CONVERGENCE PROOF FOR ACTOR-CRITIC METHODS APPLIED TO PPO AND RUDDER

Anonymous authors
Paper under double-blind review

ABSTRACT

We prove under commonly used assumptions the convergence of actor-critic reinforcement learning algorithms, which simultaneously learn a policy function, the actor, and a value function, the critic. Both functions can be deep neural networks of arbitrary complexity. Our framework allows showing convergence of the well known Proximal Policy Optimization (PPO) and of the recently introduced RUDDER. For the convergence proof we employ recently introduced techniques from the two time-scale stochastic approximation theory. Our results are valid for actor-critic methods that use episodic samples and that have a policy that becomes more greedy during learning. Previous convergence proofs assume linear function approximation, cannot treat episodic examples, or do not consider that policies become greedy. The latter is relevant since optimal policies are typically deterministic.

1 INTRODUCTION

In reinforcement learning, popular methods like Proximal Policy Optimization (PPO) (Schulman et al., 2018) lack convergence proofs. Convergence proofs for these methods are challenging, since they use deep neural networks, episodes as samples, policies that become greedy, and previous policies for trust region methods. For Q-learning, convergence to an optimal policy has been proven in Watkins & Dayan (1992); Bertsekas & Tsitsiklis (1996) as well as for TD(λ) in Dayan (1992). Convergence of SARSA to an optimal policy has been established for policies that become greedy, like "greedy in the limit with infinite exploration" (GLIE) or "restricted rank-based randomized" (RRR) (Singh et al., 2000). Policy gradient methods converge to a local optimum, since the "policy gradient theorem" (Sutton & Barto, 2018, Chapter 13.2) shows that they form a stochastic gradient of the objective. Stochastic gradients converge according to the stochastic approximation theory to an optimum (Robbins & Monro, 1951; Kushner & Clark, 1978; Tsitsiklis, 1994; Borkar & Meyn, 2000; Kushner & Yin, 2003; Borkar, 2008; Bhatnagar et al., 2013). Temporal difference (TD) convergences to a local optimum with smooth function approximation like by neural networks (Maei et al., 2009). Also Deep Q-Networks (DQNs) (Mnih et al., 2013; 2015) use a single neural network, therefore can be shown to converge, as done in Fan et al. (2020). However it is assumed that every training set of reward-state transitions is drawn iid and that a global minimum of the Q-function on the training set is provided.

We prove the convergence of general actor-critic reinforcement learning algorithms (Sutton & Barto, 2018, Chapter 13.5). Recently, actor-critic methods have had a considerable success, e.g. at defeating humans in the game Dota 2 (OpenAI et al., 2019) and in mastering the game of Starcraft II (Vinyals et al., 2019). Actor-critic algorithms simultaneously learn a policy function, the actor, and a critic function that estimates values, action-values, advantages, or redistributed rewards. The critic is responsible for credit assignment, that is, which action or state-action pair was responsible for receiving a reward. Using this credit assignment, a policy function is updated to increase the return. Actor-critic algorithms are typically policy gradient methods, but can also be reward redistribution methods like RUDDER (Arjona-Medina et al., 2019) or "backpropagation through a model" (Munro, 1987; Robinson, 1989; Robinson & Fallside, 1989; Bakker, 2007). Actor-critic algorithms have been only proven to converge for simple settings like for the neural networks that are linear (Konda & Tsitsiklis, 2000; Konda & Borkar, 1999; Xu et al., 2019; Yang et al., 2019; Liu et al., 2019). In contrast to these convergence proofs, in our setting both functions can be deep neural networks of arbitrary complexity, though they should not share weights.

The main contribution of this paper is to provide a convergence proof for general actor-critic reinforcement learning algorithms. We apply this convergence proof to two concrete actor-critic methods. First, we establish convergence of a practical variant of Proximal Policy Optimization (PPO) (Schulman et al., 2018). PPO is an actor-critic on-policy gradient method with trust region penalties to ensure a small policy gap (Schulman et al., 2015). Secondly, we prove convergence of the recently introduced RUDDER (Arjona-Medina et al., 2019). RUDDER targets the problem of sparse and delayed rewards by reward redistribution which directly and efficiently assigns reward to relevant state-action pairs. Thus, RUDDER dramatically speeds up learning for sparse and delayed rewards. In RUDDER, the critic is the reward redistributing network, which is typically an LSTM.

The main proof techniques are recent developments from the two time-scale stochastic approximation theory (Borkar, 2008). The recent addition to the theory is the introduction of controlled Markov processes (Karmakar & Bhatnagar, 2017), which can treat policies that become more greedy and trust region methods that use previous policies. The two time-scale stochastic approximation framework has been applied previously to show convergence of actor-critic algorithms (Konda & Tsitsiklis, 2000; Konda & Borkar, 1999) and, more recently, of Linear Quadratic Regulator (LQR) problems (Xu et al., 2019) and off-policy TD learning (Yang et al., 2019). However, only tabular cases or linear function approximations have been considered. In a recent work, convergence was shown for variants of PPO and Trust Region Policy Optimization (TRPO) equipped with neural networks (Liu et al., 2019). However, again the neural networks were only linear, the policy was energy-based, and the Kullback-Leibler term of the trust-region method was modified.

We aim at generalizing these proofs to learning settings which use deep neural networks, use episodes, use policies that become greedy, and use trust region methods. Therefore, the idea of stationary distributions on state-action pairs does not apply (Konda & Tsitsiklis, 2000; Konda & Borkar, 1999) and we have to enrich the framework by a controlled Markov process which describes how the policy becomes more greedy and how to use previous policies. While we are developing a framework to ensure convergence, it does not imply convergence to an optimal policy. Such proofs are in general difficult for methods that use deep neural networks, since locally stable attractors may not correspond to optimal policies (Mazumdar et al., 2019; Jin et al., 2019; Lin et al., 2019). However, convergence to a locally optimal policy can be proven for linear approximation to Q-values (Sutton et al., 2000; Konda & Tsitsiklis, 2003). Our main contributions to the convergence proof are, that we:

- use a Markov control in the two time-scale stochastic approximation framework,
- use episodes as samples instead of transitions,
- allow policies to become greedy,
- allow objectives that use previous policies (trust region methods).

In the next section, the main theorem is provided, which shows local convergence of actor-critic methods. Next, we formulate the results for PPO and RUDDER as corollaries. The third section gives a roadmap for the corresponding proofs, thereby introducing the precise framework and the results from stochastic approximation theory (Borkar, 2008; Karmakar & Bhatnagar, 2017). Finally, we discuss the technical assumptions and details for the proofs.

2 The Main Results

2.1 Abstract setting and Main Theorem

Preliminaries We consider a finite MDP defined by the 4-tuple $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ (we assume a discount factor $\gamma = 1$) where the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are sets of finite states s and actions a and $\mathcal{R}$ the set of rewards r which are bounded. Let us denote by $|\mathcal{A}|$ and $|\mathcal{S}|$ the corresponding cardinalities and $K_R > 0$ an upper bound on the absolute values of the rewards. For a given time step t, the random variables for state, action, and reward are S_t, A_t and $R_{t+1} = R(S_t, A_t)$, respectively. Furthermore, $\mathcal{P}$ has transition-reward distributions $p(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a)$. By π we denote an associated Markov-policy. The (undiscounted) return of a sequence of length T at time t is $G_t = \sum_{k=0}^{T-t} R_{t+k+1}$. As usual, the action-value function for a given policy π is $q^\pi(s, a) = \mathrm{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$. The goal is to find the optimal policy $\pi^* = \operatorname{argmax}_\pi \mathrm{E}_\pi[G_0]$. We assume that the states s are time-aware (time t can be extracted from each state) in order to guarantee stationary optimal policies.

The abstract actor-critic setting is assumed to have two loss functions: L_h for the policy and L_g for the critic. Additionally we have the following building blocks:

- We consider two classes of parameters, denoted by $\boldsymbol{\omega} \in \mathbb{R}^m$ and $\boldsymbol{\theta} \in \mathbb{R}^k$. Moreover, $\boldsymbol{z}$ denotes an additional controlled Markov process with values in a compact metric space that may allow e.g. to force the policy to get more greedy and for treating trust region methods which rely on previous policies (it may be used for other purposes as well, e.g. Markovian sampling). $\boldsymbol{z}$ will be defined in a similar abstract way as done in Karmakar & Bhatnagar (2017) to make the setting as general as possible. We defer the technical details to Section 3.1.
- The first loss $\mathrm{L}_h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$ is minimized with respect to $\boldsymbol{\theta}$ in order to find an optimal policy. This is achieved by updating a sufficiently smooth policy $\pi(\boldsymbol{\theta}, \boldsymbol{z})$, that can be controlled by $\boldsymbol{z}$. We will discuss in Section 3.2, how π can be constructed in specific situations. Next we consider two optional possibilities, how $\mathrm{L}_h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$ may be defined: it may equal the expectation (i) $\mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}, \boldsymbol{z})} [\phi(\tau, \boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})]$ or (ii) $\mathrm{E}_{\tau \sim \breve{\pi}} [\phi(\pi(.; \boldsymbol{\theta}, \boldsymbol{z}), \tau, \boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})]$ where the expectations are taken over whole episodes $\tau = (s_0, a_0, \ldots, s_T, a_T)$ (sequences) that are generated via (i) $\pi(\boldsymbol{\theta}, \boldsymbol{z})$ or (ii) a behavioral policy $\breve{\pi}$, respectively. It will be clear from the context, which of these two possibilities we are using. The function ϕ can be interpreted as a per sample loss for a sufficiently smooth neural network, that tries to find the optimal policy, evaluated only on a single trajectory τ. The detailed smoothness assumptions on L_h that need to be imposed are discussed in Section 3.2. The gradient of $\mathrm{L}_h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$ will be denoted by $h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$.
- The second loss is given by $\mathrm{L}_g(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z}) = \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}, \boldsymbol{z})} [\Phi(g(\tau; \boldsymbol{\omega}, \boldsymbol{z}), \tau, \boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})]$ and is minimized with respect to $\boldsymbol{\omega}$ in order to find an optimal critic function $g(\tau; \boldsymbol{\omega}, \boldsymbol{z})$. The functions g and Φ should again be sufficiently smooth, such that $\mathrm{L}_g(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$ satisfies (L1)–(L3) from Section 3.2. Φ can be seen as the per sample loss for the critic g. The gradient of L_g will be denoted by f.
- Since the expectations cannot be computed analytically, we do not have the exact gradients $h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$ and $f(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$. Therefore, the expectations are approximated by sampling sequences τ and computing the average gradient on the sampled sequences. In our case, the stochastic approximations $\hat{h}$ and $\hat{f}$ of the gradients h and f respectively, are created by randomly inserting only one sample trajectory τ, i.e. we are dealing with online stochastic gradients. A formal description of the sampling process can be found in Section 3.2. Our losses are then minimized using online stochastic gradient descent (SGD) with learning rates $a(n)$ and $b(n)$, where the integer $n \geq 0$ denotes the timestep of our iteration.

