ENT}}(s, \pi^*)$, during training V_ϕ need not be interpreted as a value dependent on the current policy π_θ.

Moreover, in actor-critic the expectations in (21) and (22) need to be estimated via Monte Carlo sampling from π_θ. In PCL, there is no such condition.

We may also compare PCL to hard-max temporal consistency RL algorithms, such as Q-learning (Watkins, 1989). Note that these hard-max consistencies only apply on a single step, while the temporal consistencies we consider apply to trajectories of any length. While some have proposed

using multi-step backups for hard-max Q-learning (Peng & Williams, 1996; Mnih et al., 2016), such an approach is not theoretically sound, since the rewards received after a non-optimal action do not relate to the hard-max Q-values Q°. Therefore, one can interpret the notion of temporal consistency we propose as the proper generalization of the one-step temporal consistency given by hard-max Q-values.

4. Related Work

There has been a surge of recent interest in using neural networks for both value and policy based reinforcement learning (*e.g.,* Mnih et al. (2013); Schulman et al. (2015); Levine et al. (2016); Silver et al. (2016)). We highlight several lines of work that are most relevant to this paper.

Recent work has noted some connections between value and policy based RL (Norouzi et al., 2016; O'Donoghue et al., 2017; Nachum et al., 2017), when an entropy regularized objective is considered. In particular, O'Donoghue et al. (2017) show that one can interpret policy based methods as a form of advantage function learning. They provide an analysis relating the optimal policy to the hard-max Q-values in the limit of $\tau = 0$, and thus propose to augment the actor-critic objective with offline updates that minimize a set of single-step hard-max Bellman errors. Accordingly, they combine (2) and (3) to propose a method called PGQ (policy gradient + Q-learning). By contrast, we extend the relationship to $\tau > 0$ by exploiting a notion of softmax consistency of Q-values (4). This exact softmax consistency has previously been considered by Ziebart (2010) in the context of continous control. In this work however, we highlight its significance as it gives rise to a notion of multi-step path-wise consistency.

The key idea of including a maximum entropy regularizer to encourage exploration is common in RL (Williams & Peng, 1991; Mnih et al., 2016) and inverse RL (Ziebart et al., 2008). Our proposed *discounted* entropy penalty generalizes the approach originally proposed in (Williams & Peng, 1991) beyond $\gamma = 1$, enabling applicability to infinite horizon problems. Previous work has extensively studied other exploration strategies including predictive error (Stadie et al., 2015), count based exploration (Bellemare et al., 2016), information theoretic notions of curiosity (Singh et al., 2004; Schmidhuber, 2010), and under-appreciated reward exploration (Nachum et al., 2017). We note that these methods often modify the reward function of an underlying MDP to include an *exploration bonus*. This can be easily coupled with our approach here.

Our proposed PCL algorithm bears some similiarity to multi-step Q-learning (Peng & Williams, 1996), which rather than minimizing one-step hard-max Bellman error, optimizes a Q-value function approximator by unrolling the trajectory for some number of steps before using a hard-max backup. While this method has shown some empirical success (Mnih et al., 2016), its theoretical justification is lacking, since rewards received after a non-optimal action do not relate to the hard-max Q-values Q° anymore. On the other hand, our algorithm incorporates the log-probabilities of the actions on a multi-step rollout, which is crucial for our softmax consistency.

Some other similar notions of soft temporal consistency have been previously discussed in the RL literature. Littman (1996) and Azar et al. (2012) use a *Boltzmann weighted average* operator. This operator is used by Azar et al. (2012) to propose an iterative algorithm converging to the optimal maximum reward policy inspired by the work of Kappen (2005); Todorov et al. (2016). While they use the Boltzmann weighted average, they briefly mention that a softmax (log-sum-exp) operator would have similar theoretical properties. More recently Fox et al. (2016) use a log-*weighted-average*-exp and Asadi & Littman (2016) propose a mellowmax operator, defined as log-*average*-exp. These log-average-exp operators share a similar non-expansion property with ours, and their proofs of non-expansion are very related. Additionally it is possible to show that when restricted to an infinite horizon setting, the fixed point of the mellowmax operator is a constant shift of our Q^*. In all of these cases, the proposed training algorithm optimizes a single-step consistency unlike PCL, which optimizes a multi-step consistency. Moreover, these papers do not present a clear relationship between the action values at the fixed point and the entropy regularized expected reward objective, which was the key to our formulation and algorithmic development.

Finally, there has been a considerable amount of work in reinforcement learning using off-policy data to design more sample efficient algorithms. Broadly, these methods can be understood as trading off bias (Sutton et al., 1999; Silver et al., 2014; Lillicrap et al., 2016; Gu et al., 2016) and variance (Precup, 2000; Munos et al., 2016). Previous work that has considered multi-step off-policy learning has typically used a correction (*e.g.,* via importance-sampling (Precup et al., 2001) or truncated importance sampling with bias correction (Munos et al., 2016), or eligibility traces (Precup, 2000)). By contrast, our method defines an unbiased consistency for an entire trajectory applicable to on and off policy data. That said, an empirical comparison between all of these methods and PCL is an interesting avenue for future work.

5. Experiments

We evaluated PCL across several different tasks and compared it to an A3C implementation based on Mnih et al. (2016) and a double Q-learning with prioritized experience

