ACCELERATING PRIMAL-DUAL METHODS FOR REGULARIZED MARKOV DECISION PROCESSES[*]

HAOYA LI[†], HSIANG-FU YU[‡], LEXING YING[†], AND INDERJIT S. DHILLON[§]

Abstract. Entropy regularized Markov decision processes have been widely used in reinforcement learning. This paper is concerned with the primal-dual formulation of the entropy regularized problems. Standard first-order methods suffer from slow convergence due to the lack of strict convexity and concavity. To address this issue, we first introduce a new quadratically convexified primal-dual formulation. The natural gradient ascent descent of the new formulation enjoys global convergence guarantee and exponential convergence rate. We also propose a new interpolating metric that further accelerates the convergence significantly. Numerical results are provided to demonstrate the performance of the proposed methods under multiple settings.

Key words. reinforcement learning, Markov decision process, primal-dual method, entropy regularization

1. Introduction.

1.1. Setup. Consider an infinite-horizon Markov decision process (MDP) [4, 35, 29] $\mathcal{M} = (S, A, P, r, \gamma)$, where S is a set of states of the Markov chain and A is a set of actions. P is a transition probability tensor with $P_{ass'}$ being the probability of transitioning from state s to state s' when taking action a, r is a reward matrix with r_{sa} being the reward obtained when taking action a at state s, and $\gamma \in (0, 1)$ is the discount factor. In this paper, we assume that the state space S and the action space A are finite.

A policy π is a randomized strategy over the actions at each state, i.e., for each state s, π_{sa} is the probability of choosing action a at s. For a given policy, the value function $v_\pi \in \mathrm{R}^{|S|}$ is a vector defined as

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

where the expectation is taken over all possible trajectories $\{(s_k, a_k)\}_{k \geq 0}$ starting from $s_0 = s$ following the policy π. The value function v_π satisfies the well-known Bellman equation [4]

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

where $(P_\pi)_{ss'} := \sum_{a \in A} \pi_{sa} P_{ass'}$, $(r_\pi)_s := \sum_{a \in A} \pi_{sa} r_{sa}$, and I is the identity operator. In a Markov decision problem, the goal is to find the optimal policy π^* such that

$$v_{\pi^*}(s) \geq v_\pi(s) \quad \forall s \in S,$$

[†]Department of Mathematics, Stanford University, Stanford, CA 94305 USA (lihaoya@stanford.edu, lexing@stanford.edu).
[‡]Amazon Search, Palo Alto, CA 94301 USA (rofu.yu@gmail.com).
[§]Department of Computer Science, University of Texas at Austin, Austin, TX 78712 USA, and Google, Mountain View, CA 94043 USA. This work was done while this author was at Amazon Search (inderjit@cs.utexas.edu).

for any other policy π. The corresponding optimal value function v_{π^*} will also be referred to as v^* in this paper. The existence of v^* and π^* is guaranteed by the theory of MDP [29].

In recent studies, entropy regularization has been widely used in MDP problems to encourage exploration and enhance the robustness [27, 10, 12, 2, 1, 24, 7, 42]. With the entropy regularization, the value function is defined by

$$(1.3) \qquad (v_\pi)_s := \mathbb{E} \sum_{k=0}^\infty \left(\gamma^k (r_{s_k a_k} - \tau \log \pi_{s_k a_k}) \mid s_0 = s \right),$$

where $\tau > 0$ is the regularization coefficient. v_π satisfies the regularized Bellman equation

$$(1.4) \qquad (I - \gamma P_\pi) v_\pi = r_\pi - \tau h_\pi,$$

where h_π is a vector in $\mathrm{R}^{|S|}$ with each entry $(h_\pi)_s$ given by the negative Shannon entropy of $(\pi_{sa})_{a \in A}$

$$(h_\pi)_s = \sum_{a \in A} \pi_{sa} \log \pi_{sa}.$$

Here we overload the notation v_π for the regularized value function and for the rest of the paper v_π shall always denote the regularized value function (1.3) unless otherwise specified. For the entropy regularized MDP (see [12]), there exists a unique optimal policy π^*, such that

$$(1.5) \qquad v^*(s) := v_{\pi^*}(s) \geq v_\pi(s) \quad \forall s \in S,$$

for any other policy π.

Without loss of generality, *the reward r_{sa} is assumed to be nonnegative* throughout this paper. This can be guaranteed by adding to the rewards a sufficiently large constant C. Note that such a uniform shift keeps the optimal policy π^* unchanged and shifts v^* by a constant $\frac{C}{1-\gamma}$.

