

# APPROXIMATE NEWTON POLICY GRADIENT ALGORITHMS\*

HAOYA LI<sup>†</sup>, SAMARTH GUPTA<sup>‡</sup>, HSIANGFU YU<sup>§</sup>, LEXING YING<sup>¶</sup>, AND INDERJIT DHILLON<sup>||</sup>

**Abstract.** Policy gradient algorithms have been widely applied to Markov decision processes and reinforcement learning problems in recent years. Regularization with various entropy functions is often used to encourage exploration and improve stability. This paper proposes an approximate Newton method for the policy gradient algorithm with entropy regularization. In the case of Shannon entropy, the resulting algorithm reproduces the natural policy gradient algorithm. For other entropy functions, this method results in brand-new policy gradient algorithms. We prove that all these algorithms enjoy Newton-type quadratic convergence and that the corresponding gradient flow converges globally to the optimal solution. Using synthetic and industrial-scale examples, we demonstrate that the proposed approximate Newton method typically converges in single-digit iterations, often orders of magnitude faster than other state-of-the-art algorithms.

**Key words.** policy gradient algorithm, approximate Newton method, quadratic convergence, Markov decision process, entropy regularization, reinforcement learning

**AMS subject classifications.** 49M15, 65K10, 68T05, 90C06, 90C40

**1. Introduction.** Consider an infinite-horizon Markov decision process (MDP) [4, 33]  $\mathcal{M} = (S, A, P, r, \gamma)$ , where  $S$  is a set of states of the system studied,  $A$  is a set of actions made by the agent,  $P$  is a transition probability tensor with  $P_{st}^a$  being the probability of transitioning from state  $s$  to state  $t$  when taking action  $a$ ,  $r$  is a reward tensor with  $r_s^a$  being the reward obtained when taking action  $a$  at state  $s$ , and  $0 < \gamma < 1$  is a discount factor. Throughout the paper, the state space  $S$  and the action space  $A$  are assumed to be finite. A policy  $\pi$  is a randomized rule of action-selection where  $\pi_s^a$  denotes the probability of choosing action  $a$  at state  $s$ . For a given policy  $\pi$ , the value function  $v_\pi$  is defined as

$$(1.1) \quad (v_\pi)_s = \mathbb{E} \sum_{k=0}^{\infty} (\gamma^k r_{s_k}^{a_k} \mid s_0 = s),$$

which satisfies the Bellman equation:

$$(1.2) \quad (I - \gamma P_\pi)v_\pi = r_\pi,$$

where  $(P_\pi)_{st} = \sum_a \pi_s^a P_{st}^a$ ,  $(r_\pi)_s = \sum_a \pi_s^a r_s^a$ , and  $I$  is the identity operator.

In order to promote exploration and enhance stability, one often regularizes the problem with a function  $h_\pi$  such as the negative Shannon entropy  $(h_\pi)_s = \sum_a \pi_s^a \log \pi_s^a$ . With the regularization  $h_\pi$ , the original reward  $r_\pi$  is replaced with the regularized reward  $r_\pi - \tau h_\pi$  where  $\tau > 0$  is a regularization coefficient and (1.2) becomes

$$(1.3) \quad (I - \gamma P_\pi)v_\pi = r_\pi - \tau h_\pi,$$

where we overload the notation  $v_\pi$  for the regularized value function. Other continuously differentiable entropy functions can be used as well, as we will show later. Since  $\gamma < 1$  and  $P_\pi$  is a transition probability matrix,  $(I - \gamma P_\pi)$  is invertible, and

$$(1.4) \quad v_\pi = (I - \gamma P_\pi)^{-1}(r_\pi - \tau h_\pi).$$

In a policy optimization problem, we seek a policy  $\pi$  that maximizes the value function  $v_\pi$ . According to the theory of regularized MDPs [9], when the regularization is strongly convex, there

\*Accepted by SIAM SISC May 19th, 2023.

<sup>†</sup>Stanford University, Stanford, CA ([lihaoya@stanford.edu](mailto:lihaoya@stanford.edu)).

<sup>‡</sup>Carnegie Mellon University, Pittsburgh, PA ([samarthg@andrew.cmu.edu](mailto:samarthg@andrew.cmu.edu)).

<sup>§</sup>Amazon Search, Palo Alto, CA ([rofu.yu@gmail.com](mailto:rofu.yu@gmail.com)).

<sup>¶</sup>Stanford University, Stanford, CA; Amazon Search, Palo Alto, CA ([lexing@gmail.com](mailto:lexing@gmail.com)).

<sup>||</sup>University of Texas at Austin and Google, work done while at Amazon Search ([inderjit@cs.utexas.edu](mailto:inderjit@cs.utexas.edu)).

is a unique optimal policy  $\pi^*$  such that  $(v_{\pi^*})_s \geq (v_\pi)_s$  for any policy  $\pi$  and state  $s$ . It thus suffices to maximize  $\rho^\top v_\pi$  for any positive weight vector  $\rho \in \mathbb{R}_+^{|S|}$ . Using (1.4), the problem can be stated as

$$(1.5) \quad \max_{\pi} \rho^\top (I - \gamma P_\pi)^{-1} (r_\pi - \tau h_\pi).$$

This problem can be solved by, for example, the policy gradient (PG) method. However, the vanilla PG method converges quite slowly. In [1], for instance, the vanilla PG method is shown to have the  $O(T^{-1})$  convergence rate, where  $T$  denotes the number of iterations. A widely used variant of PG is the softmax policy gradient (SPG) method, where a softmax parameterization is applied before taking gradient updates, which has been shown in [15] to require  $O(|S|^{2^{\frac{1}{1-\gamma}}})$  iterations to converge for certain MDPs without regularization. For the PG method with entropy regularization and some of its variants, the convergence rate can be improved to  $O(e^{-cT})$ , i.e., linear convergence [17], which can still be slow since the constant  $c$  in the linear convergence rate  $O(e^{-cT})$  is in general close to 0. It is also demonstrated in numerical examples that these algorithms with linear rates can experience slow convergence. For example, in the example in [40], thousands of iterations are needed for the algorithm to converge, even though the model is relatively small and sparse. Therefore, there is a clear need for designing new methods with faster convergence and one idea is to take the geometry of the problem into consideration. The Newton method, for example, preconditions the gradient with the Hessian matrix and obtains second-order local convergence. Since the exact Hessian matrix is usually too computationally expensive to obtain, the approximate Newton methods (including quasi-Newton methods), which use structurally simpler approximations of the Hessian instead, are more widely adopted in generic optimization problems and are known to enjoy superlinear convergence [25, 26].

**1.1. Contributions.** In this paper, we investigate the approximate Newton approach for solving (1.5). The main contributions of this paper are the following.

- First, we present a unified approximate Newton method for the policy optimization problem. The main observation is to decompose the Hessian as a sum of a diagonal matrix and a remainder that vanishes at the optimal solution. This inspires us to use only the diagonal matrix in the approximate Newton method. As a result, the proposed method not only leverages the second-order information but also enjoys low computational cost due to the diagonal structure of the preconditioner used. When the negative Shannon entropy is used, this method reproduces the natural policy gradient (NPG) algorithm. For other forms of entropy regularization, this method results in brand-new policy gradient algorithms.
- Second, we analyze the convergence property of the proposed approximate Newton algorithms and demonstrate local quadratic convergence both theoretically and numerically. By leveraging the framework of Newton-type methods (see [8] for example), we provide a simple and straightforward proof for quadratic convergence near the optimal policy. In the numerical tests, we verify that the proposed method leads to fast quadratic convergence even under small regularization and large discount rates (close to 1). Even for industrial-size problems with hundreds of thousands of states, the approximate Newton method converges in single-digit iterations and within a few minutes on a regular laptop. We also prove the global convergence of the approximate Newton gradient flow to the optimal solutions.

**1.2. Background and related work.** A major workhorse behind the recent success of reinforcement learning (RL) is the large family of policy gradient (PG) methods [38, 34], for example, the natural policy gradient (NPG) method [12], the actor-critic method [13], the asynchronous advantage actor-critic (A3C) method [19], the deterministic policy gradient (DPG) method [32], the trust region policy optimization (TRPO) [28], the generalized advantage estimation (GAE) [29], and proximal policy optimization (PPO) [30], to mention but a few. The NPG method is known to be drastically faster than the original PG method because the policy gradient in NPG is preconditioned by the Fisher information (an approximation of the Hessian of the KL-divergence)

matrix in order to fit the problem geometry better. This idea is extended in TRPO and PPO where the problem geometry is taken into consideration via trust region constraints (in terms of KL-divergence) and a clipping function of the relative ratio of policies in the objective function, respectively. These implicit ways (in the sense that they do not adjust the gradient by an explicit preconditioner) of adjusting the policy gradient are in essence similar to the mirror descent (MD) method [20] in generic optimization problems.

This similarity in addressing the inherent geometry of the problem is noticed by a line of recent work including [22, 9, 31, 35, 14], and the analysis techniques in MD methods have been adapted to the PG setting. The connection was first built explicitly in [22]. The authors consider a linear program formulation where the objective function is the average reward and the domain is the set of stationary state-action distributions, in which case the TRPO method can be viewed as an approximate mirror descent method and the A3C method as an MD method for the dual-averaging [21] objective. As a complement, [9] considers an actor-critic type method where the policy is updated via either a regularized greedy step or an MD step, and the value function is updated by a regularized Bellman operator, which also includes TRPO as a special case, and error propagation analysis is provided. In [31], an adaptive scaling that naturally arises in the policy gradient is applied to the proximity term of the MD formulation, and the sublinear convergence result is proved with a properly decreasing learning rate. In [35], the application to the non-tabular setting is enabled by parameterizing the policy and applying MD to the policy parameters, and the corresponding sublinear convergence result is presented.

Regularization, a strategy that considers the modified objective function with an additional penalty term on the policy, is another crucial component in the development of PG-type methods. Intuitively, regularization is able to encourage exploration in the policy iteration process and thus avoid local minima. It is also suggested [2] that regularization makes the optimization landscape smoother and thus enables possibly faster convergence. Linear convergence results are then established for regularized PG and NPG methods [1, 17, 6]. In these relatively earlier works [1, 17, 6], the regularization usually takes the form of (negative) entropy or relative entropy. In the more recent work [14] and [40] that follow the MD type methods, the regularization is extended to general convex functions with the resulting Bregman divergences different from the KL-divergence and linear convergence is guaranteed as well.

However, most of these algorithms are of either sublinear or linear convergence except the entropy regularized NPG with full step length (which is a special case of the approximate Newton method we propose), and even the linear convergence rate  $O(e^{-cT})$  can be slow since  $c$  can be close to zero. This motivates us to invent the approximate Newton policy gradient method to be introduced in [section 2](#).

## 2. Approximate Newton method.

**2.1. Approximate Newton method and entropy regularized natural policy gradient.** This section derives the approximate Newton method for the entropy regularized policy optimization problems. The idea is to approximate the Hessian with a simpler matrix whose inverse is easy to compute. We start with the negative Shannon entropy  $(h_\pi)_s = \sum_a \pi_s^a \log \pi_s^a$ .

In what follows, it is assumed that  $\pi^*$  is the optimizer of the problem stated in (1.5). By introducing  $Z_\pi := I - \gamma P_\pi$ , the objective function can be written as

$$(2.1) \quad E(\pi) \equiv \rho^\top (I - \gamma P_\pi)^{-1} (r_\pi - \tau h_\pi) = \rho^\top Z_\pi^{-1} (r_\pi - \tau h_\pi) = w_\pi^\top (r_\pi - \tau h_\pi),$$

where  $w_\pi := Z_\pi^{-\top} \rho$ .

Let us first outline the main idea of the approximate Newton method. The gradient  $\nabla_\pi E$  in  $\mathbb{R}^{|S||A|}$  of  $E(\pi)$  has entries given by

$$(2.2) \quad \frac{\partial E}{\partial \pi_s^a} = (r_s^a - \tau(\log \pi_s^a + 1)) - [(I - \gamma P^a)v_\pi]_s + c_s)(w_\pi)_s,$$

where  $c_s$  is a multiplier associated with the constraint  $\sum_a \pi_s^a = 1$  that depends on  $s$ . Our key observation is to decompose the Hessian matrix  $D^2 E(\pi)$  in  $\mathbb{R}^{|S||A| \times |S||A|}$  into two parts

$$(2.3) \quad D^2 E(\pi) = \Lambda(\pi) + \Delta(\pi),$$

where  $\Lambda(\pi)$  is a *diagonal* matrix given by  $\Lambda_{(sa),(tb)} = -\tau\delta_{\{(sa),(tb)\}} \frac{(w_\pi)_s}{\pi_s^a}$  and  $\Delta(\pi)$  is a remainder that *vanishes* at  $\pi = \pi^*$ , i.e.,  $\Delta(\pi) = O(\|\pi - \pi^*\|)$  (shown in [Theorem 2.1](#)). We emphasize that  $\Lambda(\pi)$  is in general not the diagonal part of the Hessian matrix  $D^2E(\pi)$ , but a diagonal approximation to it. With this decomposition, we can approximate the Hessian matrix  $D^2E(\pi)$  by  $\Lambda(\pi)$  and obtain the following *approximate Newton flow*:

$$\begin{aligned} \frac{d\pi_s^a}{dt} &= -(\Lambda^{-1}\nabla_\pi E)_{sa} = -(\Lambda_{(sa),(sa)})^{-1} \frac{\partial E}{\partial \pi_s^a} \\ &= \pi_s^a (r_s^a - \tau(\log \pi_s^a + 1)) - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau, \end{aligned}$$

By introducing the parameterization  $\theta_s^a = \log \pi_s^a$  and discretizing in time with learning rate  $\eta$ , we arrive at

$$\theta_s^a \leftarrow \eta(r_s^a - \tau - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau + (1 - \eta)\theta_s^a.$$

Writing this update back in terms of  $\pi_s^a$  leads to the following update rule

$$\pi_s^a \propto (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s)/\tau),$$

which coincides with the NPG scheme with entropy regularization. This result is summarized in the following theorem with the proof given in [subsection 5.1](#).