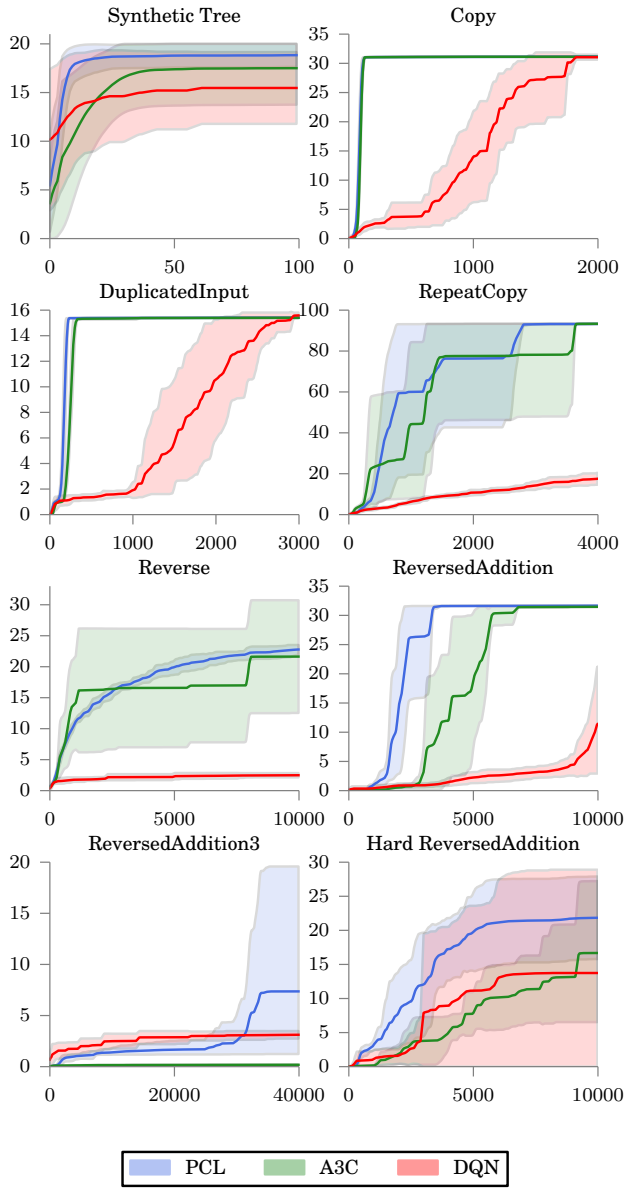

Figure 1. The results of PCL against the two baselines A3C and DQN. Each plot shows average reward across 5 randomly seeded training runs (10 for Synthetic Tree) after choosing optimal hyperparameters. We also show the single standard deviation error clipped at the min and max. The x-axis is number of training iterations. PCL exhibits comparable performance to A3C in some tasks, but clearly outperforms it on the more challenging tasks. Across all tasks, the performance of DQN is worse than PCL.

replay implementation based on Schaul et al. (2016).

5.1. A Synthetic Environment

As an initial testbed, we developed a simple synthetic environment. The environment is defined by a binary decision tree of depth 20. For each training run, the reward on each edge is sampled uniformly from $[-1, 1]$ and subsequently normalized so that the maximal reward trajectory has total reward 20. We trained using a fully-parameterized model: for each node s in the decision tree there are two parameters to determine the logits of $\pi_\theta(-|s)$ and one parameter to determine $V_\phi(s)$. In the Q-learning implementation only two parameters per node s are needed to determine $Q_\phi(s, -)$.

5.2. Algorithmic Tasks

For more complex environments, we evaluated PCL and the two baselines on the algorithmic tasks from the OpenAI Gym library (Brockman et al., 2016). This library provides six tasks, in rough order of difficulty: Copy, DuplicatedInput, RepeatCopy, Reverse, ReversedAddition, and ReversedAddition3. In each of these tasks, an agent operates on a grid of characters or digits, observing one character or digit at a time. At each time step, the agent may move one step in any direction and optionally write a character or digit to output. A reward is received on each correct emission. The agent's goal for each task is:

- **Copy:** Copy a $1 \times n$ sequence of characters to output.
- **DuplicatedInput:** Deduplicate a $1 \times n$ sequence of characters.
- **RepeatCopy:** Copy a $1 \times n$ sequence of characters first in forward order, then reverse, and finally forward again.
- **Reverse:** Copy a $1 \times n$ sequence of characters in reverse order.
- **ReversedAddition:** Observe two ternary numbers in little-endian order via a $2 \times n$ grid and output their sum.
- **ReversedAddition3:** Observe three ternary numbers in little-endian order via a $3 \times n$ grid and output their sum.

These environments have an implicit curriculum associated with them. To observe the performance of our algorithm without curriculum, we also include a task "Hard ReversedAddition" which has the same goal as ReversedAddition but does not utilize curriculum.

For these environments, we parameterized the agent by a recurrent neural network with LSTM (Hochreiter & Schmidhuber, 1997) cells of hidden dimension 128.

Our algorithm is easily amenable to the incorporation of expert trajectories. Thus, for the algorithmic tasks we also experimented with seeding the replay buffer with 10 randomly sampled expert trajectories. During training we ensured that these trajectories are not be removed from the replay buffer and always have maximal priority.

5.3. Results

We present the results of each of the three variants (PCL, A3C, DQN) in Figure 1. After finding the best hyper-

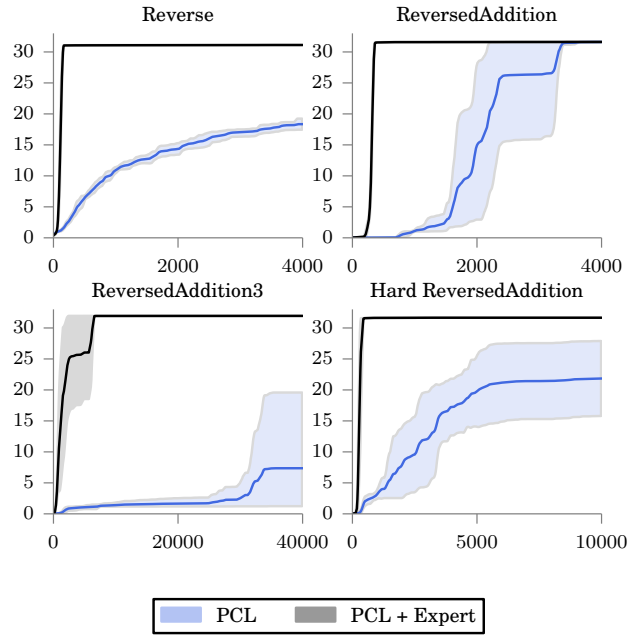

Figure 2. The results of PCL and PCL augmented with a small number of expert trajectories on the four hardest algorithmic tasks. We see that incorporating expert trajectories greatly improves performance.

parameters (see Section 5.4), we plot the average reward over training iterations for five randomly seeded runs. For the Synthetic Tree environment, the same protocol is performed but with ten seeds shared across variants.