For a given n, let us now state the discussed building blocks in a more compact and formal way:

$$\begin{aligned} \mathrm{L}_h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) &= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} [\phi(\tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] \,, \\ h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) &= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} [\nabla_{\theta_n} \log \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)\, \phi(\tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) + \nabla_{\theta_n} \phi(\tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] \,, \end{aligned} \tag{1}$$

where the first possibility for the policy loss L_h and its gradient h is listed. h is computed by the Policy Gradient Theorem, which can be found e.g. in Sutton & Barto (2018, Chapter 13.2). Next we discuss the expressions for the second possibility for L_h (when sampling via a behavioral policy $\breve{\pi}$):

$$\mathrm{L}_h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) = \mathrm{E}_{\tau \sim \breve{\pi}} [\phi(\pi(.; \boldsymbol{\theta}_n, \boldsymbol{z}_n), \tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] \,, \tag{2}$$

$$h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) = \mathrm{E}_{\tau \sim \breve{\pi}} [\nabla_{\theta_n} \phi(\pi(.; \boldsymbol{\theta}_n, \boldsymbol{z}_n), \tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] \,. \tag{3}$$

The expressions for our second loss L_g and its gradient f are as follows:

$$\begin{aligned} \mathrm{L}_g(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) &= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} [\Phi(g(\tau; \boldsymbol{\omega}_n, \boldsymbol{z}_n), \tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] \,, \\ f(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) &= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} [\nabla_{\omega_n} \Phi(g(\tau; \boldsymbol{\omega}_n, \boldsymbol{z}_n), \tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] \,, \end{aligned} \tag{4}$$

and finally, the iterative algorithm that optimizes the losses by online SGD, is given by:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - a(n)\, \hat{h}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \,, \quad \boldsymbol{\omega}_{n+1} = \boldsymbol{\omega}_n - b(n)\, \hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \,. \tag{5}$$

Main Theorem. Our main result will guarantee local convergence for (5). To this end we fix a starting point $(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)$ and determine an associated neighborhood $V_0 \times U_0$ which can be constructed by the loss assumptions (L1)–(L3) given in Section 3.2. The iterates (5) will always stay in $V_0 \times U_0$

by these assumptions. Furthermore, let us denote the loss functions that result after considering the "limit" of the control sequence $\boldsymbol{z}_n \to \boldsymbol{z}$ by $L_h(\boldsymbol{\theta}, \boldsymbol{\omega})$ (and similarly $L_g(\boldsymbol{\theta}, \boldsymbol{\omega})$). Again we refer to Section 3.1 for a precise account on this informal description. Moreover, we denote a local minimum of $L_g(\boldsymbol{\theta}, \cdot)$ by $\boldsymbol{\lambda}(\boldsymbol{\theta})$, whereas $\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0))$ should indicate a local minimum of $L_h(\cdot, \boldsymbol{\lambda}(\cdot))$ in $V_0 \times U_0$. Also here we refer to Section 3.2 for a precise discussion. We can now state our main theorem:

Theorem 1. *Fix a starting point $(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)$. Determine the associated neighborhood $V_0 \times U_0$ as in Section 3.2. Assume learning rates like (A4) for the time-scales $a(n)$ and $b(n)$ mentioned in Section 3.1. Also take the loss assumptions in Section 3.2 for granted.*
Then Eq. (5) converges to a local minimum $(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0), \boldsymbol{\lambda}(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)))$ of the associated losses (i) Eq. (1) or (ii) Eq. (2) and Eq. (3): $(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n) \to (\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0), \boldsymbol{\lambda}(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)))$ a.s. as $n \to \infty$.

2.2 Convergence Proof for PPO.

The main theorem is applied to prove convergence of PPO. Our PPO variant uses deep neural networks, softmax outputs for the policy network, regularization, trust region, or exploration terms. Regularization can be entropy, weight decay, or a trust region penalty like the Kullback-Leibler divergence. All functions are assumed to be sufficiently smooth, i.e. at least three times continuously differentiable and bounded wrt. the parameters. The losses should satisfy (L1)–(L3) from Section 3.2. The PPO alorithm aims at minimizing the following losses:

$$\mathrm{L}_h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[-\, G_0 \;+\; (z_2)_n \, \rho(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] , \tag{6}$$

$$\mathrm{L}_g^{\mathrm{TD}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[\frac{1}{2} \sum_{t=0}^{T} \left(\delta^{\mathrm{TD}}(t) \right)^2 \right] , \tag{7}$$

$$\mathrm{L}_g^{\mathrm{MC}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[\frac{1}{2} \sum_{t=0}^{T} \left(G_t \;-\; \hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n) \right)^2 \right] , \tag{8}$$

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;-\; a(n) \, \hat{h}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \, , \quad \boldsymbol{\omega}_{n+1} \;=\; \boldsymbol{\omega}_n \;-\; b(n) \, \hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \, . \tag{9}$$

Let us now briefly describe the terms in Eq. (6)–(9):

- $\hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega})$ is a function that approximates the Q-value $q^{\pi}(s_t, a_t)$.
- $\delta^{\mathrm{TD}}(t) = R(s_t, a_t) + \hat{q}^{\pi}(s_{t+1}, a_{t+1}; \boldsymbol{\omega}_{n-1}) - \hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n)$ is the temporal difference error.
- The exact gradients $h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ (from the Policy Gradient Theorem assuming causality and subtracting a baseline (Sutton & Barto, 2018, Chapter 13.2)), $f^{\mathrm{TD}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ and $f^{\mathrm{MC}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ of the respective losses can be found in Sections A.1 and A.3 in the appendix.
- $\boldsymbol{z}_n \;=\; ((z_1)_n, (z_2)_n, (z_1)_{n-1}, (z_2)_{n-1}, \boldsymbol{\theta}_{n-1}, \boldsymbol{\omega}_{n-1})$ denotes an additional controlled Markov process with values in compact sets. The controlled Markov process is essential to define the trust region term of PPO which uses previous values of $\boldsymbol{\theta}$ and z_1. Here, $z_1 \in [1, \beta]$ increases from 1 to $\beta > 1$ and $z_2 \in [0, (z_2)_0]$ decreases from $(z_2)_0 > 1$ to 0. z_1 controls the amount of greediness and z_2 the regularization. Details can be found in Section 3.2.
- $\pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)$ is a softmax policy that depends on $(z_1)_n$ to make it more greedy. We will introduce it precisely in Section 3.2, especially Eq. (15) there. π is learned using $\hat{q}$ and updated in every time-step.
- $\rho(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n)$ includes the trust region term of PPO and may also include regularization terms like weight decay or entropy regularization. For example, $\rho(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \;= \mathrm{KL}_{\epsilon}(\pi(\boldsymbol{\theta}_{n-1}, (z_1)_{n-1}), \pi(\boldsymbol{\theta}_n, (z_1)_n))$, where $\mathrm{KL}_{\epsilon}(\boldsymbol{p}, \boldsymbol{q}) = \mathrm{KL}(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})$ with $\tilde{p}_i = (p_i + \epsilon)/(1 + k\epsilon)$.

The next corollary states that the above described PPO algorithms (TD and MC versions) converge.

Corollary 1 (Convergence PPO)**.** *Fix a starting point $(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)$. Determine the associated neighborhood $V_0 \times U_0$ as in Section 3.2. Assume learning rates like (A4) for the time-scales $a(n)$ and $b(n)$ mentioned in Section 3.1. Also take the loss assumptions in Section 3.2 for granted.*
Using the same notation as in Theorem 1, the PPO algorithm Eq. (9) converges to a local minimum $(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0), \boldsymbol{\lambda}(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)))$ of the associated losses Eq. (6) and either Eq. (7) or Eq. (8): $(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n) \to (\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0), \boldsymbol{\lambda}(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)))$ a.s. as $n \to \infty$.

Proof. We apply Theorem 1 since all its assumption are fulfilled. □

2.3 CONVERGENCE PROOF FOR RUDDER.

The main theorem is applied to prove convergence of RUDDER, which excels for tasks with sparse and delayed rewards (Arjona-Medina et al., 2018). Again, we assume enough smoothness for all functions, i.e. they are at least three times continuously differentiable wrt. the parameters and bounded. The losses should satisfy (L1)–(L3) from Section 3.2. We formulate the RUDDER algorithm as a minimization problem of square losses $\mathrm{L}_h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ and $\mathrm{L}_g(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$:

$$\mathrm{L}_h \;=\; \mathrm{E}_{\tau \sim \breve{\pi}} \left[\frac{1}{2} \sum_{t=0}^{T} \left(R_{t+1}(\tau; \boldsymbol{\omega}_n) - \hat{q}(s_t, a_t; \boldsymbol{\theta}_n) \right)^2 \;+\; (z_2)_n \; \rho_{\boldsymbol{\theta}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] , \tag{10}$$

$$\mathrm{L}_g \;=\; \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[\frac{1}{2} \left(\sum_{t=0}^{T} \tilde{R}_{t+1} \;-\; g(\tau; \boldsymbol{\omega}_n) \right)^2 \;+\; (z_2)_n \; \rho_{\boldsymbol{\omega}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] . \tag{11}$$

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;-\; a(n) \; \hat{h}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \; , \quad \boldsymbol{\omega}_{n+1} \;=\; \boldsymbol{\omega}_n \;-\; b(n) \; \hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \; . \tag{12}$$

Let us now briefly describe the terms in Eq. (10)–(12):

- $\hat{q}(s, a; \boldsymbol{\theta}_n)$ is a function parametrized by $\boldsymbol{\theta}_n$ that approximates the Q-value $q(s, a)$. Note that the policy loss L_h implicitly depends on the policy π via $\hat{q}$.
- The expressions for $h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ and $f(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ can be found in Section A.2 in the appendix.
- $\tilde{R}$ is the original MDP reward.
- $R(\tau; \boldsymbol{\omega}_n)$ is the redistributed reward based on the return decomposition of g with parameter vector $\boldsymbol{\omega}_n$. For a state-action sequence τ the realization of its redistributed reward R is computed from $g(\tau; \boldsymbol{\omega}_n)$ and the realization of return variable $\sum_{t=0}^{T} \tilde{R}_{t+1}$. In practice, g is an LSTM-network.
- $\rho_{\boldsymbol{\theta}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n)$ is a regularization term for learning the Q-value approximation $\hat{q}$.
- $\rho_{\boldsymbol{\omega}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n)$ is a regularization term for learning the reward redistribution function g.
- $\breve{\pi}$ is a behavioral policy that does not depend on the parameters.
- $\boldsymbol{z}_n = ((z_1)_n, (z_2)_n, (z_1)_{n-1}, (z_2)_{n-1}, \boldsymbol{\theta}_{n-1}, \boldsymbol{\omega}_{n-1})$ denotes an additional Markov process, where we use the same construction as in the PPO setting. Details can again be found in Section 3.2.
- $\pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)$ is a softmax policy applied to $(z_1)_n \; \hat{q}$ (see Eq. (15) for a precise introduction). It depends on $(z_1)_n > 1$ which makes it more greedy and $\boldsymbol{\theta}_n$ is updated in every time-step.