**THEOREM 2.1.** *Let  $h_\pi \in \mathbb{R}^{|S|}$  be the negative Shannon entropy  $(h_\pi)_s = \sum_a \pi_s^a \log \pi_s^a$ .*

(a) *There exists a diagonal approximation  $\Lambda(\pi)$  of the Hessian matrix  $D^2E(\pi)$  given by  $\Lambda_{(sa),(tb)} = -\tau\delta_{\{(sa),(tb)\}} \frac{(w_\pi)_s}{\pi_s^a}$  such that*

$$(2.4) \quad \Lambda(\pi) - D^2E(\pi) = O(\|\pi - \pi^*\|).$$

(b) *The approximate Newton flow from  $\Lambda(\pi)$  is*

$$(2.5) \quad \frac{d\pi_s^a}{dt} = \pi_s^a (r_s^a - \tau(\log \pi_s^a + 1)) - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau.$$

*With a learning rate  $\eta$ , the gradient update is*

$$(2.6) \quad \pi_s^a \leftarrow \frac{(\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s)/\tau)}{\sum_a (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s)/\tau)}.$$

*Remark 2.2.* The policy update scheme (2.6) is the same as the entropy regularized natural policy gradient scheme in, for example, [\[6\]](#).

*Historical note.* The natural gradient methods (including the NPG method) were traditionally developed as a way of implementing the vanilla gradient descent method with an intrinsic metric that is invariant to the choice of parameters (Cf. [\[16\]](#)), and entropy regularization was originally motivated as a way of encouraging exploration and avoid the suboptimality caused by greedy solvers (Cf. [\[22\]](#)). In this regard, it was more or less a coincidence that the algorithm combining the two methods – the regularized NPG obtains a fast quadratic convergence (Cf. [\[6\]](#)). The reason behind this coincidence is that the preconditioner used in the natural gradient method in fact approximates the second-order derivatives introduced by the entropy regularization in this case, though the fisher information matrix was not designed to approximate any second-order information in the classical natural gradient literature.

**2.2. Extension to other entropy functions.** [Theorem 2.1](#) can be extended to more general entropy functions. It yields brand-new algorithms with quadratic convergence. Here we consider the entropy functions of the form

$$(2.7) \quad (h_\pi)_s = \sum_a \phi\left(\frac{\pi_s^a}{\mu_s^a}\right) \mu_s^a,$$

where  $\phi$  is convex on  $(0, +\infty)$  and  $\phi(1) = 0$ , and  $\mu_s$  is a prior distribution over  $A$  such that  $\mu_s^a > 0$ . The term  $(h_\pi)_s$  is also called the “ $f$ -divergence” between  $\pi_s$  and  $\mu_s$  [24, 3]. If there is no prior knowledge of the policy, one can use the uniform prior, i.e.,  $\mu_s^a = 1/|A|$  for all  $a$ . We further assume that  $\phi$  is twice continuously differentiable and strongly convex and that  $\phi'(x) \rightarrow -\infty$  as  $x \rightarrow 0$ . Here are some examples:

- When  $\phi(x) = x \log x$ ,  $(h_\pi)_s = \sum_a \left( \frac{\pi_s^a}{\mu_s^a} \log \frac{\pi_s^a}{\mu_s^a} \right) \mu_s^a = \sum_a \pi_s^a \log \frac{\pi_s^a}{\mu_s^a}$ . When the uniform prior is used, we recover the negative Shannon entropy regularization  $\sum_a \pi_s^a \log \pi_s^a$  used in [Theorem 2.1](#) after omitting the constant  $\log \frac{1}{|A|}$ .
- When  $\phi(x) = \frac{4}{1-\alpha^2} (1 - x^{\frac{1+\alpha}{2}})$  ( $\alpha < 1$ ), we obtain the  $\alpha$ -divergence:

$$(2.8) \quad (h_\pi)_s = \frac{4}{1-\alpha^2} - \frac{4}{1-\alpha^2} \sum_a \mu_s^a (\pi_s^a / \mu_s^a)^{\frac{1+\alpha}{2}}.$$

In particular, when  $\alpha = 0$  we obtain the Hellinger divergence  $(h_\pi)_s = 2 - 2 \sum_a \sqrt{\mu_s^a \pi_s^a}$  after dividing by 2. When  $\alpha \rightarrow -1$  we obtain the reverse-KL divergence  $(h_\pi)_s = \sum_a \mu_s^a \log \frac{\mu_s^a}{\pi_s^a}$ . Also, when  $\alpha \rightarrow 1$ , we obtain the KL-divergence  $(h_\pi)_s = \sum_a \pi_s^a \log \frac{\pi_s^a}{\mu_s^a}$ , though the limit of  $\phi(x)$  does not exist when  $\alpha \rightarrow 1$ .

In the following theorem, we extend the approximate Newton method in [Theorem 2.1](#) to the entropy functions described above. The proof of this theorem can be found in [subsection 5.2](#).

**THEOREM 2.3.** *Assume that  $\pi^*$  is the optimizer of (1.5) where  $h_\pi$  is the entropy function defined in (2.7).*

(a) *The Hessian matrix  $D^2E(\pi)$  can be approximated by a diagonal matrix  $\Lambda(\pi)$  given by*

$$(2.9) \quad \Lambda_{(sa),(tb)} = -\tau \delta_{\{(sa),(tb)\}} \frac{(w_\pi)_s \phi''(\pi_s^a / \mu_s^a)}{\mu_s^a}$$

near  $\pi^*$  such that  $\Lambda(\pi) - D^2E(\pi) = O(\|\pi - \pi^*\|)$ .

(b) *The approximate Newton flow from  $\Lambda$  is*

$$(2.10) \quad \frac{d\pi_s^a}{dt} = \mu_s^a (\phi''(\pi_s^a / \mu_s^a))^{-1} (r_s^a - \tau \phi'(\pi_s^a / \mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s) / \tau.$$

With parameterization  $\theta_s^a = \phi'(\pi_s^a / \mu_s^a)$ , the approximate Newton method from  $\Lambda(\pi)$  can be expressed as:

$$(2.11) \quad \theta_s^a \leftarrow \eta (r_s^a - [(I - \gamma P^a)v_\pi]_s + c_s) / \tau + (1 - \eta) \theta_s^a.$$

where where  $0 < \eta \leq 1$  is the learning rate and  $c_s$  is a multiplier introduced by the constraint  $\sum_a \pi_s^a = 1$ .

For particular choices of  $\phi$ , the corresponding approximate Newton update scheme can be obtained directly by plugging  $\phi$  into (2.11).

- For the case  $\phi(x) = x \log x$  and  $(h_\pi)_s = \sum_a \pi_s^a \log \frac{\pi_s^a}{\mu_s^a}$ , one can solve the multipliers  $c_s$  explicitly as in [Theorem 2.1](#) and obtain the NPG method with prior distribution  $\mu$ :

$$(2.12) \quad \pi_s^a \leftarrow \frac{(\mu_s^a)^\eta (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s) / \tau)}{\sum_a (\mu_s^a)^\eta (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s) / \tau)}.$$

- For the case  $\phi(x) = \frac{4}{1-\alpha^2} (1 - x^{\frac{1+\alpha}{2}})$  ( $\alpha < 1$ ) and  $h_\pi$  given by the  $\alpha$ -divergence (2.8), we have  $\theta_s^a = -\frac{2}{1-\alpha} (\pi_s^a / \mu_s^a)^{\frac{\alpha-1}{2}}$ , thus by (2.11) the update scheme is

$$(2.13) \quad \pi_s^a \leftarrow \mu_s^a \left( (1 - \eta) (\pi_s^a / \mu_s^a)^{\frac{\alpha-1}{2}} + \frac{\alpha-1}{2} \eta (r_s^a - [(I - \gamma P^a)v_\pi]_s + c_s) / \tau \right)^{\frac{2}{\alpha-1}}.$$

The remaining problem in the update schemes is the determination of the multipliers  $c_s$ , since in general they cannot be solved explicitly as in the case of the negative Shannon entropy (Cf. [Theorem 2.1](#)). Since  $\phi$  is strongly convex, we know that  $\phi'$  is strictly increasing, and thus  $-\phi'$  is a strictly decreasing function mapping from  $(0, +\infty)$  to  $(-\sup \phi', +\infty)$  since  $\lim_{x \rightarrow 0+0} \phi'(x) = -\infty$ .

Let  $\psi := (-\phi')^{-1}$ , then  $\psi : (-\sup \phi', +\infty) \rightarrow (0, +\infty)$  is a strictly decreasing function that satisfies  $\lim_{x \rightarrow -\sup \phi'+0} \psi(x) = +\infty$  and  $\lim_{x \rightarrow +\infty} \psi(x) = 0$ . From [\(2.11\)](#), the equation of the multiplier  $c_s$  corresponding to  $\sum_a \pi_s^a = 1$  is:

$$(2.14) \quad \sum_a \mu_s^a \psi \left( -\frac{\eta}{\tau} c_s - (1-\eta) \phi' \left( \pi_s^a / \mu_s^a \right) - \frac{\eta}{\tau} (r_s^a - [(I - \gamma P^a) v_\pi]_s) \right) = 1,$$

or equivalently,

$$(2.15) \quad \sum_a \mu_s^a \psi (\tilde{c}_s + x_a) = 1,$$

where

$$(2.16) \quad \tilde{c}_s = -\frac{\eta}{\tau} c_s, \quad x_a = -(1-\eta) \phi' \left( \pi_s^a / \mu_s^a \right) - \frac{\eta}{\tau} (r_s^a - [(I - \gamma P^a) v_\pi]_s).$$

We claim that the determination of  $\tilde{c}_s$  in equation [\(2.15\)](#) (and thus the determination of  $c_s$ ) can be done in a similar way as in [\[39\]](#) based on the following lemma. The proof of this lemma can be found in [subsection 5.3](#).

**LEMMA 2.4.** *Let  $L \in \mathbb{R} \cup \{-\infty\}$ . Assume that  $\psi : (L, +\infty) \rightarrow (0, +\infty)$  is a strictly decreasing function that satisfies  $\lim_{x \rightarrow L+0} \psi(x) = +\infty$  and  $\lim_{x \rightarrow +\infty} \psi(x) = 0$  and assume also that  $\mu_i > 0$ , then for any  $x_1, x_2, \dots, x_k$ , there is a unique solution to the equation:*

$$(2.17) \quad \mu_1 \psi(x + x_1) + \dots + \mu_k \psi(x + x_k) = 1.$$

Moreover, the solution is on the interval

$$(2.18) \quad \left[ \max \left\{ L - \min_{1 \leq i \leq k} x_i, \min_{1 \leq i \leq k} \left\{ \psi^{-1} \left( \frac{1}{k \mu_i} \right) - x_i \right\} \right\}, \max_{1 \leq i \leq k} \left\{ \psi^{-1} \left( \frac{1}{k \mu_i} \right) - x_i \right\} \right].$$

Leveraging [Lemma 2.4](#) and the monotonicity of the function  $\mu_1 \psi(x + x_1) + \dots + \mu_k \psi(x + x_k) - 1$ , many of the established numerical methods (e.g. bisection) for nonlinear equations can be applied to determine the solution for [\(2.17\)](#). This routine can be used to find  $\tilde{c}_s$  in [\(2.15\)](#) and thus the multipliers  $c_s$  in [\(2.14\)](#) as stated in [Proposition 2.5](#) whose proof is given in [subsection 5.4](#).

**PROPOSITION 2.5.** *The multipliers  $c_s$  in the update scheme [\(2.11\)](#) can be determined uniquely such that the updated policy  $\pi$  satisfies  $\pi_s^a \geq 0$  and  $\sum_a \pi_s^a = 1$ .*

When the  $\alpha$ -divergence is used, we have  $\phi = \frac{4}{1-\alpha^2} (1 - x^{\frac{1+\alpha}{2}})$  and  $\phi'(x) = \frac{2}{\alpha-1} x^{\frac{\alpha-1}{2}}$ , then  $L = -\sup \phi' = 0$  and  $\psi(x) = (-\phi')^{-1}(x) = (\phi')^{-1}(-x) = (\frac{1-\alpha}{2} x)^{\frac{2}{\alpha-1}}$ . The algorithm proposed in this section is summarized in [Algorithm 2.1](#) below.