The gap between PCL and A3C is hard to discern in some of the more simple tasks such as Copy, Reverse, and RepeatCopy. However, a noticeable gap exists in the Synthetic Tree and DuplicatedInput results and more significant gaps are clear in the harder tasks: ReversedAddition, ReversedAddition3, and Hard ReversedAddition. Across all the experiments, it is clear that the prioritized DQN performs worse than PCL. These results show the viability of PCL as an RL algorithm that can be competitive and in some cases significantly improve upon strong baselines.

We present the results of PCL along with PCL augmented with expert trajectories in Figure 2. We observe that the incorporation of expert trajectories helps a considerable amount. Despite the number of expert trajectories we provide being small (10) compared to the batch size (400), their inclusion in the training process significantly improves the agent's performance. Incorporating expert trajectories in PCL is relatively trivial compared to the specialized methods developed for other policy based algorithms. While we did not compare to other algorithms that take advantage of expert trajectories, this success shows the promise of using path-wise consistencies. The ability of PCL to easily incorporate expert trajectories is a very desirable property in real-world applications such as robotics.

5.4. Implementation Details

For our hyperparameter search, we found it simple to parameterize the critic learning rate in terms of the actor learning rate as $\eta_\phi = C\eta_\pi$, where C is the *critic weight*.

For the Synthetic Tree environment we used a batch size of 10, rollout of $d = 3$, discount of $\gamma = 1.0$, and a replay buffer capacity of 10,000. We fixed the α parameter for PCL's replay buffer to 1 and used $\epsilon = 0.05$ for DQN. To find the optimal hyperparameters, we performed an extensive grid search over actor learning rate $\eta_\pi \in \{0.01, 0.05, 0.1\}$; critic weight $C \in \{0.1, 0.5, 1\}$; entropy regularizer $\tau \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1.0\}$ for A3C, PCL; and $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}, \beta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ for DQN replay buffer parameters. We used standard gradient descent for optimization.

For the algorithmic tasks we used a batch size of 400, rollout of $d = 10$, a replay buffer of capacity 100,000, ran using distributed training with 4 workers, and fixed the actor learning rate η_π to 0.005, which we found to work well across all variants. To find the optimal hyperparameters, we performed an extensive grid search over discount $\gamma \in \{0.9, 1.0\}$, $\alpha \in \{0.1, 0.5\}$ for PCL's replay buffer; critic weight $C \in \{0.1, 1\}$; entropy regularizer $\tau \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.15\}$; $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$, $\beta \in \{0.06, 0.2, 0.4, 0.5, 0.8\}$ for the prioritized DQN replay buffer; and also experimented with exploration rates $\epsilon \in \{0.05, 0.1\}$ and copy frequencies for the target DQN, $\{100, 200, 400, 600\}$. In these experiments, we used the Adam optimizer (Kingma & Ba, 2015).

All experiments were implemented using Tensorflow (Abadi et al., 2016).

6. Conclusion

We study the characteristics of the optimal policy and state values for a maximum reward objective in the presence of *discounted entropy regularization*. We prove interesting softmax consistency relations between the optimal policy and optimal state values, which generalize hard-max Bellman consistency in the absence of entropy. The softmax consistency leads us to develop Path Consistency Learning (PCL), an RL algorithm that has a flavor similar to both actor-critic in that it maintains and jointly learns a model of the state values and a model of the policy, and Q-learning in that it minimizes a measure of temporal inconsistency. Unlike standard RL algorithms, PCL applies to both on-policy and off-policy trajectory samples. Further, one-step softmax consistency naturally generalizes to a multi-step path-wise consistency, which is employed by PCL. Empirically, PCL exhibits a significant improvement over baseline methods across several algorithmic benchmarks.

7. Acknowledgment

We thank Rafael Cosman, Brendan O'Donoghue, Volodymyr Mnih, George Tucker, Irwan Bello, and the Google Brain team for insightful comments and discussions.

References

Abadi, Martín, Barham, Paul, Chen, Jianmin, Chen, Zhifeng, Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Irving, Geoffrey, Isard, Michael, et al. Tensorflow: A system for large-scale machine learning. *arXiv:1605.08695*, 2016.

Asadi, Kavosh and Littman, Michael L. A new softmax operator for reinforcement learning. *arXiv preprint arXiv:1612.05628*, 2016.

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

Bellemare, Marc, Srinivasan, Sriram, Ostrovski, Georg, Schaul, Tom, Saxton, David, and Munos, Remi. Unifying count-based exploration and intrinsic motivation. *NIPS*, 2016.

Bertsekas, Dimitri P. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 1995.

Borwein, J. and Lewis, A. *Convex Analysis and Nonlinear Optimization*. Springer, 2000.

Brockman, Greg, Cheung, Vicki, Pettersson, Ludwig, Schneider, Jonas, Schulman, John, Tang, Jie, and Zaremba, Wojciech. OpenAI Gym. arXiv:1606.01540, 2016.

Fox, Roy, Pakman, Ari, and Tishby, Naftali. G-learning: Taming the noise in reinforcement learning via soft updates. *Uncertainty in Artifical Intelligence*, 2016. URL http://arxiv.org/abs/1512.08562.

Gu, Shixiang, Holly, Ethan, Lillicrap, Timothy, and Levine, Sergey. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. *arXiv preprint arXiv:1610.00633*, 2016.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural Comput.*, 1997.

Kappen, Hilbert J. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005(11):P11011, 2005.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *ICLR*, 2015.

Kober, Jens, Bagnell, J Andrew, and Peters, Jan. Reinforcement learning in robotics: A survey. *IJRR*, 2013.

Levine, Sergey, Finn, Chelsea, Darrell, Trevor, and Abbeel, Pieter. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39): 1–40, 2016.

Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. 2010.

Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Slillicrap2015continuous, oilver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. *ICLR*, 2016.

Littman, Michael Lederman. *Algorithms for sequential decision making*. PhD thesis, Brown University, 1996.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Graves, Alex, Antonoglou, Ioannis, Wierstra, Daan, and Riedmiller, Martin A. Playing atari with deep reinforcement learning. *arXiv:1312.5602*, 2013.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Mnih, Volodymyr, Badia, Adria Puigdomenech, Mirza, Mehdi, Graves, Alex, Lillicrap, Timothy P, Harley, Tim, Silver, David, and Kavukcuoglu, Koray. Asynchronous methods for deep reinforcement learning. *ICML*, 2016.