The next corollary states that the RUDDER algorithm converges.

Corollary 2 (Convergence RUDDER)**.** *Fix a starting point $(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)$. Determine the associated neighborhood $V_0 \times U_0$ as in Section 3.2. Assume learning rates like (A4) for the time-scales $a(n)$ and $b(n)$ mentioned in Section 3.1. Also take the loss assumptions in Section 3.2 for granted. Using the same notation as in Theorem 1, the RUDDER algorithm Eq. (12) converges to a local minimum $(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0), \boldsymbol{\lambda}(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)))$ of the associated losses Eq. (10) and Eq. (11): $(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n) \;\to\; (\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0), \boldsymbol{\lambda}(\boldsymbol{\theta}^*(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)))$ a.s. as $n \;\to\; \infty$.*

Proof. We apply Theorem 1 since all its assumptions are fulfilled. □

3 ASSUMPTIONS AND PROOF OF THEOREM 1

This section aims at presenting the theoretical framework from Karmakar & Bhatnagar (2017) and Borkar (2008) that we want to apply to prove Theorem 1. We formulate the convergence result Theorem 2 and the assumptions (A1)–(A7) that we need to ensure in order to get there. Then we discuss how it can be applied to our setting.

3.1 THE STOCHASTIC APPROXIMATION THEORY: BORKAR AND KARMAKAR & BHATNAGAR.

For this section we use the formulations in Heusel et al. (2017); Karmakar & Bhatnagar (2017). Stochastic approximation algorithms are iterative procedures to find stationary points (minimum, maximum, saddle point) of functions when only noisy observations are provided. We use two time-scale stochastic approximation algorithms, i.e. two coupled iterations moving at different speeds. Convergence of these interwoven iterates can be ensured by assuming that one step size is considerably smaller than the other. The slower iterate is assumed to be slow enough to allow the fast iterate to converge while simultaneously being perturbed by the slower. The perturbations of the slower should

be small enough to ensure convergence of the faster. The iterates map at time step $n \geq 0$ the fast variable $\boldsymbol{\omega}_n \in \mathbb{R}^k$ and the slow variable $\boldsymbol{\theta}_n \in \mathbb{R}^m$ to their new values:

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;-\; a(n)\,\left(h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;+\; (\boldsymbol{m}_1)_n\right)\,, \tag{13}$$

$$\boldsymbol{\omega}_{n+1} \;=\; \boldsymbol{\omega}_n \;-\; b(n)\,\left(f(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;+\; (\boldsymbol{m}_2)_n\right)\,, \tag{14}$$

where:

- $h(.) \in \mathbb{R}^m$ and $f(.) \in \mathbb{R}^k$ are mappings for Eq. (13) and Eq. (14), respectively.
- $a(n)$ and $b(n)$ are step sizes for Eq. (13) and Eq. (14), respectively.
- $(\boldsymbol{m}_1)_n$ and $(\boldsymbol{m}_2)_n$ are Martingale difference sequences for Eq. (13) and Eq. (14), respectively.
- $\boldsymbol{z}_n$ denotes the common Markov control process for Eq. (13) and Eq. (14).

We assume that all the random variables are defined on a common probability space $(\Omega, \mathfrak{A}, P)$ with associated sigma algebra $\mathfrak{A}$ and probability measure P. Let us continue with an informal summary of the assumptions needed to ensure convergence of (13)–(14). More precise technical details can be found in Section A.4 in the appendix and in Karmakar & Bhatnagar (2017).

(A1) *Assumptions on the controlled Markov processes:* $\boldsymbol{z}_n$ takes values in a compact metric space S. It is controlled by the iterate sequences $\boldsymbol{\theta}_n$ and $\boldsymbol{\omega}_n$ and additionally by a random process $\boldsymbol{a}_n$ taking values in a compact metric space W. The dynamics wrt. n are specified by a transition kernel. Be aware that this control setting can in general be different than the already introduced MDP setting.

(A2) *Assumptions on the update functions:* f, and h are jointly continuous as well as Lipschitz in their first two arguments, and uniformly w.r.t. the third.

(A3) *Assumptions on the additive noise:* For $i = 1, 2$ the $(\boldsymbol{m}_i)_n$ are martingale difference sequences with bounded second moments.

(A4) *Assumptions on the learning rates:* Informally, the sums of the positive $a(n)$ and $b(n)$ diverge, while their squared sums converge. $a(n)$ goes to zero faster than $b(n)$.

(A5) *Assumptions on the transition kernels:* The transition kernels of $\boldsymbol{z}_n$ are continuous.

(A6) *Assumptions on the associated ODEs:* We consider occupation measures which intuitively give for the controlled Markov process the probability or density to observe a particular state-action pair from $S \times W$ for given $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ and a given control. A precise definition of these occupation measures can be found e.g. on page 68 of Borkar (2008) or page 5 in Karmakar & Bhatnagar (2017). We need the following assumptions:

- We assume that there exists only one such ergodic occupation measure for $\boldsymbol{z}_n$ on $S \times W$, denoted by $\Gamma_{\boldsymbol{\theta},\boldsymbol{\omega}}$. A main reason for assuming uniqueness is that it enables us to deal with ODEs instead of differential inclusions. Moreover, set $\tilde{f}(\boldsymbol{\theta}, \boldsymbol{\omega}) \;=\; \int f(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})\, \Gamma_{\boldsymbol{\theta},\boldsymbol{\omega}}(\mathrm{d}\boldsymbol{z}, W)$.
- For $\boldsymbol{\theta} \in \mathbb{R}^m$, the ODE $\dot{\boldsymbol{\omega}}(t) \;=\; \tilde{f}(\boldsymbol{\theta}, \boldsymbol{\omega}(t))$ has a unique asymptotically stable equilibrium $\boldsymbol{\lambda}(\boldsymbol{\theta})$ with attractor set $B_{\boldsymbol{\theta}}$ such that $\boldsymbol{\lambda} : \mathbb{R}^m \to \mathbb{R}^k$ is a Lipschitz map with global Lipschitz constant.
- The Lyapunov function $V(\boldsymbol{\theta}, .)$ associated to $\boldsymbol{\lambda}(\boldsymbol{\theta})$ is continuously differentiable.
- Next define $\tilde{h}(\boldsymbol{\theta}) \;=\; \int h(\boldsymbol{\theta}, \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{z})\, \Gamma_{\boldsymbol{\theta},\boldsymbol{\lambda}(\boldsymbol{\theta})}(\mathrm{d}\boldsymbol{z}, W)$. The ODE $\dot{\boldsymbol{\theta}}(t) \;=\; \tilde{h}(\boldsymbol{\theta}(t))$ has a global attractor set A.
- For all $\boldsymbol{\theta}$, with probability 1, $\boldsymbol{\omega}_n$ for $n \geq 1$ belongs to a compact subset $Q_{\boldsymbol{\theta}}$ of $B_{\boldsymbol{\theta}}$ "eventually".

This assumption is an adapted version of (A6)' of Karmakar & Bhatnagar (2017) to avoid too many technicalities (e.g. Karmakar & Bhatnagar (2017) uses a different control for each iterate).

(A7) *Assumption of bounded iterates:* The iterates $\boldsymbol{\theta}_n$ and $\boldsymbol{\omega}_n$ are uniformly bounded almost surely.

Convergence for Eq. (13)–(14) is given by Theorem 1 in Karmakar & Bhatnagar (2017):

Theorem 2 (Karmakar & Bhatnagar)**.** *Under the assumptions (A1)–(A7), the iterates Eq.* (13) *and Eq.* (14) *converge:* $(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n) \;\to\; \cup_{\boldsymbol{\theta}^* \in A}(\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*))$ *a.s. as* $n \;\to\; \infty$.

3.2 Application to Proof of Main Result

Next we describe how Theorem 2 yields Theorem 1 by discussing the validity of (A1)–(A7). We additionally mention details about their concrete realization in the context of PPO and RUDDER. We conclude by a discussion on how we can allow our policies to become sufficiently greedy over time.

(A1) Controlled Markov process for the abstract setting: For Eq. (1) – (3) we assume to have a controlled process that fulfills the previously discussed requirements for (A1).

(A1) Controlled Markov process for PPO and RUDDER: In our applications to RUDDER and PPO, however, the Markov control will have a much simpler form: $\boldsymbol{z}_n$ mainly consists of real sequences which obey the Markov property. Also we do not have any additional control in these situations. More concretely: $\boldsymbol{z}_n = ((z_1)_n, (z_2)_n, (z_1)_{n-1}, (z_2)_{n-1}, \boldsymbol{\theta}_{n-1}, \boldsymbol{\omega}_{n-1})$ with $(z_1)_n \in [1, \beta]$ for some $\beta > 1$, and $(z_2)_n \in [0, (z_2)_0]$ for some $(z_2)_0 > 1$. $(z_1)_n$ can be defined by $(z_1)_0 = 1$ and $(z_1)_{n+1} = (1 - \frac{1}{\beta})(z_1)_n + 1$. It consists of the partial sums of a geometric series converging to $\beta > 1$. For $(z_2)_n$ we can use any sequence satisfying the Markov Property and converging to zero, e.g. $(z_2)_0 = C$ and $(z_2)_{n+1} = \alpha(z_2)_n$ with $\alpha < 1$ or $(z_2)_{n+1} = \frac{(z_2)_n}{(z_2)_n + \alpha}$ with $1 < \alpha$. $\boldsymbol{z}_n$ then is a time-homogeneous Markov process with unique invariant measure, cf. Section A.6 in the appendix.