**2.3. Convergence of the approximate Newton gradient flow.** Recall from [\(2.10\)](#) that the approximate Newton gradient flow with the general entropy functions is

$$\frac{d\pi_s^a}{dt} = \mu_s^a (\phi''(\pi_s^a / \mu_s^a))^{-1} (r_s^a - \tau \phi'(\pi_s^a / \mu_s^a) - [(I - \gamma P^a) v_\pi]_s + c_s) / \tau,$$

**Algorithm 2.1** Approximate Newton method for the regularized MDP

**Require:** the MDP model  $\mathcal{M} = (S, A, P, r, \gamma)$ , initial policy  $\pi_{\text{init}}$ , convergence threshold  $\epsilon_{\text{tol}}$ , regularization coefficient  $\tau$ , learning rate  $\eta$ , the regularization type (KL or  $\alpha$ -divergence).

- 1: Initialize the policy  $\pi = \pi_{\text{init}}$ .
- 2: Set  $\xi = 1 + \epsilon_{\text{tol}}$  and  $k = |A|$ .
- 3: **while**  $\xi > \epsilon_{\text{tol}}$  **do**
- 4:   Calculate the regularization term  $h_\pi$  by  $(h_\pi)_s = \sum_a \mu_s^a \phi(\pi_s^a / \mu_s^a)$ .
- 5:   Calculate  $P_\pi$  and  $r_\pi$  by  $(P_\pi)_{st} = \sum_a \pi_s^a P_{st}^a$ ,  $(r_\pi)_s = \sum_a \pi_s^a r_s^a$ .
- 6:   Calculate  $v_\pi$  by (1.4), i.e.,  $v_\pi = (I - \gamma P_\pi)^{-1}(r_\pi - \tau h_\pi)$ .
- 7:   **if** the KL divergence is used **then**
- 8:      $(\pi_{\text{new}})_s^a \leftarrow \frac{(\mu_s^a)^\eta (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s) / \tau)}{\sum_a (\mu_s^a)^\eta (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s) / \tau)}$  for  $a = 1, 2, \dots, |A|$ ,  $s = 1, 2, \dots, |S|$ .
- 9:   **end if**
- 10:   **if** the  $\alpha$ -divergence is used **then**
- 11:     **for**  $s = 1, 2, \dots, |S|$  **do**
- 12:       Set  $L = 0$  and  $\psi(x) = (\frac{1-\alpha}{2}x)^{\frac{2}{\alpha-1}}$
- 13:       Calculate  $x_a = -(1-\eta)\phi'(\pi_s^a / \mu_s^a) - \frac{\eta}{\tau}(r_s^a - [(I - \gamma P^a)v_\pi]_s)$ ,  $a = 1, \dots, |A|$ .
- 14:       Solve for  $\tilde{c}_s = -\frac{\eta}{\tau}c_s$  with the bisection method on the interval described in (2.18).
- 15:       Update  $(\pi_{\text{new}})_s^a \leftarrow \mu_s^a \psi(\tilde{c}_s + x_a)$  for  $a = 1, 2, \dots, |A|$ .
- 16:     **end for**
- 17:   **end if**
- 18:    $\xi = \|\pi_{\text{new}} - \pi\|_F / \|\pi\|_F$ .
- 19:    $\pi = \pi_{\text{new}}$
- 20: **end while**

from which we can obtain the dynamics of the objective function  $E$ :

$$\begin{aligned}
\frac{dE(\pi)}{dt} &= \sum_{sa} \frac{\partial E}{\partial \pi_s^a} \frac{d\pi_s^a}{dt} \\
&= \sum_{sa} \left[ (r_s^a - \tau \phi'(\pi_s^a / \mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s)(w_\pi)_s \right. \\
(2.19) \quad &\quad \left. \cdot \mu_s^a (\phi''(\pi_s^a / \mu_s^a))^{-1} (r_s^a - \tau \phi'(\pi_s^a / \mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s) / \tau \right] \\
&= \sum_{sa} \mu_s^a (\tau \phi''(\pi_s^a / \mu_s^a))^{-1} (r_s^a - \tau \phi'(\pi_s^a / \mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s)^2 (w_\pi)_s \\
&\geq 0,
\end{aligned}$$

where we have used the gradient

$$(2.20) \quad \frac{\partial E}{\partial \pi_s^a} = (r_s^a - \tau \phi'(\pi_s^a / \mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s)(w_\pi)_s.$$

As a result, we have shown that  $\frac{dE(\pi)}{dt} \geq 0$ . Since  $E(\pi)$  is upper-bounded by  $\rho^\top v_{\pi^*}$ ,  $E(\pi)$  converges. With a closer look, the following theorem states that the limiting policy is exactly the optimal policy  $\pi^*$  and the proof is given in [subsection 5.5](#).

**THEOREM 2.6.** *The approximate Newton flow (2.10) converges globally to the optimal policy  $\pi^*$ .*

**3. Quadratic Convergence.** In this section, we study the quadratic convergence of the approximate Newton method at the learning rate  $\eta = 1$ , which corresponds to the step size used in the Newton method. Our analysis is inspired by the results in [8, 37]. The following theorem states the second-order convergence when  $\eta = 1$ , with the proof given in [subsection 5.6](#). For the

simplicity of notations, we let  $f(\pi)_{sa} = -(r_s^a - ((I - \gamma P^a)v_\pi)_s)$ , which is the additive inverse of the advantage function.

THEOREM 3.1. *Let*

$$\Phi(\pi) = \tau \sum_{sa} \mu_s^a \phi \left( \frac{\pi_s^a}{\mu_s^a} \right),$$

where  $\phi$  is twice continuously differentiable and strongly convex and that  $\phi'(x) \rightarrow -\infty$  as  $x \rightarrow 0$ . Denote the  $k$ -th policy obtained in [Algorithm 2.1](#) by  $\pi^{(k)}$ . For  $\eta = 1$ , the update scheme in [Algorithm 2.1](#) can be summarized as

$$(3.1) \quad \nabla \Phi(\pi^{(k+1)}) - \nabla \Phi(\pi^{(k)}) = - \left( f(\pi^{(k)}) + \nabla \Phi(\pi^{(k)}) - B^\top c(\pi^{(k)}) \right),$$

where  $f(\pi)_{sa} = -(r_s^a - ((I - \gamma P^a)v_\pi)_s)$  and we denote by  $B$  the  $|S|$ -by- $(|S| \times |A|)$  matrix such that  $B_{ij} = 1$  for  $|A|(i-1) + 1 \leq j \leq |A|i$  and  $B_{ij} = 0$  otherwise. Then  $\pi^{(k)}$  enjoys a quadratic local convergence to  $\pi^*$ , i.e.,  $\lim_{k \rightarrow \infty} \pi^{(k)} = \pi^*$  and

$$(3.2) \quad \left\| \pi^{(k+1)} - \pi^* \right\| \leq C \left\| \pi^{(k)} - \pi^* \right\|^2,$$

for some constant  $C$ , given that the initial policy  $\pi^{(0)}$  is sufficiently close to  $\pi^*$ .

*Remark 3.2.* It is clear that the quadratic convergence also occurs if  $\pi^{(k)}$  is in a sufficiently small neighborhood of  $\pi^*$  for some  $k \geq 1$  even if  $\pi^{(0)}$  is not. The precise description of this small neighborhood is provided in the proof (see [Subsection 5.6](#)). For a special case of this result, where  $\phi(x) = x \log x$  and  $\mu_s^a = 1/|A|$ , the algorithm is reduced to the entropy regularized NPG. A similar local convergence result for this special case has been obtained in [\[6\]](#), where the proof leverages the particular structure of Shannon entropy.

*Connection with mirror descent.* The approximate Newton algorithm [\(3.1\)](#) for  $\eta = 1$  has a deep connection with mirror descent. The vanilla mirror descent of  $-E(\pi)$  with a learning rate  $\beta$  and the Bregman divergence associated with  $\Phi$  is given by

$$\begin{aligned} \pi^{(k+1)} &= \arg \min_{\pi} \left\{ -E(\pi^{(k)}) - \nabla E(\pi^{(k)})(\pi - \pi^{(k)}) + \frac{1}{\beta} (\Phi(\pi) - \Phi(\pi^{(k)}) - \nabla \Phi(\pi^{(k)})(\pi - \pi^{(k)})) \right\} \\ &= \arg \min_{\pi} \left\{ (\text{diag}(w_{\pi^{(k)}}) \otimes I_{|A|})(f(\pi^{(k)}) + \nabla \Phi(\pi^{(k)}))(\pi - \pi^{(k)}) + \frac{1}{\beta} (\Phi(\pi) - \nabla \Phi(\pi^{(k)})\pi) \right\}, \end{aligned}$$

where  $\text{diag}(w_{\pi^{(k)}})$  is the diagonal matrix with the diagonal equal to  $w_{\pi^{(k)}} := (I - \gamma P_{\pi^{(k)}}^\top)^{-1} \rho$ ,  $\otimes$  denotes the Kronecker product, and  $I_{|A|}$  denotes the  $|A|$  by  $|A|$  identity matrix. In the last equality, the terms independent of  $\pi$  are dropped and the multiplier term in  $\nabla E$  is canceled out using  $B\pi = B\pi^{(k)} = \mathbf{1}_{|S|}$ . The first-order stationary condition of this minimization problem reads

$$(3.3) \quad \nabla \Phi(\pi^{(k+1)}) - \nabla \Phi(\pi^{(k)}) = -\beta (\text{diag}(w_{\pi^{(k)}}) \otimes I_{|A|})(f(\pi^{(k)}) + \nabla \Phi(\pi^{(k)}) - B^\top c(\pi^{(k)})).$$

This suggests that [\(3.1\)](#) can be reinterpreted as an *accelerated* mirror descent method with *adaptive* learning rates  $\beta_s \equiv 1/(w_{\pi^{(k)}})_s$  that depend on the state  $s$  and the current iterate  $\pi^{(k)}$ . Observation of the connection between mirror descent and the natural gradient method (which is similar with the approximate Newton method in this paper when the Shannon entropy is used) is given in [\[23, 10\]](#).

In [\[40\]](#), a variant of mirror descent is proposed based on an implicit update scheme

$$(3.4) \quad (\nabla \Phi(\pi^{(k+1)}))_{sa} - (\nabla \Phi(\pi^{(k)}))_{sa} = -\beta' \left( f(\pi^{(k)})_{sa} + \nabla \Phi(\pi^{(k+1)})_{sa} - (c(\pi^{(k)}))_s \right),$$

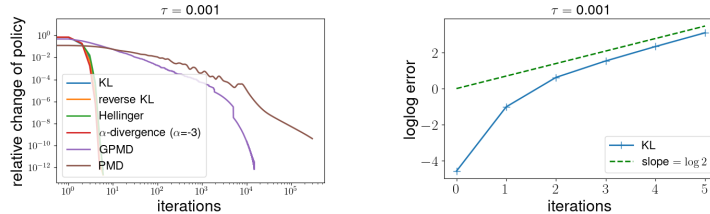
with a state independent learning rate  $\beta'$ . In the next section, we will compare this variant with our approximate Newton method [\(3.1\)](#) and show that the approximate Newton method converges orders of magnitudes faster than the ones in [\[40\]](#).



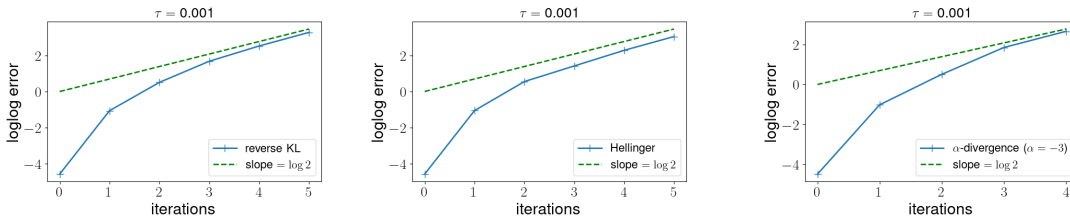
4. Numerical results.

4.1. **Experiment I.** We first test the approximate Newton methods derived in section 2 on the model in [40]. For the sake of completeness, we include the description of the model here. The MDP considered has a state space  $S$  of size 200 and an action space  $A$  of size 50. For each state  $t$  and action  $a$ , a subset  $S_t^a$  of  $S$  is uniformly randomly chosen such that  $|S_t^a| = 20$ , and  $P_{tt'}^a = 1/20$  for any  $t' \in S_t^a$ . The reward is given by  $r_s^a = U_s^a U_s$ , where  $U_s^a$  and  $U_s$  are independently uniformly chosen on  $[0, 1]$ . The discount rate  $\gamma$  is set as 0.99 and the regularization coefficient  $\tau = 0.001$ .