1.2. Primal-dual formulation. Entropy regularized MDPs enjoy regularized linear programming formulations, in the primal, dual, and primal-dual forms. In this paper, we are concerned with the primal-dual formulation (see, for example, [27, 41]):

$$\min_{v \in \mathrm{R}^{|S|}} \max_{u \in \mathrm{R}^{|S| \times |A|}} \sum_{s \in S} e_s v_s + \sum_{s \in S, a \in A} u_{sa}(r_{sa} - ((I - \gamma P_a)v)_s) - \tau \sum_{s \in S, a \in A} u_{sa} \log(u_{sa}/\tilde{u}_s),$$

where $\tilde{u}_s := \sum_{a \in A} u_{sa}$. The policy π is related to u via the relationship

$$\pi_{sa} = u_{sa}/\tilde{u}_s.$$

The main advantage of working with the primal-dual formulation is that the transition matrix P_a appears linearly in the objective function of the primal-dual problem. This linearity brings an important benefit when a stochastic gradient method is used to solve the primal-dual formulation: an unbiased estimator of the transition matrix P_a guarantees an unbiased estimator for the gradient-based update rule. This avoids the famous double-sampling problem [35] that affects any formulation that performs a nonlinear operation to the transition matrix P_a. Examples of these affected formulations include the primal formulation, where a nonlinear max or exponentiation

operator is applied to P_a, and the dual formulation, where the inverse of $I - \gamma P_\pi$ is needed. From this perspective, the primal-dual formulation is convenient in the model-free setting, where the transition probability tensor can only be estimated from samples and is thus inherently noisy.

In what follows, we shall simplify the notation by denoting $K_a = I - \gamma P_a$ and $K_\pi = I - \gamma P_\pi$. With this simplification, the primal-dual problem can be rewritten more compactly as

$$(1.6) \qquad \min_v \max_u \sum_s e_s v_s + \sum_{sa} u_{sa}(r_{sa} - (K_a v)_s) - \tau \sum_{sa} u_{sa} \log(u_{sa}/\tilde{u}_s).$$

Though theoretically appealing, the primal-dual formulation (1.6) often poses computational challenges because it is a minimax optimization. Newton-type methods are often impractical to apply because either P_a is only accessible via samples or its size is too large for practical inversion. A close look at the objective function of (1.6) suggests that it is linear with respect to both the value function v and the dual variable u in the radial direction. This lack of strict convexity/concavity makes it difficult for the first-order methods to converge.

1.3. Contributions. To overcome this difficulty, this paper proposes a *quadratically convexified* reformulation of (1.6) that shares the same solution with (1.6) and an *interpolating* natural gradient ascent descent method that significantly speeds up the convergence. More specifically, the main contributions of this paper are listed as follows:

- We propose a new quadratically convexified primal-dual formulation in which the linear weighted sum $e^\mathsf{T} v$ of (1.6) is replaced with a quadratic term $\frac{\alpha}{2}\|v\|^2$. The surprising feature is that the solution (v^*, u^*) is unchanged and is independent of the hyperparameter $\alpha > 0$. We prove that the vanilla natural gradient ascent descent (NGAD) of this quadratically convexified problem enjoys a Lyapunov function [23] and converges linearly. To the best of our knowledge, this is the first quadratically convexified primal-dual formulation of Markov decision problems.
- We propose an interpolating natural gradient ascent descent (INGAD) by introducing a new interpolating metric for the u variable. The corresponding Lyapunov function is constructed, and the convergence of the new dynamics is proved. The acceleration is verified by numerical tests under multiple settings.

1.4. Related work. Regarding the primal-dual formulation, the first primal-dual learning algorithm is given in [39]. A follow-up work [38] leverages the binary-tree data structure and adaptive importance sampling techniques to reduce the complexity. The convergence result for these two papers is, however, only for the average of all the policies rather than the policy obtained in the last iteration. In these papers, no regularization is used in the formulation and no preconditioner is used in the iterative update scheme. As a comparison, the current paper proves a last-iteration convergence result with the help of the Lyapunov method and entropy regularization, and derives an interpolating metric that accelerates the algorithm. Various studies have been carried out following the primal-dual formulation in [39]. For example, a modified form with the Q-function is proposed in [20], and the corresponding primal-dual type algorithm is derived. An extension to the infinite-horizon average-reward setting is provided in [37], but only the average-case convergence result is given. A later work [8] further extends this method to the function approximation setting. A comprehensive review of the primal-dual methods in the average reward setting is