Let us now describe the meaning of this process for RUDDER and PPO: The component $(z_1)_n$ is used as a slope parameter for the softmax policy and goes to a large value β to make the policy greedy. The softmax policy is introduced in the following rather abstract way: For a sufficiently smooth function (deep neural network) $\boldsymbol{\psi}(s; \boldsymbol{\theta}) = (\psi^1(s; \boldsymbol{\theta}), \ldots, \psi^{|\mathcal{A}|}(s; \boldsymbol{\theta}))$, a softmax policy $\pi(\boldsymbol{\theta}, z_1)$ is defined as

$$\pi(a^i \mid s; \boldsymbol{\theta}, z_1) \;=\; \frac{\exp(z_1 \; \psi^i(s; \boldsymbol{\theta}))}{\sum_j \exp(z_1 \; \psi^j(s; \boldsymbol{\theta}))} \; . \tag{15}$$

For RUDDER and PPO we use $\boldsymbol{\psi}(s; \boldsymbol{\theta}) = \hat{q}(s; \boldsymbol{\theta}_n)$ with $\hat{q}^i(s; \boldsymbol{\theta}_n)$ approximating $q^\pi(s, a^i)$. The component $(z_2)_n$ is used to weight an additional term in the objective and goes to zero over time. We require $(z_1)_{n-1}$ and $\boldsymbol{\theta}_{n-1}$ for the trust-region term ρ, for which we have to reconstruct the old policy. Further details, especially concerning β, can be found in Section 3.2 and Section A.7 in the appendix.

(A3) Martingale Difference Property and the Probabilistic Setting. Here we describe the sampling process more formally:

- The baseline probability space is given by $\Omega = [0, 1]$, $P = \mu$ and $\mathfrak{A} = \mathfrak{B}([0, 1])$, with μ denoting the Lebesgue measure and $\mathfrak{B}([0, 1])$ the Borel σ-algebra on $[0, 1]$.
- Next we introduce the set of all trajectories obtained by following π as $\tilde{\Omega}_\pi = \{\tau = (s, a)_{0:T} | \tau \text{ is chosen wrt. } \pi, S_0 = s_0, A_0 = a_0\}$. Its power set serves as related σ-algebra $\tilde{\mathfrak{A}}_\pi$.
- We endow $\tilde{\mathfrak{A}}_\pi$ with a probability measure: $\tilde{P}_\pi(\tau) = \prod_{t=1}^T p(s_t \mid s_{t-1}, a_{t-1}) \, \pi(a_t \mid s_t)$, which computes the probability of choosing a sequence τ with starting point (s_0, a_0). $\tilde{\mathfrak{A}}_\pi$ can be ordered according to the magnitude of the values of its events on P_π. We denote this ordering by $\leq$.
- We define $S_\pi : \Omega \to \tilde{\Omega}_\pi$ as $S_\pi : x \mapsto \text{argmax}_{\tau \in \tilde{\Omega}_\pi} \left\{ \sum_{\eta \leq \tau} \tilde{P}_\pi(\eta) \leq x \right\}$. This map is well defined and measurable and it describes how to get one sample from a multinomial distribution with probabilities $\tilde{P}_\pi(\tau)$, where $\tau \in \tilde{\Omega}_\pi$.
- Now we are in the position to describe the sampling process. As mentioned already in the beginning, we use an online update, i.e. we introduce functions $\hat{h}$ and $\hat{f}$, where $\hat{h}$ approximates h by using one sample trajectory instead of the expectation, the same goes for $\hat{f}$. More formally, for f we define $\hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) : [0, 1] \to \mathbb{R}^k$ as $x \mapsto S_{\pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)}(x) = \tau \mapsto \nabla_{\boldsymbol{\omega}_n} \Phi(g(\tau; \boldsymbol{\omega}_n, \boldsymbol{z}_n), \tau, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$.
- Finally we can define the martingale errors as $(\boldsymbol{m}_1)_{n+1} = \hat{h}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) - h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$ and $(\boldsymbol{m}_2)_{n+1} = \hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) - f(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)$.

Further details (regarding the sampling process and the bounds for the second moments) can be found in Sections A.5 and A.6 in the appendix.

(A2) and (A6) Smoothness for h and f and Stability of ODEs via Assumptions on Losses. We make the following assumptions on the loss functions of Eq. (1) (or Eq. (2)) and Eq. (3):

(L1) π, g, Φ and ϕ all have compact support and are at least three times continuously differentiable wrt. their parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$.

(L2) For each fixed $\boldsymbol{\theta}$ all critical points of $L_g(\boldsymbol{\theta}, \boldsymbol{\omega})$ are isolated local minima and there are only finitely many. The local minima $\{\lambda_i(\boldsymbol{\theta})\}_{i=1}^{k(\boldsymbol{\theta})}$ of $L_g(\boldsymbol{\theta}, .)$ can be expressed locally as at least twice continuously differentiable functions with associated domains of definitions $\{V_{\lambda_i(\boldsymbol{\theta})}\}_{i=1}^{k(\boldsymbol{\theta})}$.

(L3) Locally in $V_{\lambda_i(\boldsymbol{\theta})}$, $L_h(\boldsymbol{\theta}, \lambda_i(\boldsymbol{\theta}))$ has only one local minimum.

Some remarks concerning these assumptions are in order:

- *Comment on (L1)*: The parameter space of networks can be assumed to be bounded in practice.
- *Comment on (L2)*: For each starting point $(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)$ we can find a neighborhood $U_{\boldsymbol{\theta}_0}(\boldsymbol{\omega}_0)$ that connects $\boldsymbol{\omega}_0$ with a local minimum $\lambda_i(\boldsymbol{\theta}_0)$ of $L_g(\boldsymbol{\theta}_0, \cdot)$, so that $U_{\boldsymbol{\theta}_0}(\boldsymbol{\omega}_0)$ contains no further critical points, e.g. a small neighborhood around the steepest descent path on the loss surface of $L_g(\boldsymbol{\theta}_0, \cdot)$ starting at $\boldsymbol{\omega}_0$. Next we apply the implicit function theorem (IFT) to find a neighborhood V_0 around $\boldsymbol{\theta}_0$, such that $\lambda_i(\boldsymbol{\theta})$ is twice continuously differentiable there. The IFT can be applied to $f(\boldsymbol{\theta}, \cdot) = \nabla_{\boldsymbol{\omega}} L_g(\boldsymbol{\theta}, \cdot) = 0$, since the associated Hessian is positive definite and thus invertible. It can even be shown that it is twice continuously differentiable, using analytic versions of the IFT.
- *Comment on (L3)*: In a similar vein, for each $\boldsymbol{\theta} \in V_0$ we can construct neighborhoods $U_{\boldsymbol{\theta}}(\boldsymbol{\omega}_0)$ around $\boldsymbol{\omega}_0$ with $\lambda_i(\boldsymbol{\theta})$ as unique associated local minimum (we may have to shrink V_0). Define $\cup_{\boldsymbol{\theta} \in V_0}(\{\boldsymbol{\theta}\} \times U_{\boldsymbol{\theta}}(\boldsymbol{\omega}_0)) = V_0 \times U_0$.
- *Comment on compatibility with global setting:* By using a suitable regularization (e.g. weight decay) for the networks, we can assume that the algorithm Eq. (13) and Eq. (14) always stays in $V_0 \times U_0$. This heuristically justifies that for $(\boldsymbol{\theta}_0, \boldsymbol{\omega}_0)$ we localize to $V_0 \times U_0$.
- *Comment on drawbacks of assumptions:* A completely rigorous justification of this argument would require a more thorough analysis of SGD, which would of course be a very interesting future research direction. For SGD with one time scale, a result in this direction can be found in Metrikopoulos et al. (2020). It would be interesting to extend it to two timescales.
- *Comment on requirements for critical points:* It is also a widely accepted (but yet unproven) conjecture that the probability of ending in a poor local minimum is very small for sufficiently large networks, see e.g. Choromanska et al. (2015); Kawaguchi (2016); Kawaguchi et al. (2017); Kawaguchi & Bengio (2019); Kawaguchi et al. (2019). Thus, we can ensure that Eq. (13) and Eq. (14) really converges to a useful quantity (a high quality local minimum), if our networks are large enough.

Using these smoothness assumptions, it is not hard to ensure the required properties in (A2) and (A6) by relating the analysis of the loss surfaces of L_g and L_h to a stability analysis of the corresponding gradient systems. Further technical details can be found in Section A.6 in the appendix.

(A5) and (A7) Transition Kernel and Bounded Iterates. The transition kernel is continuous (c.f. Section A.6 in the appendix). Boundedness of $\boldsymbol{\theta}_n$ and $\boldsymbol{\omega}_n$ is achieved by weight decay terms in practice.

Proof of Theorem 1.

Proof. In the previous paragraphs we discussed how the assumptions of Theorem 2 can be fulfilled. □

Finite Greediness is Sufficient to Converge to the Optimal Policy. Regularization terms are weighted by $(z_2)_n$ which converges to zero, therefore the optimal policies are the same as without the regularization. There exists an optimal policy π^* that is deterministic according to Proposition 4.4.3 in Puterman (2005). We want to ensure via a parameter $(z_1)_n$ that the policy becomes more greedy during learning. If the policy is not greedy enough, estimates of the action-value or the advantage function may misguide the algorithm and the optimal policy is not found. For example, huge negative rewards if not executing the optimal actions may avoid convergence to the optimal policy if the policy is not greedy enough. $(z_1)_n$ directly enters the policy according to Eq. (15). We show that we can estimate how large $(z_1)_n$ must become in order to ensure that Q-value and policy gradient methods converge to an optimal policy, if it is the local minimum of the loss function (we cannot ensure this). For policy gradients, the optimal actions receive always the largest gradient and the policy converges to the optimal policy. The required greediness will be measured by the parameter $\beta > 1$. *In practical applications we know that β exists but do not know its value, since it depends on characteristics of the task and the optimal Q-values.* For a more formal treatment c.f. Section A.7 in the appendix, especially Lemma 2.