In the numerical experiment, we implement Algorithm 2.1 with the KL divergence, the reverse KL divergence, the Hellinger divergence, and the  $\alpha$ -divergence with  $\alpha = -3$ . We adopt the uniform prior  $\mu_s^a = 1/|A|$  in order to make a fair comparison with the policy mirror descent (PMD) and the general policy mirror descent (GPMD) method in [40]. We set the initial policy as the uniform policy, the convergence threshold as  $\epsilon_{\text{tol}} = 10^{-12}$ , and the learning rate  $\eta$  as 1. Figure 1(a) demonstrates that, for these four tests, the approximate Newton algorithm converges in 7, 7, 7, and 6 iterations, respectively. In comparison, we apply PMD and GPMD to the same MDP with the same stopping criterion. As also shown in Figure 1(a), many more iterations are needed for GPMD and PMD to reach the same precision: GPMD converges in 14822 iterations, and PMD does not reach the desired precision after  $3 \times 10^5$  iterations. For the implementation of GPMD and PMD, a quadratic regularization is used and we have already tuned the hyperparameters to optimize their performance. The number of iterations needed for GPMD and PMD to converge accords with the numerical results provided in [40].



(a) Relative change of the policy using Algorithm 2.1 and methods from [40]. (b) The policy error in the process of training using KL-divergence.



(c) The policy error in the process of training using reverse KL-divergence. (d) The policy error in the process of training using Hellinger divergence. (e) The policy error in the process of training using  $\alpha$ -divergence with  $\alpha = -3$ .

FIG. 1. Figures for the synthetic medium scale MDP. (a): Relative change of the policy  $\|\pi_{\text{new}} - \pi\|_F / \|\pi\|_F$  during training of Algorithm 2.1 compared with PMD and GPMD in [40], with the logarithmic scale used for both axes. Notice that Algorithm 2.1 converges in 6-7 iterations to  $10^{-12}$  in all cases while PMD and GPMD take more than  $10^4$  iterations. Here the quadratic regularization is used for PMD and GPMD. (b) - (e): Blue: The convergence of  $\log \|\log \|\pi - \pi^*\|_F\|$  in the training process with the KL divergence, the reverse KL divergence, the Hellinger divergence, and the  $\alpha$ -divergence with  $\alpha = -3$ , respectively. Green: A line through the origin with slope  $\log 2$ . Comparison of the convergence plots with the green reference lines shows a clear quadratic convergence for Algorithm 2.1.

In order to verify the quadratic convergence proved in section 3, we draw the plots of  $\log \|\log \|\pi - \pi^*\|_F\|$  in Figure 1(b), Figure 1(c), Figure 1(d) and Figure 1(e), where  $\pi^*$  is the final

policy and the norm used is the Frobenius norm. A green reference line with slope  $\log 2$  through the origin is plotted for comparison. If the error converges exactly at a quadratic rate, the plot of  $\log \|\log \|\pi - \pi^*\|\|$  shall be parallel to the reference line. The convergence curves approach the reference lines at the end (and are even steeper than the reference lines in the beginning), demonstrating clearly a quadratic convergence for all forms of regularization used here.

**4.2. Experiment II.** Next, we apply the approximate Newton methods derived in [section 2](#) to an MDP model constructed from the search logs of an online shopping store, with two different ranking strategies. Each issued query is represented as a state in the MDP. In response to a query, the search can be done by choosing one of the two ranking strategies (actions) to return a ranked list of products shown to the customer. Based on the shown products, the customer can refine or update the query, thus entering a new state. The reward at each state-action pair is a weighted sum of the clicks and purchases resulting from the action. Based on the data collected from two separate 5-week periods for both ranking strategies, we construct an MDP with 135k states and a very sparse transition tensor  $P$  with only 0.01% nonzero entries. The discount rate  $\gamma$  is set as 0.99 and the regularization coefficient is  $\tau = 0.001$ . In the implementation, we use the uniform prior  $\mu_s^a = 1/|A|$ .

When calculating  $v_\pi$  by  $v_\pi = (I - \gamma P_\pi)^{-1}(r_\pi - \tau h_\pi)$ , we apply the iterative solver Bi-CGSTAB [\[36\]](#), a widely used numerical method with high efficiency and robustness for solving large sparse non-symmetric systems of linear equations [\[27, 7\]](#), in order to leverage the sparsity of the transition tensor.

In the numerical experiment, we implement [Algorithm 2.1](#) with the KL divergence, the reverse KL divergence, the Hellinger divergence, and the  $\alpha$ -divergence with  $\alpha = -3$ . We set the initial policy as the uniform policy, the convergence threshold as  $\epsilon_{\text{tol}} = 10^{-12}$ , and the learning rate  $\eta$  as 1. All the tests end up with fast convergence as shown in [Figure 2\(a\)](#), where logarithmic scale is used for the vertical axis. More specifically, the approximate Newton algorithm using the KL divergence, the reverse KL divergence, the Hellinger divergence, and the  $\alpha$ -divergence with  $\alpha = -3$  converge in 6, 6, 6, 5 iterations, respectively. It is worth noticing that even though the size of the state space  $S$  here is some magnitudes larger than the examples in [subsection 4.1](#), the number of approximate Newton iterations used is about the same. The comparison with GPMD and PMD is not given for this example since they are intractable to implement due to the high computational cost caused by the large MDP model.

In [Table 1](#), we report the number of BiCGSTAB steps used in the algorithm. In each approximate Newton iteration, less than 20 BiCGSTAB steps are used in order to find  $v_\pi$ . For all four regularizers used here, altogether only about 100 BiCGSTAB steps are needed in the whole training process, thanks to the fast convergence of the approximate Newton method. As a comparison, the regularized value iteration (a special case for the method in [\[9\]](#)) typically needs thousands of matrix-vector multiplication with the MDP transition matrix, since its convergence rate is  $O(\gamma^T)$ .

Regularizer	KL	reverse-KL	Hellinger	$\alpha$ -divergence ( $\alpha = -3$ )
Approx-Newton Iterations	6	6	6	5
Total Bi-CGSTAB steps	110	109	110	83
Average Bi-CGSTAB steps	18.3	18.2	18.3	16.6

TABLE 1

*Number of approximate Newton iterations and BiCGSTAB steps used in the training process.*

As in the previous numerical example, in [Figure 2\(b\)](#), [Figure 2\(c\)](#), [Figure 2\(d\)](#) and [Figure 2\(e\)](#) we verify the quadratic convergence by comparing the plot of  $\log \|\log \|\pi - \pi^*\|\|$  with a green reference line through the origin with slope  $\log 2$ . As the convergence curves are approximately parallel to the reference lines, this verifies that the proposed algorithm converges quadratically with all the regularizations in this example as well.

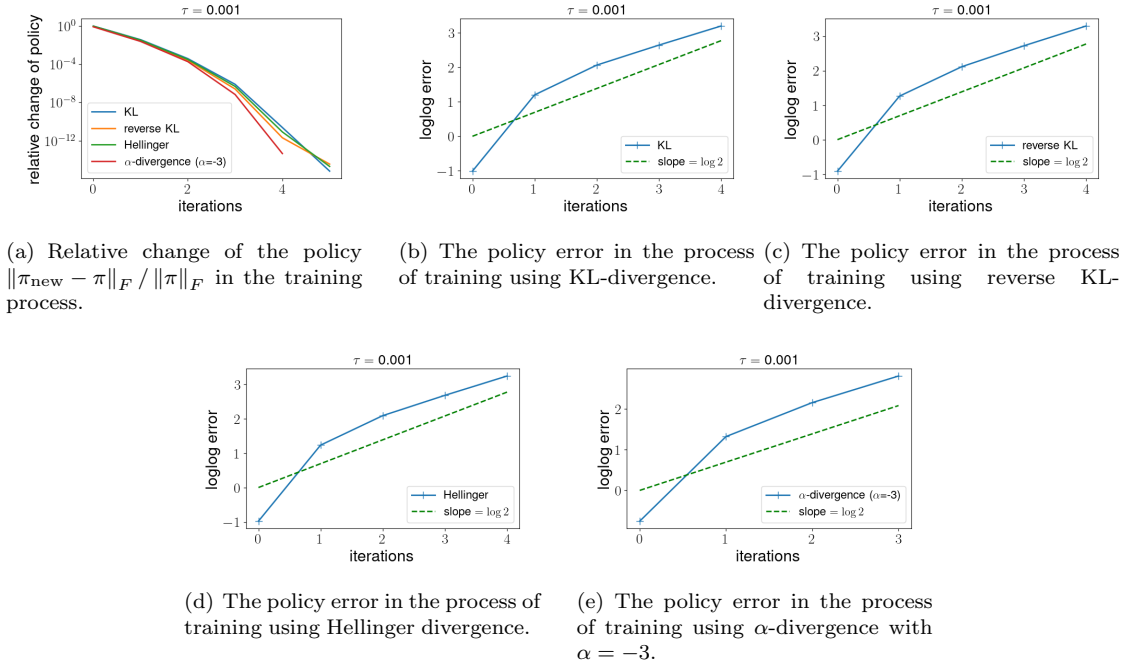


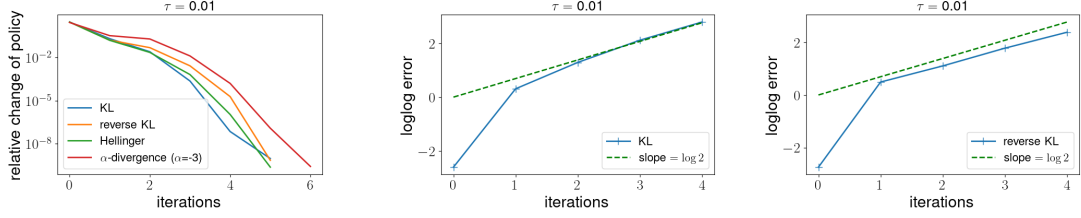
FIG. 2. Figures for the industrial-size MDP. (a): Relative change of the policy  $\|\pi_{\text{new}} - \pi\|_F / \|\pi\|_F$  in the training process of Algorithm 2.1. A logarithmic scale is used for the vertical axis. (b) - (e): Blue: The convergence of  $\log |\log \|\pi - \pi^*\|_F|$  in the training process with the KL divergence, the reverse KL divergence, the Hellinger divergence and the  $\alpha$ -divergence with  $\alpha = -3$ , respectively. Green: A line through the origin with slope  $\log 2$ .

**4.3. Experiment III.** In this section, we are concerned with an MDP with relatively large action space and state space at the same time. We consider the state space and action space with size  $|S| = 10000$  and  $|A| = 300$  with  $(S, A) = (\{0, 1, \dots, |S| - 1\}, \{0, 1, \dots, |A| - 1\})$ . Here the transition tensor is defined as  $P_{tt'}^a = 1$  when  $t' = (t + a) \bmod |S|, t \neq |S| - 1$  or  $t = t' = |S| - 1$ , and  $P_{tt'}^a = 0$  otherwise. The reward is set as  $r_s^a = 1 - \gamma$  if  $s = |S| - 1$  and  $r_s^a = 0$  otherwise, where  $\gamma = 0.99$ .

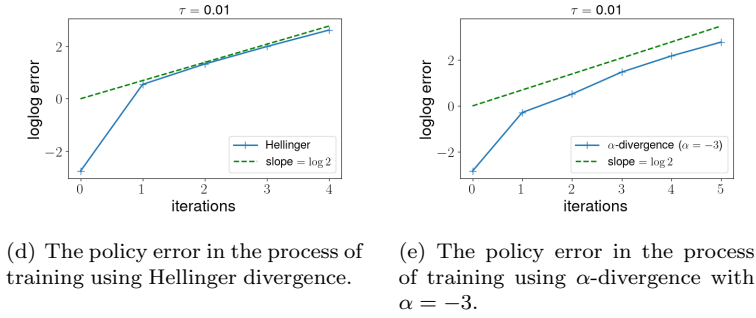
Similar to the previous tests, we apply the approximate Newton algorithm with the KL divergence, the reverse KL divergence, the Hellinger divergence, and the  $\alpha$ -divergence with  $\alpha = -3$  and the uniform prior  $\mu_s^a = 1/|A|$ . For this experiment, we use  $\tau = 0.01$  and  $\epsilon_{\text{tol}} = 10^{-9}$ . Similar to the previous examples, in all 4 tests the algorithm converges with single-digit approximate Newton iterations, as shown in Figure 3(a). The quadratic convergence can be verified in the plots of  $\log |\log \|\pi - \pi^*\|_F|$  displayed in Figure 3(b), Figure 3(c), Figure 3(d) and Figure 3(e). Due to the size and sparsity of the transition tensor, we also adopt Bi-CGSTAB for calculating  $v_\pi$ , and the number of Bi-CGSTAB iterations used is reported in Table 2. Around 400 Bi-CGSTAB steps are used, which also involves fewer matrix-vector multiplication with the transition matrix compared to traditional value iteration methods.

Regularizer	KL	reverse-KL	Hellinger	$\alpha$ -divergence ( $\alpha = -3$ )
Approx-Newton Iterations	6	6	6	7
Total Bi-CGSTAB steps	370	379	492	452
Average Bi-CGSTAB steps	61.7	63.2	82.0	64.6

TABLE 2  
Number of approximate Newton iterations and BiCGSTAB steps used in the training process.