Munos, Remi, Stepleton, Tom, Harutyunyan, Anna, and Bellemare, Marc. Safe and efficient off-policy reinforcement learning. *NIPS*, 2016.

Nachum, Ofir, Norouzi, Mohammad, and Schuurmans, Dale. Improving policy gradient by exploring underappreciated rewards. *ICLR*, 2017.

Norouzi, Mohammad, Bengio, Samy, Chen, Zhifeng, Jaitly, Navdeep, Schuster, Mike, Wu, Yonghui, and Schuurmans, Dale. Reward augmented maximum likelihood for neural structured prediction. *NIPS*, 2016.

O'Donoghue, Brendan, Munos, Remi, Kavukcuoglu, Koray, and Mnih, Volodymyr. Pgq: Combining policy gradient and q-learning. *ICLR*, 2017.

Peng, Jing and Williams, Ronald J. Incremental multi-step q-learning. *Machine learning*, 22(1-3):283–290, 1996.

Precup, Doina. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Precup, Doina, Sutton, Richard S, and Dasgupta, Sanjoy. Off-policy temporal-difference learning with function approximation. 2001.

Rummery, Gavin A and Niranjan, Mahesan. *On-line Q-learning using connectionist systems*. University of Cambridge, 1994.

Schaul, Tom, Quan, John, Antonoglou, Ioannis, and Silver, David. Prioritized experience replay. *ICLR*, 2016.

Schmidhuber, Jürgen. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Schulman, John, Levine, Sergey, and Jordan, Michael I. Trust region policy optimization. *ICML*, 2015.

Schulman, John, Moritz, Philipp, Levine, Sergey, Jordan, Michael, and Abbeel, Pieter. High-dimensional continuous control using generalized advantage estimation. *ICLR*, 2016.

Silver, David, Lever, Guy, Heess, Nicolas, Degris, Thomas, Wierstra, Daan, and Riedmiller, Martin. Deterministic policy gradient algorithms. *ICML*, 2014.

Silver, David, Huang, Aja, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016.

Singh, Satinder P and Sutton, Richard S. Reinforcement learning with replacing eligibility traces. *Mach. Learn. J.*, 1996.

Singh, Satinder P, Barto, Andrew G, and Chentanez, Nuttapong. Intrinsically motivated reinforcement learning. *NIPS*, 2004.

Stadie, Bradly C, Levine, Sergey, and Abbeel, Pieter. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Sutton, Richard S, McAllester, David A, Singh, Satinder P, Mansour, Yishay, et al. Policy gradient methods for reinforcement learning with function approximation. *NIPS*, 1999.

Tesauro, Gerald. Temporal difference learning and td-gammon. *CACM*, 1995.

Theocharous, Georgios, Thomas, Philip S, and Ghavamzadeh, Mohammad. Personalized ad recommendation systems for life-time value optimization with guarantees. *IJCAI*, 2015.

Todorov, Emanuel et al. Linearly-solvable markov decision problems. *NIPS*, 2016.

Tsitsiklis, John N and Van Roy, Benjamin. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5): 674–690, 1997.

Watkins, Christopher JCH. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.

Williams, Ronald J and Peng, Jing. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 1991.

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

Ziebart, Brian D, Maas, Andrew L, Bagnell, J Andrew, and Dey, Anind K. Maximum entropy inverse reinforcement learning. *AAAI*, 2008.

A. Proofs

Here, we provide a proof of the main path consistency theorems. We first establish the basic results for a simple one-shot decision making setting. These initial results will be useful in the proof of the general infinite horizon setting.

For the more general infinite horizon setting, we introduce and discuss the entropy regularized expected return O_{ENT} and define a "softmax" operator $\mathcal{B}^*$ (analogous to the Bellman operator for hard-max Q-values). We then show the existence of a unique fixed point V^* of $\mathcal{B}^*$, by establishing that the softmax Bellman operator ($\mathcal{B}^*$) is a contraction under the infinity norm. We then aim to relate V^* to the optimal value of the entropy regularized expected reward objective O_{ENT} which we term $V^\dagger$. We are able to show that $V^* = V^\dagger$, as expected. Subsequently, we present a policy determined by V^* which satisfies $V^*(s) = O_{\text{ENT}}(s, \pi^*)$. Then given the characterization of π^* in terms of V^*, we establish the consistency property stated in Theorem 1 of the main text. Finally, we show that a consistent solution is optimal by satisfying the KKT conditions of the constrained optimization problem (establishing Theorem 4 of the main text).

A.1. Basic results for one-shot entropy regularized optimization

For $\tau > 0$ and any vector $\mathbf{q} \in \mathbb{R}^n$, $n < \infty$, define the scalar valued function F_τ (the "softmax") by

$$F_\tau(\mathbf{q}) = \tau \log \left(\sum_{a=1}^{n} e^{q_a/\tau} \right) \tag{23}$$

and define the vector valued function $\mathbf{f}_\tau$ (the "soft indmax") by

$$\mathbf{f}_\tau(\mathbf{q}) = \frac{e^{\mathbf{q}/\tau}}{\sum_{a=1}^{n} e^{q_a/\tau}} = e^{(\mathbf{q} - F_\tau(\mathbf{q}))/\tau}, \tag{24}$$

where the exponentiation is component-wise. It is easy to verify that $\mathbf{f}_\tau = \nabla F_\tau$. Note that $\mathbf{f}_\tau$ maps any real valued vector to a probability vector. We denote the probability simplex by $\Delta = \{\boldsymbol{\pi} : \boldsymbol{\pi} \geq 0, \mathbf{1} \cdot \boldsymbol{\pi} = 1\}$, and denote the entropy function by $H(\boldsymbol{\pi}) = -\boldsymbol{\pi} \cdot \log \boldsymbol{\pi}$.