Conclusions and Outlook We showed local convergence of an abstract actor-critic setting and applied it to a version of PPO and RUDDER under practical assumptions. We intend to apply our results to similar practically relevant settings, e.g. the PPO algorithm discussed in Schulman et al. (2018). A further future direction is to guarantee convergence to an optimal policy. It would also be interesting to relax some of the required assumptions on the loss functions (e.g. by extending the techniques in Metrikopoulos et al. (2020) to two timescales) or elaborate on convergence rates.

REFERENCES

P. A. Absil and K. Kurdyka. On the stable equilibrium points of gradient systems. *Systems & Control Letters*, 55(7):573–577, 2006.

J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter. Rudder: Return decomposition for delayed rewards. *ArXiv*, 1806.07857, 2018.

J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems 33*, 2019. ArXiv 1806.07857.

B. Bakker. Reinforcement learning by backpropagation through an lstm model/critic. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 127–134, 2007. doi: 10.1109/ADPRL.2007.368179.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, Belmont, MA, 1996.

S. Bhatnagar, H. L. Prasad, and L. A. Prashanth. *Stochastic Recursive Algorithms for Optimization*. Lecture Notes in Control and Information Sciences. Springer-Verlag London, 2013. ISBN 978-1-4471-4284-3. doi: 10.1007/978-1-4471-4285-0.

V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*, volume 48 of *Texts and Readings in Mathematics*. Springer, 2008. ISBN 978-93-86279-38-5.

V. S. Borkar and S. P. Meyn. The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000. doi: 10.1137/S0363012997331639.

G. Casella and R. L. Berger. *Statistical Inference*. Wadsworth and Brooks/Cole, 2002.

A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pp. 192–204, 2015.

P. Dayan. The convergence of TD(λ) for general λ. *Machine Learning*, 8:341, 1992.

J. Fan, Z. Wang, Y. Xie, and Z. Yang. A theoretical analysis of deep q-learning. *CoRR*, abs/1901.00137, 2020.

M. Hairer. Ergodic properties of markov processes. Lecture notes, 2018.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. In I. Guyon, U. v. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6626–6637. Curran Associates, Inc., 2017. preprint arXiv:1706.08500.

C. Jin, P. Netrapalli, and M. I. Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *ArXiv*, 1902.00618, 2019.

P. Karmakar and S. Bhatnagar. Two time-scale stochastic approximation with controlled Markov noise and off-policy temporal-difference learning. *Mathematics of Operations Research*, 2017. doi: 10.1287/moor.2017.0855.

K. Kawaguchi. Deep learning without poor local minima. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 586–594, 2016.

K. Kawaguchi and Y. Bengio. Depth with nonlinearity creates no bad local minima in ResNets. *Neural Networks*, 118:167–174, 2019.

K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. *arXiv*, 1710.05468, 2017.

K. Kawaguchi, J. Huang, and L. P. Kaelbling. Effect of depth and width on local minima in deep learning. *Neural Computation*, 31(6):1462–1498, 2019.

V. R. Konda and V. S. Borkar. Actor-critic-type learning algorithms for Markov decision processes. *SIAM J. Control Optim.*, 38(1):94–123, 1999. doi: 10.1137/S036301299731669X.

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

V. R. Konda and J. N. Tsitsiklis. On actor-critic algorithms. *SIAM J. CONTROL OPTIM.*, 42(4): 1143–1166, 2003. doi: 10.1137/S0363012901385691.

H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Applied Mathematical Sciences. Springer-Verlag New York, 1978. ISBN 978-0-387-90341-5. doi: 10.1007/978-1-4684-9352-8.

H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Stochastic Modelling and Applied Probability. Springer New York, 2003. ISBN 9780387008943.

T. Lin, C. Jin, and M. I. Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *ArXiv*, 1906.00331, 2019.

B. Liu, Q. Cai, Z. Yang, and Z. Wang. Neural proximal/trust region policy optimization attains globally optimal policy. In *Advances in Neural Information Processing Systems 33*, volume ArXiv 1906.10306, 2019.

H. R. Maei, C. Szepesvári, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems 22*, pp. 1204–1212. Curran Associates, Inc., 2009.

E. V. Mazumdar, M. I. Jordan, and S. S. Sastry. On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games. *ArXiv*, 1901.00838, 2019.

P. Metrikopoulos, N. Hallak, A. Kavis, and V. Cevher. On the almost sure convergence of stochastic gradient descent in non-convex problems. *ArXiv*, 2006.11144, 2020.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing Atari with deep reinforcement learning. *ArXiv*, 1312.5602, 2013.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, , and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

P. W. Munro. A dual back-propagation scheme for scalar reinforcement learning. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society, Seattle, WA*, pp. 165–176, 1987.

OpenAI, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Jozefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. deOliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning. *arXiv*, 1912.06680, 2019.

M. L. Puterman. *Markov Decision Processes*. John Wiley & Sons, Inc., 2nd edition, 2005. ISBN 978-0-471-72782-8.

H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

A. J. Robinson. *Dynamic Error Propagation Networks*. PhD thesis, Trinity Hall and Cambridge University Engineering Department, 1989.

T. Robinson and F. Fallside. Dynamic reinforcement driven error propagation networks with application to game playing. In *Proceedings of the 11th Conference of the Cognitive Science Society, Ann Arbor*, pp. 836–843, 1989.

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. *ArXiv*, 1502.05477, 2015. 31st International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, volume 37.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *ArXiv*, 1707.06347, 2018.

S. Singh, T. Jaakkola, M. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38:287–308, 2000. doi: 10.1023/A:1007678930559.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, pp. 1057–1063, 2000.

J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16(3):185–202, 1994. doi: 10.1023/A:1022689125041.

O. Vinyals, I. Babuschkin, W. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. Agapiou, M. Jaderberg, and D. Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z.

C. J. C. H. Watkins and P. Dayan. Q-Learning. *Machine Learning*, 8:279–292, 1992.

T. Xu, S. Zou, and Y Liang. Two time-scale off-policy td learning: Non-asymptotic analysis over Markovian samples. In *Advances in Neural Information Processing Systems 32*, pp. 10633–10643, 2019.

Z. Yang, Y. Chen, M. Hong, and Z. Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. In *Advances in Neural Information Processing Systems 32*, pp. 8351–8363, 2019.

A APPENDIX

A.1 FURTHER DETAILS ON PPO

Here we describe the minimization problem for the PPO setup in a more detailed way by including the exact expression for the gradients of the respective loss functions:

$$\mathrm{L}_h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[-\, G_0 \;+\; (z_2)_n \; \rho(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] \,, \tag{16}$$

$$h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \Bigg[- \sum_{t=0}^{T} \nabla_{\boldsymbol{\theta}} \log \pi(a_t \mid s_t; \boldsymbol{\theta}_n, \boldsymbol{z}_n) \; (\hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n) \;-\; \hat{v}^{\pi}(s_t; \boldsymbol{\omega}_n))$$

$$+ (z_2)_n \; \sum_{t=0}^{T} \nabla_{\boldsymbol{\theta}_n} \log \pi(a_t \mid s_t; \boldsymbol{\theta}_n, \boldsymbol{z}_n) \; \rho(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \;+\; (z_2)_n \; \nabla_{\boldsymbol{\theta}_n} \rho(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \Bigg] \,,$$

$$\mathrm{L}_g^{\mathrm{TD}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[\frac{1}{2} \; \sum_{t=0}^{T} \left(\delta^{\mathrm{TD}}(t) \right)^2 \right] \,, \tag{17}$$

$$f^{\mathrm{TD}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[- \sum_{t=0}^{T} \delta^{\mathrm{TD}}(t) \; \nabla_{\boldsymbol{\omega}_n} \hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n) \right] \,, \tag{18}$$

$$\mathrm{L}_g^{\mathrm{MC}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;= \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[\frac{1}{2} \; \sum_{t=0}^{T} \left(G_t \;-\; \hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n) \right)^2 \right] \,, \tag{19}$$

$$f^{\mathrm{MC}}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[- \sum_{t=0}^{T} \left(G_t \;-\; \hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n) \right) \nabla_{\boldsymbol{\omega}_n} \hat{q}^{\pi}(s_t, a_t; \boldsymbol{\omega}_n) \right] \,,$$

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;-\; a(n) \; \hat{h}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \,, \quad \boldsymbol{\omega}_{n+1} \;=\; \boldsymbol{\omega}_n \;-\; b(n) \; \hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \,, \tag{20}$$

A.2 FURTHER DETAILS ON RUDDER

In a similar vein we present the minimization problem of RUDDER in more detail:

$$\mathrm{L}_h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \check{\pi}} \left[\frac{1}{2} \; \sum_{t=0}^{T} \left(R_{t+1}(\tau; \boldsymbol{\omega}_n) - \hat{q}(s_t, a_t; \boldsymbol{\theta}_n) \right)^2 \;+\; (z_2)_n \; \rho_{\boldsymbol{\theta}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] \tag{21}$$

$$h(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \check{\pi}} \left[- \sum_{t=0}^{T} \left(R_{t+1}(\tau; \boldsymbol{\omega}_n) - \hat{q}(s_t, a_t; \boldsymbol{\theta}_n) \right) \nabla_{\boldsymbol{\theta}} \hat{q}(s_t, a_t; \boldsymbol{\theta}_n) \right. \tag{22}$$

$$\left. + (z_2)_n \; \nabla_{\boldsymbol{\theta}} \rho_{\boldsymbol{\theta}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right]$$

$$\mathrm{L}_g(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[\frac{1}{2} \left(\sum_{t=0}^{T} \tilde{R}_{t+1} \;-\; g(\tau; \boldsymbol{\omega}_n) \right)^2 \;+\; (z_2)_n \; \rho_{\boldsymbol{\omega}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] \tag{23}$$

$$f(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \;=\; \mathrm{E}_{\tau \sim \pi(\boldsymbol{\theta}_n, \boldsymbol{z}_n)} \left[- \left(\sum_{t=0}^{T} \tilde{R}_{t+1} \;-\; g(\tau; \boldsymbol{\omega}_n) \right) \nabla_{\boldsymbol{\omega}} g(\tau; \boldsymbol{\omega}_n) \right. \tag{24}$$

$$\left. + (z_2)_n \; \nabla_{\boldsymbol{\omega}} \rho_{\boldsymbol{\omega}}(\tau, \boldsymbol{\theta}_n, \boldsymbol{z}_n) \right] \,,$$

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;-\; a(n) \; \hat{h}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \,, \quad \boldsymbol{\omega}_{n+1} \;=\; \boldsymbol{\omega}_n \;-\; b(n) \; \hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) \,, \tag{25}$$