(a) Relative change of the policy  $\|\pi_{\text{new}} - \pi\|_F / \|\pi\|_F$  in the training process. (b) The policy error in the process of training using KL-divergence. (c) The policy error in the process of training using reverse KL-divergence.



(d) The policy error in the process of training using Hellinger divergence. (e) The policy error in the process of training using  $\alpha$ -divergence with  $\alpha = -3$ .

FIG. 3. Figures for an MDP with both the state space and action space relatively large. (a): Relative change of the policy  $\|\pi_{\text{new}} - \pi\|_F / \|\pi\|_F$  in the training process of Algorithm 2.1. A logarithmic scale is used for the vertical axis. (b) - (e): Blue: The convergence of  $\log \|\log \|\pi - \pi^*\|_F\|$  in the training process with the KL divergence, the reverse KL divergence, the Hellinger divergence and the  $\alpha$ -divergence with  $\alpha = -3$ , respectively. Green: A line through the origin with slope  $\log 2$ .

## 5. Proofs.

### 5.1. Proof of Theorem 2.1.

*Proof. Step 1: expand  $E(\pi)$  and prove the first-order condition (5.2).* For any  $\epsilon \in \mathbb{R}^{|S| \times |A|}$ , introduce  $r_\epsilon \in \mathbb{R}^{|S|}$  and  $Z_\epsilon \in \mathbb{R}^{|S| \times |S|}$  such that

$$(5.1) \quad (r_\epsilon)_s := \sum_a \epsilon_s^a r_s^a, \quad (Z_\epsilon)_{st} := \sum_a \epsilon_s^a (\delta_{st} - \gamma P_{st}^a),$$

where  $\delta_{st} = 1$  if  $s = t$  and  $\delta_{st} = 0$  otherwise. Then  $r_\epsilon$  and  $Z_\epsilon$  are linear with respect to  $\epsilon$ , which is helpful when expressing the first-order conditions and simplifying the expansion of  $E(\pi)$ .

Now we proceed to prove that for any  $\epsilon$  with  $\sum_a \epsilon_s^a = 0$  and  $|\epsilon_s^a| < \pi_s^a$ , at  $\pi = \pi^*$

$$(5.2) \quad r_\epsilon - \tau Dh_\pi \epsilon - Z_\epsilon Z_\pi^{-1} (r_\pi - \tau h_\pi) = 0,$$

where  $Dh_\pi \in \mathbb{R}^{|S| \times |S| \times |A|}$  is the gradient matrix of  $h_\pi$  with respect to  $\pi$ .

Since  $\pi$  is a policy,  $\sum_a \pi_s^a = 1$  for any  $s$ . Thus

$$(5.3) \quad (Z_\pi)_{st} = \delta_{st} - \gamma \sum_a \pi_s^a P_{st}^a = \sum_a \pi_s^a (\delta_{st} - \gamma P_{st}^a).$$

Now consider a policy  $\pi + \epsilon$ , i.e.,  $\sum_a \epsilon_s^a = 0$  and  $\pi_s^a + \epsilon_s^a \geq 0$ , then thanks to (5.3) one can obtain:

$$(5.4) \quad Z_{\pi+\epsilon} = Z_\pi + Z_\epsilon, \quad r_{\pi+\epsilon} = r_\pi + r_\epsilon,$$

where  $Z_\epsilon$  and  $r_\epsilon$  are defined in (5.1), i.e.,  $(Z_\epsilon)_{st} = \sum_a \epsilon_s^a (\delta_{st} - \gamma P_{st}^a)$ .  $(r_\epsilon)_s = \sum_a \epsilon_s^a r_s^a$ . Leveraging

the linearity (5.4), we obtain the expansion:

$$\begin{aligned}
(5.5) \quad E(\pi + \epsilon) &= \rho^\top Z_{\pi+\epsilon}^{-1}(r_{\pi+\epsilon} - \tau h_{\pi+\epsilon}) = \rho^\top (Z_\pi + Z_\epsilon)^{-1}(r_\pi + r_\epsilon - \tau h_{\pi+\epsilon}) \\
&= \rho^\top Z_\pi^{-1}(I - Z_\epsilon Z_\pi^{-1} + Z_\epsilon Z_\pi^{-1} Z_\epsilon Z_\pi^{-1})(r_\pi + r_\epsilon - \tau h_\pi - \tau D h_\pi \epsilon - \frac{1}{2} \epsilon^\top \tau D^2 h_\pi \epsilon) + O(\|\epsilon\|^3) \\
&= E(\pi) + w_\pi^\top [-Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi) + (r_\epsilon - \tau D h_\pi \epsilon)] + w_\pi^\top (-\frac{1}{2} \epsilon^\top \tau D^2 h_\pi \epsilon) \\
&\quad + w_\pi^\top [-Z_\epsilon Z_\pi^{-1}(r_\epsilon - \tau D h_\pi \epsilon) + Z_\epsilon Z_\pi^{-1} Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi)] + O(\|\epsilon\|^3),
\end{aligned}$$

where  $Dh_\pi$  is a second-order tensor that maps from  $S \times A$  to  $S$ , and  $D^2h_\pi$  is a third-order tensor that maps from  $(S \times A)^{\otimes 2}$  to  $S$ . With this expansion, one can see that

$$\frac{\partial E}{\partial \pi_s^a} = (r_s^a - \tau(\log \pi_s^a + 1) - [(I - \gamma P^a)v_\pi]_s + c_s)(w_\pi)_s,$$

where  $c_s$  is a multiplier that only depends on  $s$ . Then at  $\pi = \pi^*$ ,

$$\frac{\partial E}{\partial \pi_s^a} = (r_s^a - \tau(\log \pi_s^a + 1) - [(I - \gamma P^a)v_\pi]_s + c_s)(w_\pi)_s = 0.$$

Since  $w_\pi = (I - \gamma P_\pi^\top)^{-1} \rho = \rho + \sum_{i=1}^{\infty} \gamma^i (P_\pi^\top)^i \rho$  and all elements of  $\rho$  are positive, we also know that all elements of  $w_\pi$  are positive. Thus at  $\pi = \pi^*$ ,

$$r_s^a - \tau(\log \pi_s^a + 1) - [(I - \gamma P^a)v_\pi]_s + c_s = 0.$$

Multiplying the left hand side with  $\epsilon_s^a$  and taking the sum over  $a$ , we obtain:

$$(r_\epsilon - \tau D h_\pi \epsilon - Z_\epsilon v_\pi)_s + c_s \sum_a \epsilon_s^a = 0, \quad \forall s, \quad \forall \epsilon.$$

Since  $\sum_a \epsilon_s^a = 0$  for any  $s$  and  $v_\pi = Z_\pi^{-1}(r_\pi - \tau h_\pi)$ , we have

$$r_\epsilon - \tau D h_\pi \epsilon - Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi) = 0, \quad \forall \epsilon,$$

at  $\pi = \pi^*$ , which proves (5.2).

**Step 2: Derive the decomposition (2.3) with the obtained expansion and first-order condition.** With (5.2), one can approximate the second-order term in (5.5) for  $\pi$  near  $\pi^*$ :

$$\begin{aligned}
&w_\pi^\top (-\frac{1}{2} \epsilon^\top \tau D^2 h_\pi \epsilon) + w_\pi^\top [-Z_\epsilon Z_\pi^{-1}(r_\epsilon - \tau D h_\pi \epsilon) + Z_\epsilon Z_\pi^{-1} Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi)] \\
&= \frac{1}{2} \epsilon^\top \Lambda(\pi) \epsilon - w_\pi^\top Z_\epsilon Z_\pi^{-1} (r_\epsilon - \tau D h_\pi \epsilon - Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi)) \\
&\approx \frac{1}{2} \epsilon^\top \Lambda(\pi) \epsilon.
\end{aligned}$$

By (5.2) and that  $h$  is twice continuously differentiable, the approximate Hessian  $\Lambda$  converge to the true Hessian as  $\pi$  converges to  $\pi^*$ , and their difference  $\Lambda(\pi) - D^2 E(\pi) = O(\|\pi - \pi^*\|)$ . Hence, the second-order derivatives of  $E(\pi)$  can be approximated by

$$(5.6) \quad \frac{\partial^2 E}{\partial \pi_s^a \partial \pi_t^b} \approx \Lambda_{(sa),(tb)} = -\tau \delta_{\{(sa),(tb)\}} \frac{(w_\pi)_s}{\pi_s^a},$$

from which we have shown that  $\Lambda$  is diagonal.

**Step 3: Derive the approximate Newton flow and the policy update scheme with the obtained decomposition.** Using this approximate second-order derivative as a preconditioner,  $w_\pi$  is canceled out in the policy gradient algorithm, which yields the gradient flow:

$$\frac{d\pi_s^a}{dt} = \pi_s^a (r_s^a - \tau(\log \pi_s^a + 1) - [(I - \gamma P^a)v_\pi]_s + c_s) / \tau.$$

Adopting the parameterization  $\pi_s^a = \exp(\theta_s^a)$ , we have

$$(5.7) \quad \frac{d\theta_s^a}{dt} = (r_s^a - \tau(\theta_s^a + 1) - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau.$$

With a learning rate  $\eta$ , this becomes

$$(5.8) \quad \theta_s^a \leftarrow \eta(r_s^a - \tau - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau + (1 - \eta)\theta_s^a,$$

which corresponds to

$$(5.9) \quad \pi_s^a \leftarrow (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a - \tau - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau),$$

and  $c_s$  is determined by the condition that  $\sum_a \pi_s^a = 1$ . Equivalently, we have

$$\pi_s^a \leftarrow \frac{(\pi_s^a)^{1-\eta} \exp(\eta(r_s^a - \tau - [(I - \gamma P^a)v_\pi]_s)/\tau)}{\sum_a (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a - \tau - [(I - \gamma P^a)v_\pi]_s)/\tau)} = \frac{(\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s)/\tau)}{\sum_a (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v_\pi)_s)/\tau)},$$

where we cancel out the factors independent of  $a$  and obtain (2.6). This finishes the proof.  $\square$

### 5.2. Proof of Theorem 2.3.

*Proof.* Similar with (5.2), we first prove that for any  $\epsilon$  with  $\sum_a \epsilon_s^a = 0$  and  $|\epsilon_s^a| < \pi_s^a$ , at  $\pi = \pi^*$

$$(5.10) \quad r_\epsilon - \tau Dh_\pi \epsilon - Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi) = 0,$$

Similar to the proof of Theorem 2.1, by direct calculations one can get:

$$(5.11) \quad \frac{\partial E}{\partial \pi_s^a} = (r_s^a - \tau \phi'(\pi_s^a/\mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s)(w_\pi)_s,$$

where  $c_s$  is a multiplier that only depends on  $s$ . Since all elements of  $w_\pi$  are positive, at  $\pi = \pi^*$ ,

$$(5.12) \quad (r_s^a - \tau \phi'(\pi_s^a/\mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s) = 0.$$

By multiplying (5.12) with  $\epsilon_s^a$  and summing over  $a$ , one can obtain:

$$r_\epsilon - \tau Dh_\pi \epsilon - Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi) = 0, \quad \forall \epsilon,$$

at  $\pi = \pi^*$ , which proves (5.10). Since the only difference between the functional  $E(\pi)$  defined here and the  $E(\pi)$  in Theorem 2.1 lies in the regularizer  $h$ , one can still obtain the expansion:

$$\begin{aligned} & E(\pi + \epsilon) - E(\pi) - w_\pi^\top [-Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi) + (r_\epsilon - \tau Dh_\pi \epsilon)] \\ &= w_\pi^\top \left( -\frac{1}{2} \epsilon^\top \tau D^2 h_\pi \epsilon \right) - w_\pi^\top Z_\epsilon Z_\pi^{-1} (r_\epsilon - \tau Dh_\pi \epsilon - Z_\epsilon Z_\pi^{-1}(r_\pi - \tau h_\pi)) + O(\|\epsilon\|^3) \\ &= \frac{1}{2} \epsilon^\top \Lambda(\pi) \epsilon + O(\|\epsilon\|^2 \|\pi - \pi^*\|) + O(\|\epsilon\|^3). \end{aligned}$$

Hence we have  $D^2 E(\pi) - \Lambda(\pi) = O(\|\pi - \pi^*\|)$ . Using this expansion, one can derive an approximation for the second-order derivatives:

$$\frac{\partial^2 E}{\partial \pi_s^a \partial \pi_t^b} \approx \Lambda_{(sa),(tb)} = -\tau \delta_{\{(sa),(tb)\}} \frac{(w_\pi)_s \phi''(\pi_s^a/\mu_s^a)}{\mu_s^a},$$

which proves (2.9) and shows that  $\Lambda$  is diagonal. The approximate Newton flow thus becomes

$$\frac{d\pi_s^a}{dt} = \mu_s^a (\phi''(\pi_s^a/\mu_s^a))^{-1} (r_s^a - \tau \phi'(\pi_s^a/\mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau,$$

which proves (2.10), or equivalently,

$$(5.13) \quad \frac{d(\phi'(\pi_s^a/\mu_s^a))}{dt} = (r_s^a - \tau\phi'(\pi_s^a/\mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau.$$

Let  $\theta_s^a = \phi'(\pi_s^a/\mu_s^a)$ , then

$$\frac{d\theta_s^a}{dt} = (r_s^a - \tau\theta_s^a - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau,$$

With a learning rate  $\eta$ , this becomes

$$\theta_s^a \leftarrow \eta(r_s^a - [(I - \gamma P^a)v_\pi]_s + c_s)/\tau + (1 - \eta)\theta_s^a, \quad \square$$

which proves (2.11).