Lemma 5.

$$F_\tau(\mathbf{q}) = \max_{\boldsymbol{\pi} \in \Delta} \Big\{ \boldsymbol{\pi} \cdot \mathbf{q} + \tau H(\boldsymbol{\pi}) \Big\} \tag{25}$$

$$= \mathbf{f}_\tau(\mathbf{q}) \cdot \mathbf{q} + \tau H(\mathbf{f}_\tau(\mathbf{q})) \tag{26}$$

Proof. First consider the constrained optimization problem on the right hand side of (25). The Lagrangian is given by $L = \boldsymbol{\pi} \cdot (\mathbf{q} - \tau \log \boldsymbol{\pi}) + \lambda(1 - \mathbf{1} \cdot \boldsymbol{\pi})$, hence $\nabla L = \mathbf{q} - \tau \log \boldsymbol{\pi} - \tau - \lambda$. The KKT conditions for this optimization problems are the following system of $n + 1$ equations

$$\mathbf{1} \cdot \boldsymbol{\pi} = 1 \tag{27}$$

$$\tau \log \boldsymbol{\pi} = \mathbf{q} - v \tag{28}$$

for the $n + 1$ unknowns, $\boldsymbol{\pi}$ and v, where $v = \lambda + \tau$. Note that for any v, satisfying (28) requires the unique assignment $\boldsymbol{\pi} = \exp((\mathbf{q} - v)/\tau)$, which also ensures $\boldsymbol{\pi} > 0$. To subsequently satisfy (27), the equation $1 = \sum_a \exp((q_a - v)/\tau) = e^{-v/\tau} \sum_a \exp(q_a/\tau)$ must be solved for v; since the right hand side is strictly decreasing in v, the solution is also unique and in this case given by $v = F_\tau(\mathbf{q})$. Therefore $\boldsymbol{\pi} = \mathbf{f}_\tau(\mathbf{q})$ and $v = F_\tau(\mathbf{q})$ provide the unique solution to the KKT conditions (27)-(28). Since the objective is strictly concave, $\boldsymbol{\pi}$ must be the unique global maximizer, establishing (26). It is then easy to show $F_\tau(\mathbf{q}) = \mathbf{f}_\tau(\mathbf{q}) \cdot \mathbf{q} + \tau H(\mathbf{f}_\tau(\mathbf{q}))$ by algebraic manipulation, which establishes (25). □

Corollary 6 (Optimality Implies Consistency)**.** *If* $v^* = \max_{\boldsymbol{\pi} \in \Delta} \Big\{ \boldsymbol{\pi} \cdot \mathbf{q} + \tau H(\boldsymbol{\pi}) \Big\}$ *then*

$$v^* = q_a - \tau \log \pi^*_a \ \textit{for all}\ a, \tag{29}$$

where $\boldsymbol{\pi}^* = \mathbf{f}_\tau(\mathbf{q})$.

Proof. From Lemma 5 we know $v^* = F_\tau(\mathbf{q}) = \boldsymbol{\pi}^* \cdot (\mathbf{q} - \tau \log \boldsymbol{\pi}^*)$ where $\boldsymbol{\pi}^* = \mathbf{f}_\tau(\mathbf{q})$. From the definition of $\mathbf{f}_\tau$ it also follows that $\log \pi^*_a = (q_a - F_\tau(\mathbf{q}))/\tau$ for all a, hence $v^* = F_\tau(\mathbf{q}) = q_a - \tau \log \pi^*_a$ for all a. □

Corollary 7 (Consistency Implies Optimality)**.** *If $v \in \mathbb{R}$ and $\boldsymbol{\pi} \in \Delta$ jointly satisfy*

$$v = q_a - \tau \log \pi_a \quad \textit{for all } a, \tag{30}$$

then $v = F_\tau(\mathbf{q})$ and $\boldsymbol{\pi} = \mathbf{f}_\tau(\mathbf{q})$; that is, $\boldsymbol{\pi}$ must be an optimizer for (25) and v is its corresponding optimal value.

Proof. Any v and $\boldsymbol{\pi} \in \Delta$ that jointly satisfy (30) must also satisfy the KKT conditions (27)-(28); hence $\boldsymbol{\pi}$ must be the unique maximizer for (25) and v its corresponding objective value. □

Although these results are elementary, they reveal a strong connection between optimal state values (v), optimal action values ($\mathbf{q}$) and optimal policies ($\boldsymbol{\pi}$) under the softmax operators. In particular, Lemma 5 states that, if $\mathbf{q}$ is an optimal action value at some current state, the optimal state value must be $v = F_\tau(\mathbf{q})$, which is simply the entropy regularized value of the optimal policy, $\boldsymbol{\pi} = \mathbf{f}_\tau(\mathbf{q})$, at the current state.

Corollaries 6 and 7 then make the stronger observation that this mutual consistency between the optimal state value, optimal action values and optimal policy probabilities must hold for *every* action, not just in expectation over actions sampled from $\boldsymbol{\pi}$; and furthermore that achieving mutual consistency in this form is *equivalent* to achieving optimality.

Below we will also need to make use of the following properties of F_τ.

Lemma 8. *For any vector $\mathbf{q}$,*

$$F_\tau(\mathbf{q}) = \sup_{\mathbf{p} \in \Delta} \Big\{ \mathbf{p} \cdot \mathbf{q} - \tau \mathbf{p} \cdot \log \mathbf{p} \Big\}. \tag{31}$$

Proof. Let F_τ^* denote the conjugate of F_τ, which is given by

$$F_\tau^*(\mathbf{p}) = \sup_{\mathbf{q}} \Big\{ \mathbf{q} \cdot \mathbf{p} - F_\tau(\mathbf{q}) \Big\} = \tau \mathbf{p} \cdot \log \mathbf{p} \tag{32}$$

for $\mathbf{p} \in \mathrm{dom}(F_\tau^*) = \Delta$. Since F_τ is closed and convex, we also have that $F_\tau = F_\tau^{**}$ (Borwein & Lewis, 2000, Section 4.2); hence

$$F_\tau(\mathbf{q}) = \sup_{\mathbf{p} \in \Delta} \Big\{ \mathbf{q} \cdot \mathbf{p} - F_\tau^*(\mathbf{p}) \Big\}. \tag{33}$$

□

Lemma 9. *For any two vectors $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$,*

$$F_\tau(\mathbf{q}^{(1)}) - F_\tau(\mathbf{q}^{(2)}) \leq \max_a \Big\{ q_a^{(1)} - q_a^{(2)} \Big\}. \tag{34}$$