A.3 CAUSALITY AND REWARD-TO-GO

This section is meant to provide the reader with more details concerning the causality assumption that leads to the formula for h in Eq. (16) for PPO. We can derive a formulation of the policy gradient with reward-to-go. For ease of notation, instead of using $\tilde{P}_{\pi}(\tau)$ as in previous sections, we here

denote the probability of state-action sequence $\tau = \tau_{0,T} = (s_0, s_0, s_1, a_1, \ldots, s_T, a_T)$ with policy π as

$$p(\tau) \;=\; p(s_0)\, \pi(a_0 \mid s_0) \prod_{t=1}^{T} p(s_t \mid s_{t-1}, a_{t-1})\, \pi(a_t \mid s_t) \tag{26}$$

$$=\; p(s_0) \prod_{t=1}^{T} p(s_t \mid s_{t-1}, a_{t-1}) \prod_{t=0}^{T} \pi(a_t \mid s_t)\;.$$

The probability of state-action sequence $\tau_{0,t} = (s_0, s_0, s_1, a_1, \ldots, s_t, a_t)$ with policy π is

$$p(\tau_{0,t}) \;=\; p(s_0)\, \pi(a_0 \mid s_0) \prod_{k=1}^{t} p(s_k \mid s_{k-1}, a_{k-1})\, \pi(a_k \mid s_k) \tag{27}$$

$$=\; p(s_0) \prod_{k=1}^{t} p(s_k \mid s_{k-1}, a_{k-1}) \prod_{k=0}^{t} \pi(a_k \mid s_k)\;.$$

The probability of state-action sequence $\tau_{t+1,T} = (s_{t+1}, a_{t+1}, \ldots, s_T, a_T)$ with policy π given (s_t, a_t) is

$$p(\tau_{t+1,T} \mid s_t, a_t) \;=\; \prod_{k=t+1}^{T} p(s_k \mid s_{k-1}, a_{k-1})\, \pi(a_k \mid s_k) \tag{28}$$

$$=\; \prod_{k=t+1}^{T} p(s_k \mid s_{k-1}, a_{k-1}) \prod_{k=t+1}^{T} \pi(a_k \mid s_k)\;.$$

The expectation of $\sum_{t=0}^{T} R_{t+1}$ is

$$\mathrm{E}_{\pi}\left[\sum_{t=0}^{T} R_{t+1}\right] \;=\; \sum_{t=0}^{T} \mathrm{E}_{\pi}\left[R_{t+1}\right]\;. \tag{29}$$

With $R_{t+1} \sim p(r_{t+1} \mid s_t, a_t)$, the random variable R_{t+1} depends only on (s_t, a_t). We define the expected reward $\mathrm{E}_{r_{t+1}}\left[R_{t+1} \mid s_t, a_t\right]$ as a function $r(s_t, a_t)$ of (s_t, a_t):

$$r(s_t, a_t) \;:=\; \mathrm{E}_{r_{t+1}}\left[R_{t+1} \mid s_t, a_t\right] \;=\; \sum_{r_{t+1}} p(r_{t+1} \mid s_t, a_t)\, r_{t+1}\;. \tag{30}$$

Causality. We assume that the reward $R_{t+1} = R(s_t, a_t) \sim p(r_{t+1} \mid s_t, a_t)$ only depends on the past but not on the future. The state-action pair (s_t, a_t) is determined by the past and not by the future. Relevant is only how likely we observe (s_t, a_t) and not what we do afterwards.

Causality is derived from the Markov property of the MDP and means:

$$\mathrm{E}_{\tau \sim \pi}\left[R_{t+1}\right] \;=\; \mathrm{E}_{\tau_{0,t} \sim \pi}\left[R_{t+1}\right]\;. \tag{31}$$

That is

$$\mathrm{E}_{\tau \sim \pi}\left[R_{t+1}\right] \;=\; \sum_{s_1} \sum_{a_1} \sum_{s_2} \sum_{a_2} \cdots \sum_{s_T} \sum_{a_T} p(\tau)\, r(s_t, a_t) \tag{32}$$

$$= \sum_{s_1} \sum_{a_1} \sum_{s_2} \sum_{a_2} \cdots \sum_{s_T} \sum_{a_T} \prod_{l=1}^{T} p(s_l \mid s_{l-1}, a_{l-1}) \prod_{l=1}^{T} \pi(a_l \mid s_l)\, r(s_t, a_t)$$

$$= \sum_{s_1} \sum_{a_1} \sum_{s_2} \sum_{a_2} \cdots \sum_{s_t} \sum_{a_t} \prod_{l=1}^{t} p(s_l \mid s_{l-1}, a_{l-1}) \prod_{l=1}^{t} \pi(a_l \mid s_l)\, r(s_t, a_t)$$

$$\sum_{s_{t+1}} \sum_{a_{t+1}} \sum_{s_{t+2}} \sum_{a_{t+2}} \cdots \sum_{s_T} \sum_{a_T} \prod_{l=t+1}^{T} p(s_l \mid s_{l-1}, a_{l-1}) \prod_{l=t+1}^{T} \pi(a_l \mid s_l)$$

$$= \sum_{s_1} \sum_{a_1} \sum_{s_2} \sum_{a_2} \cdots \sum_{s_t} \sum_{a_t} \prod_{l=1}^{t} p(s_l \mid s_{l-1}, a_{l-1}) \prod_{l=1}^{t} \pi(a_l \mid s_l)\, r(s_t, a_t)$$

$$= \mathrm{E}_{\tau_{0,t} \sim \pi}\left[R_{t+1}\right]\;.$$

Policy Gradient Theorem. We now assume that the policy π is parametrized by $\boldsymbol{\theta}$, that is, $\pi(a_t \mid s_t) = \pi(a_t \mid s_t; \boldsymbol{\theta})$. We need the gradient with respect to $\boldsymbol{\theta}$ of $\prod_{t=a}^{b} \pi(a_t \mid s_t)$:

$$\begin{aligned}
\nabla_\theta \prod_{t=a}^{b} \pi(a_t \mid s_t; \boldsymbol{\theta}) \;&=\; \sum_{s=a}^{b} \prod_{t=a, t\neq s}^{b} \pi(a_t \mid s_t; \boldsymbol{\theta})\, \nabla_\theta \pi(a_s \mid s_s; \boldsymbol{\theta}) \qquad (33)\\
&= \prod_{t=a}^{b} \pi(a_t \mid s_t; \boldsymbol{\theta}) \sum_{s=a}^{b} \frac{\nabla_\theta \pi(a_s \mid s_s; \boldsymbol{\theta})}{\pi(a_s \mid s_s; \boldsymbol{\theta})}\\
&= \prod_{t=a}^{b} \pi(a_t \mid s_t; \boldsymbol{\theta}) \sum_{s=a}^{b} \nabla_\theta \log \pi(a_s \mid s_s; \boldsymbol{\theta})\,.
\end{aligned}$$

It follows that

$$\nabla_\theta \mathrm{E}_\pi \left[R_{t+1}\right] \;=\; \mathrm{E}_\pi \left[\sum_{s=1}^{t} \nabla_\theta \log \pi(a_s \mid s_s; \boldsymbol{\theta})\, R_{t+1}\right]\,. \qquad (34)$$

We only have to consider the **reward to go**. Since a_0 does not depend on π, we have $\nabla_\theta \mathrm{E}_\pi \left[R_1\right] = 0$. Therefore

$$\begin{aligned}
\nabla_\theta \mathrm{E}_\pi \left[\sum_{t=0}^{T} R_{t+1}\right] \;&=\; \sum_{t=0}^{T} \nabla_\theta \mathrm{E}_\pi \left[R_{t+1}\right] \qquad (35)\\
&= \mathrm{E}_\pi \left[\sum_{t=1}^{T} \sum_{k=1}^{t} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, R_{t+1}\right]\\
&= \mathrm{E}_\pi \left[\sum_{k=1}^{T} \sum_{t=k}^{T} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, R_{t+1}\right]\\
&= \mathrm{E}_\pi \left[\sum_{k=1}^{T} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta}) \sum_{t=k}^{T} R_{t+1}\right]\\
&= \mathrm{E}_\pi \left[\sum_{k=1}^{T} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, G_k\right]\,.
\end{aligned}$$

We can express this by Q-values.

$$\begin{aligned}
&\mathrm{E}_\pi \left[\sum_{k=1}^{T} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, G_k\right] \qquad (36)\\
&= \sum_{k=1}^{T} \mathrm{E}_\pi \left[\nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, G_k\right]\\
&= \sum_{k=1}^{T} \mathrm{E}_{\tau_{0,k}\sim\pi} \left[\nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, \mathrm{E}_{\tau_{k+1,T}\sim\pi} \left[G_k \mid s_k, a_k\right]\right]\\
&= \sum_{k=1}^{T} \mathrm{E}_{\tau_{0,k}\sim\pi} \left[\nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, q^\pi(s_k, a_k)\right]\\
&= \mathrm{E}_{\tau\sim\pi} \left[\sum_{k=1}^{T} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, q^\pi(s_k, a_k)\right]\,.
\end{aligned}$$

We have finally:

$$\nabla_\theta \mathrm{E}_\pi \left[\sum_{t=0}^{T} R_{t+1}\right] \;=\; \mathrm{E}_{\tau\sim\pi} \left[\sum_{k=1}^{T} \nabla_\theta \log \pi(a_k \mid s_k; \boldsymbol{\theta})\, q^\pi(s_k, a_k)\right]\,. \qquad (37)$$

A.4 PRECISE STATEMENT OF ASSUMPTIONS

Here we provide a precise formulation of the assumptions from Karmakar & Bhatnagar (2017). The formulation we use here is mostly taken from Heusel et al. (2017):

(A1) *Assumptions on the controlled Markov processes:* The controlled Markov process $\boldsymbol{z}$ takes values in a compact metric space S. It is controlled by the iterate sequences $\boldsymbol{\theta}_n\}$ and $\boldsymbol{\omega}_n$ and furthermore $\boldsymbol{z}_n$ by a random process $\boldsymbol{a}_n$ taking values in a compact metric space W. For B Borel in S the $\boldsymbol{z}_n$ dynamics for $n \geq 0$ is determined by a transition kernel $\tilde{p}$:

$$\mathrm{P}(\boldsymbol{z}_{n+1} \in B | \boldsymbol{z}_l, \boldsymbol{a}_l, \boldsymbol{\theta}_l, \boldsymbol{\omega}_l, l \leqslant n) = \int_B \tilde{p}(\mathrm{d}\boldsymbol{z} | \boldsymbol{z}_n, \boldsymbol{a}_n, \boldsymbol{\theta}_n, \boldsymbol{\omega}_n).$$

(A2) *Assumptions on the update functions:* $h : \mathbb{R}^{m+k} \times S^{(1)} \to \mathbb{R}^m$ is jointly continuous as well as Lipschitz in its first two arguments, and uniformly w.r.t. the third. This means that for all $\boldsymbol{z} \in S$:

$$\|h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z}) \; - \; h(\boldsymbol{\theta}', \boldsymbol{w}', \boldsymbol{z})\| \leqslant \; L^{(1)} \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| + \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|\right).$$

Similarly for f, where the Lipschitz constant is $L^{(2)}$.