given in a recent thesis [13], and a generalization to the general utility maximization formulation is provided. The primal-dual method has also been used to find risk-sensitive policies, for example, in [43], where a risk function is integrated into the primal-dual objective through the dual variable. In the optimization literature, the primal-dual formulation is often called the saddle-point problem: for example, [34] considers a linear relaxation version of the saddle-point problem in [37] to address large-scale problems. However, it is worth noting that no (entropy) regularization is used in the papers mentioned above. Entropy regularization is known to be able to make the landscape of the optimization problem smoother and is thus a crucial element of recent linear convergence results [7, 22, 19]. Linear convergence results can be developed with the presence of preconditioners. For example, in [18], the authors show that the natural policy gradient method with an exact evaluation of the gradient has a linear convergence rate after sufficiently many gradient steps, where the convergence rate relies on an advantage function gap. Without regularization or preconditioners, gradient-type methods can take exponential time to converge [21].

Besides the primal-dual formulations, the discussion below briefly touches on the primal and dual formulations. For the entropy regularized Markov decision process, the primal formulation [41] takes the form

$$(1.7) \qquad v_s = \tau \log\left(\sum_a \exp\left(\frac{r_{sa} + \gamma \sum_{s'} P_{ass'} v_{s'}}{\tau},\right)\right),$$

which leads to a value iteration algorithm. Let $\varphi(v): \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ be the fixed-point map such that $\varphi(v)_s = \tau \log\left(\sum_{a \in A} \exp\left(\tau^{-1}(r_{sa} + \gamma \sum_{s' \in S} P_{ass'} v_{s'})\right)\right)$. By calculating the derivative matrix, we have

$$\|D\varphi(v)\|_\infty = \max_s \sum_{s'} |(D\varphi(v))_{ss'}|$$
$$= \max_s \sum_{s'} \frac{\gamma \sum_a P_{ass'} \exp\left(\tau^{-1}(r_{sa} + \gamma \sum_{s''} P_{ass''} v_{s''})\right)}{\sum_a \exp\left(\tau^{-1}(r_{sa} + \gamma \sum_{s''} P_{ass''} v_{s''})\right)} = \gamma.$$

Hence φ is a contraction map and converges to a fixed point, which is the solution to (1.7) at a linear rate $O(\gamma^T)$, where T is the number of iterations. After obtaining the optimal value function v, the corresponding policy π is given by [41]:

(1.8)

$$\pi_{sa} = \frac{\exp\left(\tau^{-1}\left(r_{sa} + \gamma \sum_{s' \in S} P_{ass'} v_{s'}\right)\right)}{\sum_a \exp\left(\tau^{-1}\left(r_{sa} + \gamma \sum_{s' \in S} P_{ass'} v_{s'}\right)\right)} = \exp\left(\tau^{-1}\left(r_{sa} - \sum_{s'} (I - \gamma P_a)_{ss'} v_{s'}\right)\right).$$

As a result of the aforementioned double-sampling problem, the value-iteration algorithm based on (1.7) is mainly used in the model-based setting, but due to the nice properties of φ, it appears as an important ingredient in various other algorithms. For example, in [3] and [30], the authors use the function φ as an alternative softmax operator and form a Q-learning type algorithm, and in [26], the function φ appears as a result of the inner optimization of an entropy regularized trust region–type formulation and is used to form the loss function. In [10], the mean squared regularized Bellman error is employed to establish the optimization problem.

An alternative way to solve a regularized Markov decision problem in the model-based setting is the dual formulation [41], in which one seeks a policy π that solves the following optimization problem:

$$(1.9) \qquad \max_\pi e^\mathsf{T} v_\pi := e^\mathsf{T}(I - \gamma P_\pi)^{-1}(r_\pi - \tau h_\pi),$$

where $e \succ 0$ is a weight vector. By the existence and uniqueness of the optimal value function and optimal policy and the optimality (1.5), it is clear that any choice of e leads to the optimal policy and the optimal value function. A variety of policy gradient algorithms can be used to solve the dual problem. Examples include [40, 36, 15, 32, 31, 33], to mention only a few. Recently, [22] proposes a quasi-Newton policy gradient algorithm, where an approximate Hessian of the objective function in (1.9) is used as a preconditioner for the gradient, resulting in a quadratic convergence rate by better fitting the problem geometry.