### 5.3. Proof of Lemma 2.4.

*Proof.* Let

$$g(x) = \mu_1\psi(x + x_1) + \cdots + \mu_k\psi(x + x_k).$$

Since  $\psi : (L, +\infty) \rightarrow (0, +\infty)$  is decreasing,  $g(x)$  is positive and decreasing on  $(L - \min_{1 \leq i \leq k} x_i, +\infty)$ .

When  $x \rightarrow -\min_{1 \leq i \leq k} x_i$  from the right,  $g(x) \rightarrow +\infty$  since at least one of the terms go to  $+\infty$ . If

$$\min_{1 \leq i \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_i}\right) - x_i \right\} \geq L - \min_{1 \leq i \leq k} x_i, \text{ when } x = \min_{1 \leq i \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_i}\right) - x_i \right\}$$

$$\begin{aligned} g(x) &= \sum_{i=1}^k \mu_i \psi \left( \min_{1 \leq j \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_j}\right) - x_j \right\} + x_i \right) \\ &= \sum_{i=1}^k \mu_i \psi \left( \min_{1 \leq j \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_j}\right) - x_j \right\} + x_i - \psi^{-1}\left(\frac{1}{k\mu_i}\right) + \psi^{-1}\left(\frac{1}{k\mu_i}\right) \right) \\ &\geq \sum_{i=1}^k \mu_i \psi \left( \psi^{-1}\left(\frac{1}{k\mu_i}\right) \right) = \sum_{i=1}^k \mu_i \times \frac{1}{k\mu_i} = k \times \frac{1}{k} = 1. \end{aligned}$$

Since  $\psi^{-1}\left(\frac{1}{k\mu_i}\right) \geq L$ , we have  $\max_{1 \leq i \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_i}\right) - x_i \right\} \geq \max_{1 \leq i \leq k} \{L - x_i\} = L - \min_{1 \leq i \leq k} x_i$ . Then

when  $x = \max_{1 \leq i \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_i}\right) - x_i \right\}$ ,

$$g(x) = \sum_{i=1}^k \mu_i \psi \left( \max_{1 \leq j \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_j}\right) - x_j \right\} + x_i \right) \leq \sum_{i=1}^k \mu_i \psi \left( \psi^{-1}\left(\frac{1}{k\mu_i}\right) \right) = 1.$$

By the continuity of  $g$ , there exists a solution  $x$  to (2.17) on

$$\left[ \max \left\{ L - \min_{1 \leq i \leq k} x_i, \min_{1 \leq i \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_i}\right) - x_i \right\} \right\}, \max_{1 \leq i \leq k} \left\{ \psi^{-1}\left(\frac{1}{k\mu_i}\right) - x_i \right\} \right],$$

and the solution is unique by the strict monotonicity of  $g$  on  $(L - \min_{1 \leq i \leq k} x_i, \infty)$ .  $\square$

### 5.4. Proof of Proposition 2.5.

*Proof.* By Lemma 2.4 there is a unique solution  $\tilde{c}_s$  to the equation  $\sum_a \mu_s^a \psi(\tilde{c}_s + x_a) = 1$ , where  $x_a$  is defined as in (2.16). Now update the policy by

$$\pi_s^a \leftarrow \mu_s^a \psi(\tilde{c}_s + x_a) = \sum_a \mu_s^a \psi \left( -\frac{\eta}{\tau} c_s - (1 - \eta) \phi'(\pi_s^a/\mu_s^a) - \frac{\eta}{\tau} (r_s^a - [(I - \gamma P^a)v_\pi]_s) \right) \quad \square$$

one ensures that  $\pi_s^a \geq 0$  and  $\sum_a \pi_s^a = 1$ , and that the multiplier  $c_s$  with this property is unique.

### 5.5. Proof of Theorem 2.6.

*Proof.* In subsection 2.3 we have proved that the approximate Newton flow:

$$\frac{d\pi_s^a}{dt} = \mu_s^a (\phi''(\pi_s^a/\mu_s^a))^{-1} (r_s^a - \tau \phi'(\pi_s^a/\mu_s^a) - [(I - \gamma P^a)v_\pi]_s + c_s) / \tau$$

converges globally, so it suffices to show that the limiting policy is the optimal policy. Denote the limiting policy by  $\pi^\diamond$ . Since  $\mu_s^a > 0$  and  $(\phi''((\pi^\diamond)_s^a/\mu_s^a))^{-1} > 0$ , we have

$$(5.14) \quad r_s^a - \tau \phi'((\pi^\diamond)_s^a/\mu_s^a) - [(I - \gamma P^a)v_{\pi^\diamond}]_s + c_s = 0,$$

and  $c_s$  is a multiplier which ensures  $\sum_a (\pi^\diamond)_s^a = 1$ . From the theory of regularized MDP (Cf. [9]), we know that the optimal policy  $\pi^*$  is the unique solution to the Bellman maximal equation:

$$(5.15) \quad v = \max_{\pi} r_{\pi} + \gamma P_{\pi} v - \tau h_{\pi}.$$

Since  $v_{\pi^\diamond} = (I - \gamma P_{\pi^\diamond})^{-1} (r_{\pi^\diamond} - \tau h_{\pi^\diamond})$ , we have  $v_{\pi^\diamond} - \gamma P_{\pi^\diamond} v_{\pi^\diamond} = r_{\pi^\diamond} - \tau h_{\pi^\diamond}$ , or equivalently

$$v_{\pi^\diamond} = r_{\pi^\diamond} + \gamma P_{\pi^\diamond} v_{\pi^\diamond} - \tau h_{\pi^\diamond}.$$

Thus it now suffices to show that  $\pi^\diamond$  is the optimizer of the constrained maximization problem  $\max_{\pi} r_{\pi} + \gamma P_{\pi} v_{\pi^\diamond} - \tau h_{\pi}$ , or in the component form:

$$(5.16) \quad \max_{\pi} \sum_a (r_s^a + \gamma (P^a v_{\pi^\diamond})_s) \pi_s^a - \tau \sum_a \mu_s^a \phi(\pi_s^a/\mu_s^a).$$

Since  $\phi$  is convex and  $\mu$  is positive,  $\tau \sum_a \mu_s^a \phi(\pi_s^a/\mu_s^a)$  is also a convex function in  $\pi_s$ . By the theory of convex optimization (Cf. [5], chapter 5), the Karush-Kuhn-Tucker (KKT) condition is sufficient for the optimality when the objective function is convex, and the KKT condition for the problem (5.16) is

$$\begin{aligned} r_s^a + \gamma (P^a v_{\pi^\diamond})_s - \tau \phi'(\pi_s^a/\mu_s^a) + \lambda_s &= 0, \\ \sum_a \pi_s^a &= 1, \\ \pi_s^a &\geq 0, \end{aligned}$$

where  $\lambda_s$  is the Lagrange multiplier. Now let  $\pi = \pi^\diamond$  and  $\lambda_s = c_s - (v_{\pi^\diamond})_s$ . From the first-order condition (5.14) one can directly observe that the KKT condition above is satisfied, which makes  $\pi^\diamond$  the optimizer for (5.16) and  $v_{\pi^\diamond}$  the solution to the Bellman equation (5.15). Thus  $v_{\pi^\diamond}$  and  $\pi^\diamond$  are indeed the optimal value function and the optimal policy, which closes the proof.  $\square$

### 5.6. Proof of Theorem 3.1.

*Proof.* The proof is divided into three steps. First, we present some results needed in proving the local convergence. We then demonstrate the local convergence of  $\pi^{(k)}$  to  $\pi^*$  using induction in the second step. Finally, we prove that the convergence rate is quadratic.

**Step 1. Preparation.** From the scheme

$$(5.17) \quad \nabla \Phi(\pi^{(k+1)}) - \nabla \Phi(\pi^{(k)}) = - \left( f(\pi^{(k)}) + \nabla \Phi(\pi^{(k)}) - B^\top c(\pi^{(k)}) \right),$$

one can obtain the inequality

$$\begin{aligned} (5.18) \quad \|f(\pi^{(k+1)}) - f(\pi^{(k)})\| &\geq \frac{(f(\pi^{(k)}) - f(\pi^{(k+1)}))^\top (\pi^{(k+2)} - \pi^{(k+1)})}{\|\pi^{(k+2)} - \pi^{(k+1)}\|} \\ &= \frac{-(f(\pi^{(k+1)}) + \nabla \Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k)}))^\top (\pi^{(k+2)} - \pi^{(k+1)})}{\|\pi^{(k+2)} - \pi^{(k+1)}\|} \\ &= \frac{-(f(\pi^{(k+1)}) + \nabla \Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k+1)}))^\top (\pi^{(k+2)} - \pi^{(k+1)})}{\|\pi^{(k+2)} - \pi^{(k+1)}\|} \\ &= \frac{(\nabla \Phi(\pi^{(k+2)}) - \nabla \Phi(\pi^{(k+1)}))^\top (\pi^{(k+2)} - \pi^{(k+1)})}{\|\pi^{(k+2)} - \pi^{(k+1)}\|}, \end{aligned}$$



where we use the constraint  $B\pi^{(k+1)} = B\pi^{(k+2)} = \mathbf{1}_{|S|}$ . By a direct calculation of  $\nabla^2\Phi$  from the definition of  $\Phi$ , we can see that  $\nabla^2\Phi$  is diagonal and  $\Phi$  is strongly convex since  $\phi$  is strongly convex. As a result, there is some constant  $\omega > 0$  such that

$$(5.19) \quad (\nabla\Phi(\pi) - \nabla\Phi(\tilde{\pi}))^\top (\pi - \tilde{\pi}) \geq \omega \|\pi - \tilde{\pi}\|^2$$

for any  $\pi$  and  $\tilde{\pi}$ . Thus from (5.18) one can deduce that

$$(5.20) \quad \begin{aligned} \left\| f(\pi^{(k+1)}) - f(\pi^{(k)}) \right\| &\geq \frac{(\nabla\Phi(\pi^{(k+2)}) - \nabla\Phi(\pi^{(k+1)}))^\top (\pi^{(k+2)} - \pi^{(k+1)})}{\|\pi^{(k+2)} - \pi^{(k+1)}\|} \\ &\geq \frac{\omega \|\pi^{(k+2)} - \pi^{(k+1)}\|^2}{\|\pi^{(k+2)} - \pi^{(k+1)}\|} = \omega \|\pi^{(k+2)} - \pi^{(k+1)}\|. \end{aligned}$$

Let  $K$  be a closed set contained in  $\{\pi : B^\top\pi = \mathbf{1}_{|S|}, \pi_s^a > 0\}$  such that  $K$  contains a ball centered at  $\pi^*$  with radius  $\delta_0 > 0$ , which is guaranteed to exist since  $(\pi^*)_s^a > 0$ . Define the conjugate function of  $\Phi$  as

$$(5.21) \quad \Phi^*(x) = \max_{\pi \in \Delta} \left[ \sum_{sa} \pi_{sa} x_{sa} - \Phi(\pi) \right],$$

where  $\Delta = \{\pi : B^\top\pi = \mathbf{1}_{|S|}, \pi_{sa} \geq 0\}$ . Since  $\Phi$  is  $\omega$ -strongly convex and  $\Delta$  is a closed convex set, it can be deduced from classical convex analysis results (see [11] for example) that  $\nabla\Phi^*$  is  $\frac{1}{\omega}$ -Lipschitz continuous, and  $\pi = \nabla\Phi^*(\nabla\Phi(\pi))$ . Moreover, from the definition of  $\Phi^*$  one can observe that  $\Phi^*(x + B^\top c) = \Phi^*(x) + \mathbf{1}_{|S|}^\top c$ , and thus  $\nabla\Phi^*(x + B^\top c) = \nabla\Phi^*(x)$ . Similar results concerning the conjugate functions have also been used in [18] and [9]. Thanks to the properties of  $\Phi^*$ , we have the identity

$$(5.22) \quad \pi^{(k+1)} = \nabla\Phi^*(\nabla\Phi(\pi^{(k+1)})) = \nabla\Phi^*(B^\top c(\pi^{(k)}) - f(\pi^{(k)})) = \nabla\Phi^*(-f(\pi^{(k)})),$$

where we have used the update scheme (5.17). Moreover, by the result of Theorem 2.6 we have  $\frac{d(\nabla\Phi(\pi))}{dt} = 0$  at  $\pi = \pi^*$ , so  $f(\pi^*) + \nabla\Phi(\pi^*) = B^\top c(\pi^*)$  and

$$(5.23) \quad \pi^* = \nabla\Phi^*(\nabla\Phi(\pi^*)) = \nabla\Phi^*(B^\top c(\pi^*) - f(\pi^*)) = \nabla\Phi^*(-f(\pi^*)),$$

Since  $\nabla\Phi^*$  and  $-f$  are continuous on  $K$ , it can be concluded from (5.22) and (5.23) that there exists  $\delta_1 > 0$  such that  $\|\pi^{(k+1)} - \pi^*\| < \frac{1}{16} \min\{\frac{\omega}{M}, \delta_0\}$  whenever  $\|\pi^{(k)} - \pi^*\| \leq \delta_1$ , where  $M = \sup_{\pi \in K} |\nabla^2 f(\pi)|$ .