(A3) *Assumptions on the additive noise:* For $i = 1, 2$, $\{(\boldsymbol{m}_i)_n\}$ are martingale difference sequences with bounded second moments. More precisely, $(\boldsymbol{m}_i)_n$ are martingale difference sequences w.r.t. increasing σ-fields

$$\mathfrak{F}_n \; = \; \sigma(\boldsymbol{\theta}_l, \boldsymbol{\omega}_l, (\boldsymbol{m}_1)_l, (\boldsymbol{m}_2)_l, \boldsymbol{z}_l, l \leqslant n),$$

satisfying $\mathrm{E}\left[\|(\boldsymbol{m}_i)_n\|^2 \mid \mathfrak{F}_n\right] \; \leqslant \; B_i$ for $n \geq 0$ and a given constants B_i.

(A4) *Assumptions on the learning rates:*

$$\sum_n a(n) \; = \; \infty, \quad \sum_n a^2(n) \; < \; \infty,$$
$$\sum_n b(n) \; = \; \infty, \quad \sum_n b^2(n) \; < \; \infty,$$

and $a(n) \; = \; \mathrm{o}(b(n))$. Furthermore, $a(n), b(n), n \geq 0$ are non-increasing.

(A5) *Assumptions on the transition kernels:* The state-action map

$$S \times W \times \mathbb{R}^{m+k} \ni (\boldsymbol{z}, \boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\omega}) \mapsto \; \tilde{p}(\mathrm{d}\boldsymbol{y} \mid \boldsymbol{z}, \boldsymbol{a}, \boldsymbol{\theta}, \boldsymbol{\omega})$$

is continuous (the topology on the spaces of probability measures is induced by weak convergence).

(A6) *Assumptions on the associated ODEs:* We consider occupation measures which intuitively give for the controlled Markov process the probability or density to observe a particular state-action pair from $S \times W$ for given $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ and a given control. A precise definition of these occupation measures can be found e.g. on page 68 of Borkar (2008) or page 5 in Karmakar & Bhatnagar (2017). We have following assumptions:

- We assume that there exists only one such ergodic occupation measure for $\boldsymbol{z}_n$ on $S \times W$, denoted by $\Gamma_{\boldsymbol{\theta},\boldsymbol{\omega}}$. A main reason for assuming uniqueness is that it enables us to deal with ODEs instead of differential inclusions. Moreover, set $\tilde{f}(\boldsymbol{\theta}, \boldsymbol{\omega}) \; = \; \int f(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z}) \, \Gamma_{\boldsymbol{\theta},\boldsymbol{\omega}}(\mathrm{d}\boldsymbol{z}, W)$.
- We assume that for $\boldsymbol{\theta} \in \mathbb{R}^m$, the ODE $\dot{\boldsymbol{\omega}}(t) \; = \; \tilde{f}(\boldsymbol{\theta}, \boldsymbol{\omega}(t))$ has a unique asymptotically stable equilibrium $\boldsymbol{\lambda}(\boldsymbol{\theta})$ with attractor set $B_{\boldsymbol{\theta}}$ such that $\boldsymbol{\lambda} : \mathbb{R}^m \to \mathbb{R}^k$ is a Lipschitz map with global Lipschitz constant.
- The Lyapunov function $V(\boldsymbol{\theta}, .)$ associated to $\boldsymbol{\lambda}(\boldsymbol{\theta})$ is continuously differentiable.
- Next define $\tilde{h}(\boldsymbol{\theta}) \; = \; \int h(\boldsymbol{\theta}, \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{z}) \, \Gamma_{\boldsymbol{\theta},\boldsymbol{\lambda}(\boldsymbol{\theta})}(\mathrm{d}\boldsymbol{z}, W)$. We assume that the ODE $\dot{\boldsymbol{\theta}}(t) \; = \; \tilde{h}(\boldsymbol{\theta}(t))$ has a global attractor set A.
- For all $\boldsymbol{\theta}$, with probability 1, $\boldsymbol{\omega}_n$ for $n \geq 1$ belongs to a compact subset $Q_{\boldsymbol{\theta}}$ of $B_{\boldsymbol{\theta}}$ "eventually".

This assumption is an adapted version of (A6)' of Karmakar & Bhatnagar (2017), to avoid too many technicalities (e.g. in Karmakar & Bhatnagar (2017) two controls are used, which we avoid here to not overload notation).

(A7) *Assumption of bounded iterates:* $\sup_n \|\boldsymbol{\theta}_n\| \; < \; \infty$ and $\sup_n \|\boldsymbol{\omega}_n\| \; < \; \infty$ a.s.

A.5 FURTHER DETAILS CONCERNING THE SAMPLING PROCESS

Let us formulate the construction of the sampling process in more detail: We introduced the function S_π in the main paper as follows:

$$S_\pi : \Omega \to \tilde{\Omega}_\pi,\ x \mapsto \operatorname*{argmax}_{\tau \in \tilde{\Omega}_\pi} \left\{ \sum_{\eta \leq \tau} \tilde{P}_\pi(\eta) \leq x \right\}.$$

Now S_π basically divides the interval $[0,1]$ into finitely many disjoint subintervals, such that the i-th subinterval I_i maps to the i-th element $\tau_i \in \tilde{\Omega}_\pi$, and additionally the length of I_i is given by $\tilde{P}_\pi(\tau_i)$. S_π is measurable, because the pre-image of any element of the sigma-algebra $\tilde{\mathfrak{A}}_\pi$ wrt. S_π is just a finite union of subintervals of $[0,1]$, which is clearly contained in the Borel-algebra. Basically S_π just describes how to get one sample from a multinomial distribution with (finitely many) probabilities $\tilde{P}_\pi(\tau)$, where $\tau \in \tilde{\Omega}_\pi$. Compare with inverse transform sampling, e.g. Theorem 2.1.10. in Casella & Berger (2002) and applications thereof. For the reader's convenience let's briefly recall this important concept here in a formal way:

Lemma 1 (Inverse transform sampling)**.** *Let X have continuous cumulative distribution $F_X(x)$ and define the random variable Y as $Y = F_X(X)$. Then Y is uniformly distributed on $(0,1)$.*

A.6 FURTHER DETAILS FOR PROOF OF THEOREM 1

Here we provide further technical details needed to ensure the assumptions stated before to prove our main theorem Theorem 1.

(A1) *Assumptions on the controlled Markov processes:* Let us start by discussing more details for controlled processes that appear in the PPO and RUDDER setting. Let us focus on the process related to $(z_1)_n$: Let $\beta > 1$ and let the real sequence z_n be defined by $(z_1)_1 = 1$ and $(z_1)_{n+1} = (1 - \frac{1}{\beta})(z_1)_n + 1$. The z_n's are nothing more but the partial sums of a geometric series converging to β.

The sequence $(z_1)_n$ can also be interpreted as a time-homogeneous Markov process $(\boldsymbol{z}_1)_n$ with transition probabilities given by

$$P(z, y) = \delta_{(1-\frac{1}{\beta})z+1}, \tag{38}$$

where δ denotes the Dirac measure, and with the compact interval $[1, \beta]$ as its range. We use the standard notation for discrete time Markov processes, described in detail e.g. in Hairer (2018). Its unique invariant measure is clearly δ_β. So integrating wrt. this invariant measure will in our case just correspond to taking the limit $(z_1)_n \to \beta$.

(A2) *h and f are Lipschitz:* By the mean value theorem it is enough to show that the derivatives wrt. $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ are bounded uniformly wrt. $\boldsymbol{z}$. We only show details for f, since for h similar considerations apply. By the explicit formula for L_g, we see that $f(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$ can be written as:

$$\sum_{\substack{s_1,\dots,s_T \\ a_1,\dots,a_T}} \prod_{t=1}^{T} p(s_t \mid s_{t-1}, a_{t-1}) \pi(a_t \mid s_t, \boldsymbol{\theta}, \boldsymbol{z}) \nabla_{\boldsymbol{\omega}} \Phi(g(\tau; \boldsymbol{\omega}, \boldsymbol{z}), \tau, \boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z}).$$

The claim can now be readily deduced from the assumptions (L1)–(L3).

(A3) *Martingale difference property and estimates:* From the results in the main paper on the probabilistic setting, $(\boldsymbol{m}_1)_{n+1}$ and $(\boldsymbol{m}_2)_{n+1}$ can easily be seen to be martingale difference sequences wrt. their filtrations $\mathfrak{F}_n$. Indeed, the sigma algebras created by $\boldsymbol{\omega}_n$ and $\boldsymbol{\theta}_n$ already describe $\tilde{\mathfrak{A}}_{\pi_{\boldsymbol{\theta}_n}}$, and thus:

$$\mathrm{E}[(\boldsymbol{m}_i)_{n+1} | \mathfrak{F}_n] = \mathrm{E}[\hat{f}(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n) | \mathfrak{F}_n] - \mathrm{E}[f(\boldsymbol{\theta}_n, \boldsymbol{\omega}_n, \boldsymbol{z}_n)] = 0.$$

It remains to show that $\mathrm{E}[|(\boldsymbol{m}_i)_{n+1}|^2 | \mathfrak{F}_n] \leq B_i$ for $i = 1, 2$. This, however, is also clear, since all the involved expressions are bounded uniformly again by the assumptions (L1)–(L3) on the losses (e.g. one can observe this by writing down the involved expressions explicitly as indicated in the previous point (A2)).

(A4) *Assumptions on the learning rates:* These standard assumptions are taken for granted.

(A5) *Transition kernels:* The continuity of the transition kernels is clear from Eq. (38) (continuity is wrt. to the weak topology in the space of probability measures. So in our case, this again boils down to using continuity of the test functions).