Proof. Observe that by Lemma 8

$$\begin{aligned}
F_\tau(\mathbf{q}^{(1)}) - F_\tau(\mathbf{q}^{(2)}) &= \sup_{\mathbf{p}^{(1)} \in \Delta} \Big\{ \mathbf{q}^{(1)} \cdot \mathbf{p}^{(1)} - F_\tau^*(\mathbf{p}^{(1)}) \Big\} - \sup_{\mathbf{p}^{(2)} \in \Delta} \Big\{ \mathbf{q}^{(2)} \cdot \mathbf{p}^{(2)} - F_\tau^*(\mathbf{p}^{(2)}) \Big\} && (35) \\
&= \sup_{\mathbf{p}^{(1)} \in \Delta} \Big\{ \inf_{\mathbf{p}^{(2)} \in \Delta} \Big\{ \mathbf{q}^{(1)} \cdot \mathbf{p}^{(1)} - \mathbf{q}^{(2)} \cdot \mathbf{p}^{(2)} - (F_\tau^*(\mathbf{p}^{(1)}) - F_\tau^*(\mathbf{p}^{(2)})) \Big\} \Big\} && (36) \\
&\leq \sup_{\mathbf{p}^{(1)} \in \Delta} \Big\{ \mathbf{p}^{(1)} \cdot (\mathbf{q}^{(1)} - \mathbf{q}^{(2)}) \Big\} \quad \text{by choosing } \mathbf{p}^{(2)} = \mathbf{p}^{(1)} && (37) \\
&\leq \max_a \Big\{ q_a^{(1)} - q_a^{(2)} \Big\}. && (38)
\end{aligned}$$

□

Corollary 10. *F_τ is an ∞-norm contraction; that is, for any two vectors $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$,*

$$\Big| F_\tau(\mathbf{q}^{(1)}) - F_\tau(\mathbf{q}^{(2)}) \Big| \leq \|\mathbf{q}^{(1)} - \mathbf{q}^{(2)}\|_\infty \tag{39}$$

Proof. Immediate from Lemma 9. □

A.2. Preliminaries

Although some of the results in the main body of the paper are expressed in terms of finite horizon problems, we will prove that all the desired properties hold for the more general *infinite horizon* case; the application to the finite horizon case is then immediate. We continue to assume deterministic dynamics, that the action space is finite, and that the state space is discrete.

To make the proofs clearer, we introduce some additional definitions and slightly alter the notation. Since the state s' reached after taking action a in state s is uniquely specified in a deterministic environment, we will denote it by $s' = [s, a]$.

For any policy π, define the entropy regularized expected return by

$$\tilde{V}^\pi(s_k) = O_{\text{ENT}}(s_k, \pi) = \mathbb{E}_{\pi(a_k s_{k+1:\infty})}\left[\sum_{i=0}^{\infty} \gamma^i \big(r(s_{k+i}, a_{k+i}) - \tau \log \pi(a_{k+i}|s_{k+i})\big)\right] \tag{40}$$

We will find it convenient to also work with the on-policy Bellman operator defined by

$$(\mathcal{B}^\pi V)(s) = \pi(-|s) \cdot (\mathbf{q}_s - \tau \log \pi(-|s)), \quad \text{where} \tag{41}$$

$$q_{s,a} = r(s, a) + \gamma V([s, a]) \tag{42}$$

for each state s and action a, where $\mathbf{q}_s$ denotes a vector values over choices of a for a given s, and $\pi(-, s)$ denotes the vector of conditional action probabilities specified by π at state s.

Lemma 11. *For any policy π and state s_k, $\tilde{V}^\pi(s_k)$ satisfies the recurrence*

$$\tilde{V}^\pi(s_k) = \mathbb{E}_{\pi(a_k|s_k)}\Big[r(s_k, a_k) + \gamma \tilde{V}^\pi([s_k, a_k]) - \tau \log \pi(a_k|s_k)\Big] \tag{43}$$

$$= \pi(-|s_k) \cdot \big(\tilde{\mathbf{q}}^\pi_{s_k} - \tau \log \pi(-|s_k)\big) \quad \textit{where} \quad \tilde{\mathbf{q}}^\pi_{s,a} = r(s, a) + \gamma \tilde{V}^\pi([s, a]) \tag{44}$$

$$= (\mathcal{B}^\pi \tilde{V}^\pi)(s_k). \tag{45}$$

Moreover, $\mathcal{B}^\pi$ is a contraction mapping.

Proof. By the definition of $\tilde{V}^\pi(s_k)$ in (40) we have

$$\tilde{V}^\pi(s_k) = \mathbb{E}_{\pi(a_k s_{k+1:\infty})}\Bigg[r(s_k, a_k) - \tau \log \pi(a_k|s_k) + \gamma \sum_{j=0}^{\infty} \gamma^j \big(r(s_{k+1+j}, a_{k+1+j}) - \tau \log \pi(a_{k+1+j}|s_{k+1+j})\big)\Bigg] \tag{46}$$

$$= \mathbb{E}_{\pi(a_k|s_k)}\Bigg[r(s_k, a_k) - \tau \log \pi(a_k|s_k) + \gamma \mathbb{E}_{\pi(a_{k+1} s_{k+2:\infty})}\left[\sum_{j=0}^{\infty} \gamma^j \big(r(s_{k+1+j}, a_{k+1+j}) - \tau \log \pi(a_{k+1+j}|s_{k+1+j})\big) \,\middle|\, s_{k+1} = [s_k, a_k]\right]\Bigg] \tag{47}$$

$$= \mathbb{E}_{\pi(a_k|s_k)}\Big[r(s_k, a_k) - \tau \log \pi(a_k|s_k) + \gamma \tilde{V}^\pi([s_k, a_k])\Big] \tag{48}$$

$$= \pi(-|s_k) \cdot \big(\tilde{\mathbf{q}}^\pi_{s_k} - \tau \log \pi(-|s_k)\big) \tag{49}$$

$$= (\mathcal{B}^\pi \tilde{V}^\pi)(s_k). \tag{50}$$

The fact that $\mathcal{B}^\pi$ is a contraction mapping follows directly from standard arguments about the on-policy Bellman operator (Tsitsiklis & Van Roy, 1997). □

Note that this lemma shows $\tilde{V}^\pi$ is a fixed point of the corresponding on-policy Bellman operator $\mathcal{B}^\pi$.