**Step 2. Prove the convergence by induction.** Now let  $\delta = \min\{\frac{\omega}{16M}, \frac{\delta_0}{16}, \delta_1\}$ . Assuming that  $\|\pi^{(0)} - \pi^*\| < \delta$ , we proceed to prove that  $\|\pi^{(k)} - \pi^*\| \leq \frac{1}{2} \min\{\frac{\omega}{M}, \delta_0\}$  for any  $k$  by induction. To this end, we first strengthen the induction hypothesis to

$$(5.24) \quad \begin{aligned} \left\| \pi^{(k)} - \pi^* \right\| &\leq \left( \frac{1}{2} - \frac{1}{2^{k+2}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\}, \quad k = 0, 1, \dots, n, \\ \left\| \pi^{(k+1)} - \pi^{(k)} \right\| &\leq \left( \frac{1}{2} - \frac{1}{2^{k+2}} \right) \left\| \pi^{(k)} - \pi^{(k-1)} \right\|, \quad k = 1, 2, \dots, n. \end{aligned}$$

We first prove (5.24) for  $n = 1$ . Note that

$$(5.25) \quad \left\| \pi^{(0)} - \pi^* \right\| \leq \delta \leq \left( \frac{1}{2} - \frac{1}{2^{0+2}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\},$$

by the definition of  $\delta$ , and that

$$(5.26) \quad \left\| \pi^{(1)} - \pi^* \right\| \leq \frac{1}{16} \min\left\{ \frac{\omega}{M}, \delta_0 \right\} \leq \left( \frac{1}{2} - \frac{1}{2^{1+2}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\},$$

by the definition of  $\delta_1$  and the fact that  $\|\pi^{(0)} - \pi^*\| \leq \delta_1$ . Then

$$(5.27) \quad \left\| \pi^{(1)} - \pi^{(0)} \right\| \leq \left\| \pi^{(1)} - \pi^* \right\| + \left\| \pi^{(0)} - \pi^* \right\| \leq \frac{1}{8} \min\left\{ \frac{\omega}{M}, \delta_0 \right\}.$$

In addition, from (5.25) and (5.26) we know that  $\pi^{(0)} \in K$  and  $\pi^{(1)} \in K$ . Then by (5.20),

$$(5.28) \quad \begin{aligned} \left\| \pi^{(2)} - \pi^{(1)} \right\| &\leq \frac{1}{\omega} \left\| f(\pi^{(1)}) - f(\pi^{(0)}) \right\|, \\ &= \frac{1}{\omega} \left\| \nabla f(\pi^{(0)} + \zeta((\pi^{(1)} - \pi^{(0)}))) (\pi^{(1)} - \pi^{(0)}) \right\| \\ &= \frac{1}{\omega} \left\| (\nabla f(\pi^{(0)} + \zeta((\pi^{(1)} - \pi^{(0)}))) - \nabla f(\pi^*)) (\pi^{(1)} - \pi^{(0)}) \right\| \\ &\leq \frac{M}{\omega} \left\| (\pi^{(0)} + \zeta((\pi^{(1)} - \pi^{(0)})) - \pi^*) (\pi^{(1)} - \pi^{(0)}) \right\| \\ &\leq \frac{M}{\omega} \max\left\{ \left\| \pi^{(1)} - \pi^* \right\|, \left\| \pi^{(0)} - \pi^* \right\| \right\} \left\| \pi^{(1)} - \pi^{(0)} \right\| \end{aligned}$$

where we have used the identity  $\nabla f(\pi^*)(\pi^{(1)} - \pi^{(0)}) = 0$  and the fact that  $\pi^{(1)}$  and  $\pi^{(0)}$  are contained in  $K$ . In fact, we can prove that

$$\nabla f(\pi^*)(\pi^{(k+1)} - \pi^{(k)}) = 0, \quad \text{for any } k,$$

as follows. Since  $f(\pi)_{sa} = -(r_s^a - ((I - \gamma P^a)v_\pi)_s)$  has a similar form with  $E(\pi)$ , we can directly obtain  $\nabla f(\pi)$

$$(5.29) \quad (\nabla f(\pi))_{sa,tb} = \lambda_{sa,t}(\pi) (-f(\pi)_{tb} + \tilde{c}(\pi)_t - \nabla \Phi(\pi)_{tb}),$$

where  $\lambda_{sa,t}(\pi) = Z_\pi^{-\top} \tilde{\rho}_{sa}$  and  $\tilde{\rho}_{sa}$  is the  $s$ -th row of  $I - \gamma P^a$ . Since  $f(\pi^*) + \nabla \Phi(\pi^*) = B^\top c(\pi^*)$ , we have

$$(5.30) \quad \begin{aligned} &(\nabla f(\pi^*)(\pi^{(k+1)} - \pi^{(k)}))_{sa} \\ &= \sum_{tb} \lambda_{sa,t}(\pi^*) (-f(\pi^*)_{tb} + \tilde{c}(\pi^*)_t - \nabla \Phi(\pi^*)_{tb}) (\pi_{tb}^{(k+1)} - \pi_{tb}^{(k)}) \\ &= \sum_{tb} \lambda_{sa,t}(\pi^*) (\tilde{c}(\pi^*)_t - c(\pi^*)_t) (\pi_{tb}^{(k+1)} - \pi_{tb}^{(k)}) \\ &= \sum_t \left[ \left( \lambda_{sa,t}(\pi^*) (\tilde{c}(\pi^*)_t - c(\pi^*)_t) \right) \left( \sum_b (\pi_{tb}^{(k+1)} - \pi_{tb}^{(k)}) \right) \right] \\ &= 0, \end{aligned}$$

where the last equality results from the fact that  $\sum_b \pi_{tb}^{(k+1)} = \sum_b \pi_{tb}^{(k)} = 1$  for any  $t$ . Now from (5.25), (5.26) and (5.28), we obtain

$$(5.31) \quad \begin{aligned} \left\| \pi^{(2)} - \pi^{(1)} \right\| &\leq \frac{M}{\omega} \max\left\{ \left\| \pi^{(1)} - \pi^* \right\|, \left\| \pi^{(0)} - \pi^* \right\| \right\} \left\| \pi^{(1)} - \pi^{(0)} \right\| \\ &\leq \frac{M}{\omega} \cdot \frac{1}{16} \min\left\{ \frac{\omega}{M}, \delta_0 \right\} \left\| \pi^{(1)} - \pi^{(0)} \right\| \\ &\leq \left( \frac{1}{2} - \frac{1}{2^{1+2}} \right) \left\| \pi^{(1)} - \pi^{(0)} \right\| \end{aligned}$$

Now, assuming that the induction hypothesis (5.24) holds for some  $n \geq 1$ , we have

$$\begin{aligned}
(5.32) \quad \left\| \pi^{(n+1)} - \pi^{(*)} \right\| &\leq \left\| \pi^{(n+1)} - \pi^{(n)} \right\| + \left\| \pi^{(n)} - \pi^{*} \right\| \\
&\leq \left( \prod_{k=1}^n \left( \frac{1}{2} - \frac{1}{2^{k+2}} \right) \right) \left\| \pi^{(1)} - \pi^{(0)} \right\| + \left\| \pi^{(n)} - \pi^{*} \right\| \\
&\leq \frac{1}{2^n} \cdot \frac{1}{8} \min\left\{ \frac{\omega}{M}, \delta_0 \right\} + \left( \frac{1}{2} - \frac{1}{2^{n+2}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\} \\
&= \left( \frac{1}{2} - \frac{1}{2^{n+2}} + \frac{1}{2^{n+3}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\} \\
&= \left( \frac{1}{2} - \frac{1}{2^{(n+1)+2}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\},
\end{aligned}$$

which also implies that  $\pi^{(n+1)} \in K$ . Now using the same reasoning as (5.28) but  $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)})$  replaced by  $(\pi^{(n)}, \pi^{(n+1)}, \pi^{(n+2)})$ , one obtains

$$(5.33) \quad \left\| \pi^{(n+2)} - \pi^{(n+1)} \right\| \leq \frac{M}{\omega} \max\left\{ \left\| \pi^{(n+1)} - \pi^{*} \right\|, \left\| \pi^{(n)} - \pi^{*} \right\| \right\} \left\| \pi^{(n+1)} - \pi^{(n)} \right\|.$$

After plugging (5.32) and the induction hypothesis into this inequality, we get

$$\begin{aligned}
(5.34) \quad \left\| \pi^{(n+2)} - \pi^{(n+1)} \right\| &\leq \frac{M}{\omega} \cdot \left( \frac{1}{2} - \frac{1}{2^{n+3}} \right) \min\left\{ \frac{\omega}{M}, \delta_0 \right\} \left\| \pi^{(n+1)} - \pi^{(n)} \right\| \\
&\leq \left( \frac{1}{2} - \frac{1}{2^{(n+1)+2}} \right) \left\| \pi^{(n+1)} - \pi^{(n)} \right\|
\end{aligned}$$

With (5.32) and (5.34) we have shown that (5.24) holds with  $n$  replaced by  $n+1$ . As a result, (5.24) holds for any  $n$ . From the second inequality in (5.24), it is clear that  $\pi^{(k)}$  converges (at least exponentially fast). Denote the limit of  $\pi^{(k)}$  by  $\tilde{\pi}$  for now, we obtain from (5.17) that

$$(5.35) \quad f(\tilde{\pi}) + \nabla\Phi(\tilde{\pi}) - B^\top c(\tilde{\pi}) = 0,$$

thus  $\tilde{\pi} = \pi^*$  by [Theorem 2.6](#).

**Step 3. Prove the convergence rate is quadratic.** Since  $\pi^{(k)}$  converges to  $\pi^*$  and  $\nabla f$  is Lipschitz continuous on  $K$ , we have

$$(5.36) \quad \lim_{k \rightarrow \infty} \frac{f(\pi^{(k+1)}) - f(\pi^{(k)}) - \nabla f(\pi^*) (\pi^{(k+1)} - \pi^{(k)})}{\left\| \pi^{(k+1)} - \pi^{(k)} \right\|} = 0.$$

On the other hand, we have

$$\begin{aligned}
(5.37) \quad & f(\pi^{(k+1)}) - f(\pi^{(k)}) - \nabla f(\pi^*) (\pi^{(k+1)} - \pi^{(k)}) \\
&= f(\pi^{(k+1)}) + \nabla\Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k)}) - \nabla f(\pi^*) (\pi^{(k+1)} - \pi^{(k)}) \\
&= f(\pi^{(k+1)}) + \nabla\Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k)}),
\end{aligned}$$

where we have used (5.30). Combining with (5.36) we arrive at

$$(5.38) \quad \lim_{k \rightarrow \infty} \frac{f(\pi^{(k+1)}) + \nabla\Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k)})}{\left\| \pi^{(k+1)} - \pi^{(k)} \right\|} = 0.$$

With the last three lines of (5.18), we obtain

$$(5.39) \quad \lim_{k \rightarrow \infty} \frac{(\nabla\Phi(\pi^{(k+2)}) - \nabla\Phi(\pi^{(k+1)}))^\top (\pi^{(k+2)} - \pi^{(k+1)})}{\left\| \pi^{(k+1)} - \pi^{(k)} \right\| \left\| \pi^{(k+2)} - \pi^{(k+1)} \right\|} = 0,$$

by multiplying the unit vector  $\frac{\pi^{(k+2)} - \pi^{(k+1)}}{\|\pi^{(k+2)} - \pi^{(k+1)}\|}$  to the fraction in (5.38). Then by (5.19) we get

$$(5.40) \quad 0 = \lim_{k \rightarrow \infty} \frac{\|\pi^{(k+2)} - \pi^{(k+1)}\|^2}{\|\pi^{(k+1)} - \pi^{(k)}\| \|\pi^{(k+2)} - \pi^{(k+1)}\|} = \lim_{k \rightarrow \infty} \frac{\|\pi^{(k+2)} - \pi^{(k+1)}\|}{\|\pi^{(k+1)} - \pi^{(k)}\|},$$

from which we can conclude that  $\pi^{(k)}$  converges to  $\pi^*$  superlinearly, i.e.,

$$(5.41) \quad \lim_{k \rightarrow \infty} \frac{\|\pi^{(k+1)} - \pi^*\|}{\|\pi^{(k)} - \pi^*\|} = 0.$$