(A6) *Stability properties of the ODEs:*
• By the explanations for (A1) we mentioned that integrating wrt. the ergodic occupation measure in our case corresponds to taking the limit $\boldsymbol{z}_n \to \boldsymbol{z}$ (since our Markov processes can be interpreted as sequences). Thus $\tilde{f}(\boldsymbol{\theta}, \boldsymbol{\omega}) = f(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$. In the sequel we will also use the following abbreviations: $f(\boldsymbol{\theta}, \boldsymbol{\omega}) = f(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$, $h(\boldsymbol{\theta}, \boldsymbol{\omega}) = h(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{z})$, etc.. Now consider the ODE

$$\dot{\boldsymbol{\omega}}(t) = f(\boldsymbol{\theta}, \boldsymbol{\omega}(t)), \tag{39}$$

where $\boldsymbol{\theta}$ is fixed. Eq. (39) can be seen as a gradient system for the function L_g. By standard results on gradient systems (cf. e.g. Section 4 in Absil & Kurdyka (2006) for a nice summary), which guarantee equivalence between strict local minima of the loss function and asymptotically stable points of the associated gradient system, we can use the assumptions (L1)–(L3) and the remarks thereafter from the main paper to ensure that there exists a unique asymptotically stable equilibrium $\boldsymbol{\lambda}(\boldsymbol{\theta})$ of Eq. (39).
• The fact that $\boldsymbol{\lambda}(\boldsymbol{\theta})$ is smooth enough can be deduced by the Implicit Function Theorem as discussed in the main paper.
• For Eq. (39) $L_g(\boldsymbol{\theta}, \boldsymbol{\omega}) - L_g(\boldsymbol{\theta}, \boldsymbol{\lambda}(\boldsymbol{\theta}))$ can be taken as associated Lyapunov function $V_{\boldsymbol{\theta}}(\boldsymbol{\omega})$, and thus $V_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ clearly is differentiable wrt. $\boldsymbol{\omega}$ for any $\boldsymbol{\theta}$.
• The slow ODE $\dot{\boldsymbol{\theta}}(t) = h(\boldsymbol{\theta}(t), \boldsymbol{\lambda}(\boldsymbol{\theta}(t))$ also has a unique asymptotically stable fixed point, which again is guaranteed by our assumptions and the standard results on gradient systems.

(A7) *Assumption of bounded iterates:* This follows from the assumptions on the loss functions.

A.7 Finite Greediness is sufficient to converge to the optimal policy

Here we provide details on how the optimal policy can be deduced using only a finite parameter $\beta > 1$. The Q-values for policy π are:

$$q^{\pi}(s_t, a_t) \;=\; \mathrm{E}_{\pi}\left[\sum_{\tau=t}^{T} R_{\tau+1} \mid s_t, a_t\right] \;=\; \sum_{\substack{s_t,\ldots,s_T \\ a_t,\ldots,a_T}} \prod_{\tau=t}^{T-1} p(s_{\tau+1} \mid s_\tau, a_\tau) \prod_{\tau=t}^{T} \pi(a_\tau \mid s_\tau) \sum_{\tau=t}^{T} R_{\tau+1}\,.$$

The optimal policy π^* is known to be deterministic $\left(\prod_{t=1}^{T} \pi^*(a_t \mid s_t) \in \{0, 1\}\right)$. Let us assume that the optimal policy is also unique. Then we are going to show the following result:

Lemma 2. *For $i_{\max} = \arg\max_i q^{\pi^*}(s, a^i)$ and $v^{\pi^*}(s) = \max_i q^{\pi^*}(s, a^i)$. We define*

$$0 \;<\; \epsilon \;<\; \min_{s, i \neq i_{\max}} \left(v^{\pi^*}(s) \;-\; q^{\pi^*}(s, a^i)\right), \tag{40}$$

We assume a function $\psi(s, a^i)$ that defines the actual policy π via

$$\pi(a^i \mid s; \beta) \;=\; \frac{\exp(\beta\, \psi(s, a^i))}{\sum_j \exp(\beta\, \psi(s, a^j))}\,. \tag{41}$$

We assume that the function ψ already identified the optimal actions, which will occur during learning at some time point when the policy is getting more greedy:

$$0 \;<\; \delta \;<\; \min_{s, i \neq i_{\max}} \left(\psi(s, a^{i_{\max}}) \;- \psi(s, a^i)\right). \tag{42}$$

Hence,

$$\lim_{\beta \to \infty} \pi(a^i \mid s; \beta) \;=\; \pi^*(a^i \mid s)\,. \tag{43}$$

We assume that

$$\beta \;> \max\left(\frac{\log(|\mathcal{A}| - 1)}{\delta}, -\log\left(\frac{\epsilon}{2T\,(|\mathcal{A}| - 1)\,|\mathcal{S}|^T\,|\mathcal{A}|^T\,(T+1)\,K_R}\right)/\delta\right). \tag{44}$$

Then we can make the statement for all s:

$$\forall_{j, j \neq i} :\; q^{\pi}(s, a^i) \;>\; q^{\pi}(s, a^j) \;\Rightarrow\; i = i_{\max}\,, \tag{45}$$

therefore the Q-values $q^{\pi}(s, a^i)$ determine the optimal policy as the action with the largest Q-value can be chosen.

More importantly, β is large enough to allow Q-value based methods and policy gradients converge to the optimal policy if it is the local minimum of the loss functions. For Q-value based methods the optimal action can be determined if the optimal policy is the minimum of the loss functions. For policy gradients the optimal action receives always the largest gradient and the policy converges to the optimal policy.

Proof. We already discussed that the optimal policy π^* is known to be deterministic $\left(\prod_{t=1}^{T} \pi^*(a_t \mid s_t) \in \{0,1\}\right)$. Let us assume that the optimal policy is also unique. Since

$$\pi(a^i \mid s; \beta) \;=\; \frac{\exp(\beta\,(\psi(s,a^i) \;-\; \psi(s,a^{i_{\max}})))}{\sum_j \exp(\beta\,\psi(s,a^j) \;-\; \psi(s,a^{i_{\max}}))}\,, \tag{46}$$

we have

$$\pi(a^{i_{\max}} \mid s; \beta) \;=\; \frac{1}{1 \;+\; \sum_{j, j\neq i_{\max}} \exp(\beta\,\psi(s,a^j) \;-\; \psi(s,a^{i_{\max}}))} \;>\; \frac{1}{1 \;+\; (|\mathcal{A}|-1)\;\exp(-\,\beta\,\delta)} \tag{47}$$

$$=\; 1 \;-\; \frac{(|\mathcal{A}|-1)\;\exp(-\,\beta\,\delta)}{1 \;+\; (|\mathcal{A}|-1)\;\exp(-\beta\,\delta)} \;>\; 1 \;-\; (|\mathcal{A}|-1)\;\exp(-\,\beta\,\delta)$$

and for $i \neq i_{\max}$

$$\pi(a^i \mid s; \beta) \;=\; \frac{\exp(\beta\,(\psi(s,a^i) \;-\; \psi(s,a^{i_{\max}})))}{1 \;+\; \sum_{j, j\neq i_{\max}} \exp(\beta\,\psi(s,a^j) \;-\; \psi(s,a^{i_{\max}}))} \;<\; \exp(-\,\beta\,\delta)\,. \tag{48}$$

For $\prod_{t=1}^{T} \pi^*(a_t \mid s_t) = 1$, we have

$$\prod_{t=1}^{T} \pi(a_t \mid s_t) \;>\; (1 \;-\; (|\mathcal{A}|-1)\;\exp(-\,\beta\,\delta))^T \;>\; 1 - \;T\,(|\mathcal{A}|-1)\;\exp(-\,\beta\,\delta)\,, \tag{49}$$

where in the last step we used that $(|\mathcal{A}|-1)\exp(-\beta\delta) < 1$ by definition of β in (44) so that an application of Bernoulli's inequality is justified. For $\prod_{t=1}^{T} \pi^*(a_t \mid s_t) = 0$, we have

$$\prod_{t=1}^{T} \pi(a_t \mid s_t) \;<\; \exp(-\,\beta\,\delta)\,. \tag{50}$$

Therefore

$$\left|\prod_{t=1}^{T} \pi^*(a_t \mid s_t) \;-\; \prod_{t=1}^{T} \pi(a_t \mid s_t)\right| \;<\; T\,(|\mathcal{A}|-1)\;\exp(-\,\beta\,\delta)\,. \tag{51}$$

Using Eq. (51) and the definition of β in Eq. (44) we get:

$$\left|q^{\pi^*}(s,a^i) \;-\; q^{\pi}(s,a^i)\right| \tag{52}$$

$$= \left|\sum_{\substack{s_1,..,s_T \\ a_1,...,a_T}} \prod_{t=1}^{T} p(s_t \mid s_{t-1}, a_{t-1}) \left(\prod_{t=1}^{T} \pi^*(a_t \mid s_t) \;-\; \prod_{t=1}^{T} \pi(a_t \mid s_t)\right) \sum_{t=0}^{T} R_{t+1}\right|$$

$$< \sum_{\substack{s_1,..,s_T \\ a_1,...,a_T}} \prod_{t=1}^{T} p(s_t \mid s_{t-1}, a_{t-1}) \left|\prod_{t=1}^{T} \pi^*(a_t \mid s_t) \;-\; \prod_{t=1}^{T} \pi(a_t \mid s_t)\right| (T+1)\;K_R$$

$$< \sum_{\substack{s_1,..,s_T \\ a_1,...,a_T}} \left|\prod_{t=1}^{T} \pi^*(a_t \mid s_t) \;-\; \prod_{t=1}^{T} \pi(a_t \mid s_t)\right| (T+1)\;K_R$$

$$< \;|\mathcal{S}|^T\;|\mathcal{A}|^T\;\frac{\epsilon}{2|\mathcal{S}|^T\;|\mathcal{A}|^T\;(T+1)\;K_R}\;(T+1)\;K_R \;=\; \epsilon/2\,.$$

Now from the condition that $q^{\pi}(s,a^i) \;>\; q^{\pi}(s,a^j)$ for all $j \neq i$ we can conclude that

$$q^{\pi^*}(s,a^j) \;-\; q^{\pi^*}(s,a^i) \;<\; (q^{\pi}(s,a^j) \;+\; \epsilon/2) \;-\; (q^{\pi}(s,a^i) \;-\; \epsilon/2) \;<\; \epsilon \tag{53}$$

for all $j \neq i$. Thus for $j \neq i$ it follows that $j \neq i_{\max}$ and consequently $i = i_{\max}$.

□