Lemma 12. *For any π, the on-policy Bellman operator is monotonic: if $V^{(1)} \geq V^{(2)}$ then $\mathcal{B}^\pi V^{(1)} \geq \mathcal{B}^\pi V^{(2)}$.*

Proof. Assume $V^{(1)} \geq V^{(2)}$ and note that for any state s

$$(\mathcal{B}^\pi V^{(2)})(s) - (\mathcal{B}^\pi V^{(1)})(s) = \gamma\, \pi(-|s) \cdot \left(V^{(2)}([s,a]) - V^{(1)}([s,a]) \right) \tag{51}$$

$$\leq 0 \quad \text{since it was assumed that } V^{(2)} \leq V^{(1)}. \tag{52}$$

□

A.3. Proof of main optimality claims

Define the optimal value function by

$$V^\dagger(s) = \max_\pi O_{\text{ENT}}(s,\pi) = \max_\pi \tilde{V}^\pi(s) \text{ for all } s. \tag{53}$$

For $\tau > 0$, define the softmax Bellman operator $\mathcal{B}^*$ by

$$(\mathcal{B}^* V)(s) = \tau \log \sum_a \exp((r(s,a) + \gamma V([s,a]))/\tau) \tag{54}$$

$$= F_\tau(\mathbf{q}_s) \quad \text{where} \quad q_{s,a} = r(s,a) + \gamma V([s,a]) \quad \text{for all } a. \tag{55}$$

Lemma 13. *For $\gamma < 1$, the fixed point of the softmax Bellman operator, $V^* = \mathcal{B}^* V^*$, exists and is unique.*

Proof. First observe that the softmax Bellman operator is a contraction in the infinity norm. That is, consider two value functions, $V^{(1)}$ and $V^{(2)}$, and define $\|V\|_\infty = \max_{s \in S} |V(s)|$. We then have

$$\left\| \mathcal{B}^* V^{(1)} - \mathcal{B}^* V^{(2)} \right\|_\infty = \max_s \left| (\mathcal{B}^* V^{(1)})(s) - (\mathcal{B}^* V^{(2)})(s) \right| \tag{56}$$

$$= \max_s \left| F_\tau(\mathbf{q}_s^{(1)}) - F_\tau(\mathbf{q}_s^{(2)}) \right| \tag{57}$$

$$\leq \max_s \max_a \left| q_{s,a}^{(1)} - q_{s,a}^{(2)} \right| \qquad \text{by Corollary 10} \tag{58}$$

$$= \gamma \max_s \max_a \left| V^{(1)}([s,a]) - V^{(2)}([s,a]) \right| \tag{59}$$

$$\leq \gamma \left\| V^{(1)} - V^{(2)} \right\|_\infty < \left\| V^{(1)} - V^{(2)} \right\|_\infty \text{ if } \gamma < 1. \tag{60}$$

The existence and uniqueness of V^* then follows from the contraction map fixed-point theorem (Bertsekas, 1995). □

Lemma 14. *For any π, if $V \geq \mathcal{B}^* V$ then $V \geq (\mathcal{B}^\pi)^k V$ for all k.*

Proof. Observe for any s that the assumption implies

$$V(s) \geq (\mathcal{B}^* V)(s) \tag{61}$$

$$= \max_{\tilde{\pi}(-|s) \in \Delta} \sum_a \tilde{\pi}(a|s)(r(s,a) + \gamma V([s,a]) - \tau \log \tilde{\pi}(a|s)) \tag{62}$$

$$\geq \sum_a \pi(a|s)(r(s,a) + \gamma V([s,a]) - \tau \log \pi(a|s)) \tag{63}$$

$$= (\mathcal{B}^\pi V)(s). \tag{64}$$

The result then follows by the monotonicity of $\mathcal{B}^\pi$ (Lemma 12). □

Corollary 15. *For any π, if $V \geq \mathcal{B}^* V$ then $V \geq \tilde{V}^\pi$.*

Proof. Consider an arbitrary state s. From Lemma 11 we know that $\tilde{V}^\pi$ is a fixed point of $\mathcal{B}^\pi$. Since $\mathcal{B}^\pi$ is a contraction map, we must have $\limsup_{k \to \infty} \left((\mathcal{B}^\pi)^k V \right)(s) = \tilde{V}^\pi(s)$. But from the assumption on V, we also have $V(s) \geq \left((\mathcal{B}^\pi)^k V \right)(s)$ for all k, by Lemma 14; hence $V(s) \geq \tilde{V}^\pi(s)$. □

Next, given the existence of V^*, we define a specific policy π^* as follows

$$\pi^*(-|s) = \mathbf{f}_\tau(\mathbf{q}^*_s), \quad \text{where} \tag{65}$$

$$q^*_{s,a} = r(s,a) + \gamma V^*([s,a]). \tag{66}$$

Note that we are simply defining π^* at this stage and have not as yet proved it has any particular properties; but we will see shortly that it is, in fact, an optimal policy.

Lemma 16. *$V^* = \tilde{V}^{\pi^*}$; that is, for π^* defined in (65), V^* gives its entropy regularized expected return from any state.*

Proof. We establish the claim by showing $\mathcal{B}^*\tilde{V}^{\pi^*} = \tilde{V}^{\pi^*}$. In particular, for an arbitrary state s consider

$$(\mathcal{B}^*\tilde{V}^{\pi^*})(s) = F_\tau(\tilde{\mathbf{q}}^{\pi^*}_s) \qquad \text{by (55)} \tag{67}$$

$$= \pi^*(-|s) \cdot (\tilde{\mathbf{q}}^{\pi^*}_s - \tau \log \pi^*(-|s)) \qquad \text{by Lemma 5} \tag{68}$$

$$= \tilde{V}^{\pi^*}(s) \qquad \text{by Lemma 11.} \tag{69}$$

□

Theorem 17. *The fixed point of the softmax Bellman operator is the optimal value function: $V^* = V^\dagger$.*

Proof. Since $V^* \geq \mathcal{B}^*V^*$ (in fact, $V^* = \mathcal{B}^*V^*$) we have $V^* \geq \tilde{V}^\pi$ for all π by Corollary 15, hence $V^* \geq V^\dagger$. Next observe that by Lemma 16 we have $V^\dagger \geq \tilde{V}^{\pi^*} = V^*$. Finally, by Lemma 13, we know that the fixed point $V^* = \mathcal{B}^*V^*$ is unique, hence $V^\dagger = V^*$.