In fact, for any  $\epsilon$  (assume  $\epsilon < 1/2$  without loss of generality), there is some  $k(\epsilon)$  such that for any  $k > k(\epsilon)$ ,  $\frac{\|\pi^{(k+2)} - \pi^{(k+1)}\|}{\|\pi^{(k+1)} - \pi^{(k)}\|} < \epsilon$ , then for any  $k > k(\epsilon)$

$$\begin{aligned} \|\pi^{(k+1)} - \pi^*\| &\leq \sum_{n=k+1}^{\infty} \|\pi^{(n+1)} - \pi^{(n)}\| \leq \sum_{n=k+1}^{\infty} \epsilon^{n-k} \|\pi^{(k+1)} - \pi^{(k)}\| \\ &\leq \frac{\epsilon}{1-\epsilon} \|\pi^{(k+1)} - \pi^{(k)}\| \leq 2\epsilon \|\pi^{(k+1)} - \pi^{(k)}\|. \end{aligned}$$

Then

$$\begin{aligned} \|\pi^{(k)} - \pi^*\| &\geq \|\pi^{(k+1)} - \pi^{(k)}\| - \|\pi^{(k+1)} - \pi^*\| \\ &\geq \left(\frac{1}{2\epsilon} - 1\right) \|\pi^{(k+1)} - \pi^{(k)}\|. \end{aligned}$$

For any  $G > 0$ , take  $\epsilon = 1/(2G + 2)$ , then for any  $k > k(\epsilon)$ ,

$$(5.42) \quad \|\pi^{(k)} - \pi^*\| \geq \left(\frac{1}{2\epsilon} - 1\right) \|\pi^{(k+1)} - \pi^{(k)}\| = (G + 1) \|\pi^{(k+1)} - \pi^{(k)}\| > G \|\pi^{(k+1)} - \pi^*\|,$$

which shows that  $\lim_{k \rightarrow \infty} \frac{\|\pi^{(k)} - \pi^*\|}{\|\pi^{(k+1)} - \pi^*\|} = +\infty$  and thus (5.41) holds. Now, from (5.17) and (5.30) we have

$$\begin{aligned} &\left\| f(\pi^{(k+1)}) + \nabla \Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k)}) \right\| \\ &= \left\| f(\pi^{(k+1)}) - f(\pi^{(k)}) - \nabla f(\pi^*) (\pi^{(k+1)} - \pi^{(k)}) \right\| \\ &= \left\| \left( \int_0^1 [\nabla f(\pi^{(k)} + t(\pi^{(k+1)} - \pi^{(k)})) - \nabla f(\pi^*)] dt \right) (\pi^{(k+1)} - \pi^{(k)}) \right\| \\ &\leq \tilde{C} \|\pi^{(k)} - \pi^*\| \|\pi^{(k+1)} - \pi^{(k)}\| \end{aligned}$$

for some constant  $\tilde{C}$ , where we used (5.41) and the Lipschitz continuity of  $\nabla f$  in the last equality. Multiplying both sides by  $\pi^{(k+2)} - \pi^{(k+1)}$ , and by (5.19) and the last three lines of (5.18) we have

$$\begin{aligned} &\omega \|\pi^{(k+2)} - \pi^{(k+1)}\|^2 \\ &\leq \left( \nabla \Phi(\pi^{(k+2)}) - \nabla \Phi(\pi^{(k+1)}) \right)^\top (\pi^{(k+2)} - \pi^{(k+1)}) \\ &= \left( f(\pi^{(k+1)}) + \nabla \Phi(\pi^{(k+1)}) - B^\top c(\pi^{(k)}) \right)^\top (\pi^{(k+2)} - \pi^{(k+1)}) \\ &\leq \tilde{C} \|\pi^{(k)} - \pi^*\| \|\pi^{(k+1)} - \pi^{(k)}\| \|\pi^{(k+2)} - \pi^{(k+1)}\|, \end{aligned}$$

which implies that

$$(5.43) \quad \|\pi^{(k+2)} - \pi^{(k+1)}\| \leq \tilde{C} \|\pi^{(k)} - \pi^*\| \|\pi^{(k+1)} - \pi^{(k)}\|,$$

with some constant  $\tilde{C}$ . From (5.41), we have

$$(5.44) \quad \lim_{k \rightarrow \infty} \frac{\|\pi^{(k)} - \pi^{(k+1)}\|}{\|\pi^{(k)} - \pi^*\|} = \lim_{k \rightarrow \infty} \frac{\|\pi^{(k+1)} - \pi^{(k+2)}\|}{\|\pi^{(k+1)} - \pi^*\|} = 1.$$

Combining this with (5.43) leads to

$$(5.45) \quad \left\| \pi^{(k+1)} - \pi^* \right\| \leq C \left\| \pi^{(k)} - \pi^* \right\|^2,$$

for some constant  $C$ , which closes the proof.  $\square$

**6. Conclusion and Discussion.** In this paper, we present a fast approximate Newton method for the policy gradient algorithm with provable quadratic convergence. The proposed method gives a systematic theory that includes the well-known natural policy gradient algorithm as a special case and naturally extends to other regularizers such as the reverse KL divergence, the Hellinger divergence, and the  $\alpha$ -divergence.

With a relatively simple proof, we show the local quadratic convergence of the proposed approximate Newton method as well as the global convergence of the approximate Newton gradient flow to the optimal solution. The quadratic convergence is confirmed numerically on both medium and large sparse models. In contrast with mirror descent type first-order methods (e.g. [40]) that take up to tens of thousands of iterations even with manually tuned learning rate, the proposed approximate Newton algorithms typically converge in less than 10 iterations, despite the large discount rate ( $\approx 1$ ) and small regularization coefficient ( $\approx 0$ ).

For future work, we plan to adapt the technique used here to other gradient-based algorithms for solving the MDP problems. Other forms of  $f$ -divergence can also be included. An interesting direction is to apply different types of numerical schemes for ordinary differential equations (ODEs) to the approximate Newton gradient flow presented in Subsection 2.3, which can be helpful for obtaining a good initial policy such that the discrete approximate Newton method is able to achieve fast quadratic convergence. Another direction is to consider continuous MDP problems by leveraging function approximation, effective spatial discretization, or model reduction.

## REFERENCES

- [1] A. AGARWAL, S. M. KAKADE, J. D. LEE, AND G. MAHAJAN, *Optimality and approximation with policy gradient methods in Markov decision processes*, in Conference on Learning Theory, PMLR, 2020.
- [2] Z. AHMED, N. LE ROUX, M. NOROUZI, AND D. SCHUURMANS, *Understanding the impact of entropy on policy optimization*, in International Conference on Machine Learning, PMLR, 2019.
- [3] S. M. ALI AND S. D. SILVEY, *A general class of coefficients of divergence of one distribution from another*, Journal of the Royal Statistical Society: Series B (Methodological), 28 (1966), pp. 131–142.
- [4] R. BELLMAN, *A markovian decision process*, Journal of mathematics and mechanics, 6 (1957), pp. 679–684.
- [5] S. BOYD, S. P. BOYD, AND L. VANDENBERGHE, *Convex optimization*, Cambridge university press, 2004.
- [6] S. CEN, C. CHENG, Y. CHEN, Y. WEI, AND Y. CHI, *Fast global convergence of natural policy gradient methods with entropy regularization*, July 2020, <https://arxiv.org/abs/2007.06558>.
- [7] L. G. DE PILLIS, *A comparison of iterative methods for solving nonsymmetric linear systems*, Acta Applicandae Mathematica, 51 (1998), pp. 141–159.
- [8] J. E. DENNIS AND J. J. MORÉ, *A characterization of superlinear convergence and its application to quasi-newton methods*, Mathematics of computation, 28 (1974), pp. 549–560.
- [9] M. GEIST, B. SCHERRER, AND O. PIETQUIN, *A theory of regularized Markov decision processes*, in International Conference on Machine Learning, PMLR, 2019.
- [10] S. GUNASEKAR, B. WOODWORTH, AND N. SREBRO, *Mirrorless mirror descent: A natural derivation of mirror descent*, in International Conference on Artificial Intelligence and Statistics, 2021.
- [11] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Fundamentals of convex analysis*, Springer Science & Business Media, 2004.
- [12] S. M. KAKADE, *A natural policy gradient*, in Advances in Neural Information Processing Systems, 2001.
- [13] V. R. KONDA AND J. N. TSITSIKLIS, *Actor-critic algorithms*, in Advances in Neural Information Processing Systems, 2000.
- [14] G. LAN, *Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes*, Feb. 2021, <https://arxiv.org/abs/2102.00135>.
- [15] G. LI, Y. WEI, Y. CHI, Y. GU, AND Y. CHEN, *Softmax policy gradient methods can take exponential time to converge*, Feb. 2021, <https://arxiv.org/abs/2102.11270>.

- [16] J. MARTENS, *New insights and perspectives on the natural gradient method*, Journal of Machine Learning Research, 21 (2020), pp. 1–76.
- [17] J. MEI, C. XIAO, C. SZEPEŠVARI, AND D. SCHUURMANS, *On the global convergence rates of softmax policy gradient methods*, in International Conference on Machine Learning, PMLR, 2020.
- [18] A. MENSCH AND M. BLONDEL, *Differentiable dynamic programming for structured prediction and attention*, in International Conference on Machine Learning, 2018.
- [19] V. MNIH, A. P. BADIA, M. MIRZA, A. GRAVES, T. LILLICRAP, T. HARLEY, D. SILVER, AND K. KAVUKCUOĞLU, *Asynchronous methods for deep reinforcement learning*, in International Conference on Machine Learning, PMLR, 2016.
- [20] A. S. NEMIROVSKIJ AND D. B. YUDIN, *Problem complexity and method efficiency in optimization*, Wiley, 1983.
- [21] Y. NESTEROV, *Primal-dual subgradient methods for convex problems*, Mathematical programming, 120 (2009), pp. 221–259.
- [22] G. NEU, A. JONSSON, AND V. GÓMEZ, *A unified view of entropy-regularized Markov decision processes*, May 2017, <https://arxiv.org/abs/1705.07798>.
- [23] G. RASKUTTI AND S. MUKHERJEE, *The information geometry of mirror descent*, IEEE Transactions on Information Theory, 61 (2015), pp. 1451–1457.
- [24] A. RÉNYI, *On measures of entropy and information*, in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics, 1961.
- [25] A. RODOMANOV AND Y. NESTEROV, *New results on superlinear convergence of classical quasi-newton methods*, Journal of optimization theory and applications, 188 (2021), pp. 744–769.
- [26] A. RODOMANOV AND Y. NESTEROV, *Rates of superlinear convergence for classical quasi-newton methods*, Mathematical Programming, (2021), pp. 1–32.
- [27] Y. SAAD, *Iterative methods for sparse linear systems*, SIAM, 2003.
- [28] J. SCHULMAN, S. LEVINE, P. ABBEEL, M. JORDAN, AND P. MORITZ, *Trust region policy optimization*, in International Conference on Machine Learning, PMLR, 2015.
- [29] J. SCHULMAN, P. MORITZ, S. LEVINE, M. JORDAN, AND P. ABBEEL, *High-dimensional continuous control using generalized advantage estimation*, June 2015, <https://arxiv.org/abs/1506.02438>.
- [30] J. SCHULMAN, F. WOLSKI, P. DHARIWAL, A. RADFORD, AND O. KLIMOV, *Proximal policy optimization algorithms*, July 2017, <https://arxiv.org/abs/1707.06347>.
- [31] L. SHANI, Y. EFRONI, AND S. MANNOR, *Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps*, in Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
- [32] D. SILVER, G. LEVER, N. HEES, T. DEGRIS, D. WIERSTRA, AND M. RIEDMILLER, *Deterministic policy gradient algorithms*, in International Conference on Machine Learning, PMLR, 2014.
- [33] R. S. SUTTON AND A. G. BARTO, *Reinforcement learning: An introduction*, MIT press, 2018.
- [34] R. S. SUTTON, D. A. MCALLESTER, S. P. SINGH, AND Y. MANSOUR, *Policy gradient methods for reinforcement learning with function approximation*, in Advances in Neural Information Processing Systems, 2000.
- [35] M. TOMAR, L. SHANI, Y. EFRONI, AND M. GHAVAMZADEH, *Mirror descent policy optimization*, May 2020, <https://arxiv.org/abs/2005.09814>.
- [36] H. A. VAN DER VORST, *Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of non-symmetric linear systems*, SIAM Journal on scientific and Statistical Computing, 13 (1992), pp. 631–644.
- [37] L. WANG AND M. YAN, *Hessian informed mirror descent*, June 2021, <https://arxiv.org/abs/2106.13477>.
- [38] R. J. WILLIAMS, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, Machine learning, 8 (1992), pp. 229–256.
- [39] L. YING, *Mirror descent algorithms for minimizing interacting free energy*, Journal of Scientific Computing, 84 (2020), pp. 1–14.
- [40] W. ZHAN, S. CEN, B. HUANG, Y. CHEN, J. D. LEE, AND Y. CHI, *Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence*, May 2021, <https://arxiv.org/abs/2105.11066>.