□

Corollary 18 (Optimality Implies Consistency)**.** *The optimal state value function V^* and optimal policy π^* satisfy $V^*(s) = r(s,a) + \gamma V^*([s,a]) - \tau \log \pi^*(a|s)$ for every state s and action a.*

Proof. First note that

$$q^*_{s,a} = r(s,a) + \gamma V^*([s,a]) \qquad \text{by (66)} \tag{70}$$

$$= r(s,a) + \gamma V^{\pi^*}([s,a]) \qquad \text{by Lemma 16} \tag{71}$$

$$= q^{\pi^*}_{s,a} \qquad \text{by (42).} \tag{72}$$

Then observe that for any state s,

$$V^*(s) = F_\tau(\mathbf{q}^*_s) \qquad \text{by (55)} \tag{73}$$

$$= F_\tau(\mathbf{q}^{\pi^*}_s) \qquad \text{from above} \tag{74}$$

$$= \pi^*(-|s) \cdot (\mathbf{q}^{\pi^*}_s - \tau \log \pi^*(-|s)) \qquad \text{by Lemma 5} \tag{75}$$

$$= q^{\pi^*}_{s,a} - \tau \log \pi^*(a|s) \text{ for all } a \qquad \text{by Corollary 6} \tag{76}$$

$$= q^*_{s,a} - \tau \log \pi^*(a|s) \qquad \text{from above.} \tag{77}$$

□

Corollary 19 (Consistency Implies Optimality)**.** *If V and π satisfy $V(s) = r(s,a) + \gamma V([s,a]) - \tau \log \pi(a|s)$ for all s and a, then $V = V^*$ and $\pi = \pi^*$.*

Proof. We will show that satisfying the constraint for every s and a implies $\mathcal{B}^*V = V$; it will then immediately follow that $V = V^*$ and $\pi = \pi^*$ by Lemma 13. Let $q_{s,a} = r(s,a) + \gamma V([s,a])$. Consider an arbitrary state s, and observe that

$$(\mathcal{B}^*V)(s) = F_\tau(\mathbf{q}_s) \qquad \text{by (55)} \tag{78}$$

$$= \max_{\mathbf{q} \in \Delta} \Big\{ \boldsymbol{\pi} \cdot (\mathbf{q}_s - \tau \log \boldsymbol{\pi}) \Big\} \qquad \text{by Lemma 5} \tag{79}$$

$$= q_{s,a} - \tau \log \pi(a|s) \text{ for all } a \qquad \text{by Corollary 7, since } V \text{ and } \pi \text{ are consistent} \tag{80}$$

$$= V(s) \qquad \text{by Corollary 7.} \tag{81}$$

□

A.4. Proof of Theorem 1 from Main Text

Proof. Consider the policy π^* defined in (65). From Corollary 16 we know that $\tilde{V}^{\pi^*} = V^*$ and from Theorem 17 we know $V^* = V^\dagger$, hence $\tilde{V}^{\pi^*} = V^\dagger$; that is, π^* is the optimizer of $O_{\text{ENT}}(s, \pi)$ for any state s (including s_0). Therefore, this must be the same π^* as considered in the premise of Theorem 1. The assertion (16) in Theorem 1 then follows directly from Corollary 18. □

A.5. Proof of Corollary 2 from Main Text

Proof. From (13) and (4) we have that $V^*(s) = F_\tau(\mathbf{q}_s^*)$ where $q^*_{s,a}$ matches the definition given in (66). We have already established in the proof of Theorem 1 that the optimal policy π^* satisfies the definition given in (65). The claim then follows immediately. □

A.6. Proof of Corollary 3 from Main Text

Proof. We prove the claim

$$-V^*(s_1) + \gamma^{t-1} V^*(s_t) + R(s_{1:t}) - \tau G(s_{1:t}, \pi^*) = 0 \tag{82}$$

for all $t \geq 1$ by induction on t. For the base case, consider $t = 1$ and observe that $R^\gamma(s_{1:1}) = 0$ and $G^\gamma(s_{1:1}, \pi^*) = 0$, hence (82) reduces to $-V^*(s_1) + V^*(s_1) = 0$.

For the *induction hypothesis* (IH), assume

$$-V^*(s_1) + \gamma^{t-2} V^*(s_{t-1}) + R(s_{1:t-1}) - \tau G(s_{1:t-1}, \pi^*) = 0. \tag{83}$$

Then consider the left hand side of (82):

$$\begin{aligned}
&-V^*(s_1) + \gamma^{t-1} V^*(s_t) + R(s_{1:t}) - \tau G(s_{1:t}, \pi^*) && && (84)\\
&\quad = -V^*(s_1) + \gamma^{t-2} V^*(s_{t-1}) + R(s_{1:t-1}) - \tau G(s_{1:t-1}, \pi^*) && && (85)\\
&\qquad + \gamma^{t-1} V^*(s_t) - \gamma^{t-2} V^*(s_{t-1}) + \gamma^{t-2} r(s_{t-1}, a_{t-1}) - \tau \gamma^{t-2} \log \pi^*(a_{t-1}|s_{t-1}) && && (86)\\
&\quad = \gamma^{t-2} \Big(\gamma V^*(s_t) - V^*(s_{t-1}) + r(s_{t-1}, a_{t-1}) - \tau \log \pi^*(a_{t-1}|s_{t-1}) \Big) && \text{by the IH} && (87)\\
&\quad = 0; && && (88)
\end{aligned}$$

where the last step follows because V^* and π^* satisfy the consistency property:

$$V^*(s_{t-1}) = r(s_{t-1}, a_{t-1}) + \gamma V^*(s_t) - \tau \log \pi^*(a_{t-1}|s_{t-1}) \tag{89}$$

for all s_{t-1} and a_{t-1}, such that $s_t = [s_{t-1}, a_{t-1}]$. □

A.7. Proof of Theorem 4 from Main Text

Proof. Consider a policy π_θ and value function V_ϕ that satisfy the consistency property $V_\phi(s) = r(s, a) + \gamma V_\phi([s, a]) - \tau \log \pi_\theta(a|s)$ for all s and a, where $s' = [s, a]$. Then by Corollary 19, we must have $V_\phi = V^*$ and $\pi_\theta = \pi^*$. □