The word *primal-dual* also appears in other types of formulations where the dual variables do not represent the policy. For example, in [11], the authors apply the natural policy gradient method to constrained MDPs (CMDPs), where the dual variables are the multipliers of the constraints. Similarly, in [9], the dual variables come from the constraints in CMDPs. In this paper, the Lyapunov method is used to give a theoretical analysis of the natural gradient flow of the method we propose. The idea of Lyapunov methods has also been applied to discrete time control problems [17, 16] and to discrete Markovian systems [25]. Recently it has also been used to address the safety problem, where safety usually appears as additional constraints in the model [28, 9, 5], and the Lyapunov function is usually defined on the state space and is used explicitly in the policy iteration or in finding the controller.

1.5. Notation. For a vector $x \in \mathbb{R}^d$, $\mathrm{diag}(x)$ denotes a diagonal matrix with size $d \times d$ and the kth diagonal element being x_k, $1 \leq k \leq d$. For $u \in \mathbb{R}^{|S||A|}$, we denote the $((s-1)|A| + a)$th element as u_{sa}. While $u_{s\cdot}$ denotes the vector in $\mathbb{R}^{|A|}$ with the ath element being u_{sa}, $u_{\cdot a}$ denotes the vector in $\mathbb{R}^{|S|}$ with the sth element being u_{sa}. The states of the MDP are typically referred to as s, s', and s'', while the actions are referred to as a and a'. The vector with length d and all elements equal to 1 is denoted by $\mathbf{1}_d$, and the subscript d is often omitted when there is no ambiguity. The $d \times d$ identity matrix is denoted by I_d, again with the subscript d often omitted when there is no ambiguity. For a matrix B, B^{H} denotes its Hermitian transpose. If a scalar function is applied to a vector, then the result is defined elementwise unless otherwise specified, e.g., for $x \in \mathbb{R}^d$, $\exp(x) \in \mathbb{R}^d$ with $\exp(x)_k = \exp(x_k)$ for $1 \leq k \leq d$.

1.6. Contents. The rest of the paper is organized as follows. Section 2 derives the quadratically convexified primal-dual formulation, proves its equivalence with (1.6), and shows that the vanilla NGAD of the new formulation converges linearly using a Lyapunov function method. Section 3 introduces an interpolating metric by leveraging the flexibility of the underlying metric described by the block diagonal part of the Hessian. The convergence rate of the INGAD based on this new interpolating metric is significantly improved. We also provide a Lyapunov-style proof for global convergence and an analysis of the exponential convergence rate in the last-iterate sense. Finally, section 4 demonstrates the numerical performance of these proposed natural gradient methods.

2. Quadratically convexified primal-dual formulation.

2.1. Formulation. In what follows, we use $E_0(v, u)$ to denote the objective of the standard entropy regularized primal-dual formulation

$$(2.1) \qquad \min_v \max_u E_0(v, u) := \sum_s e_s v_s + \sum_{sa} u_{sa}(r_{sa} - (K_a v)_s) - \tau \sum_{sa} u_{sa} \log \frac{u_{sa}}{\tilde{u}_s}.$$

Since $E_0(v, u)$ is linear in v and linear along the radial direction of u, first-order optimization methods typically experience slow convergence. To address the issue in the v variable, we propose a quadratically convexified primal-dual formulation:

$$(2.2) \qquad \min_v \max_u E(v, u) := \frac{\alpha}{2} \sum_s v_s^2 + \sum_{sa} u_{sa}(r_{sa} - (K_a v)_s) - \tau \sum_{sa} u_{sa} \log \frac{u_{sa}}{\tilde{u}_s}.$$

Though these two formulations look quite different, they are indeed equivalent when $r_{sa} > 0$ in the following sense:

- They share the same optimal value function v^*.
- The optimal dual variable u^* differs only by an s-dependent scaling factor. This implies that the optimal policy $\pi_{sa}^* \equiv u_{sa}^*/\tilde{u}_s^*$ are the same.

One geometric way to see this equivalence is to go through the associated primal formulations

$$(2.3) \qquad \min_v e^\mathsf{T} v, \text{ s.t. } \forall s, v_s \geq \tau \log \left(\sum_{a \in A} \exp \left(\frac{r_{sa} + \gamma \sum_{s'} P_{ass'} v_{s'}}{\tau} \right) \right)$$

and

$$(2.4) \qquad \min_v \frac{\alpha}{2} \|v\|^2, \text{ s.t. } \forall s, v_s \geq \tau \log \left(\sum_{a \in A} \exp \left(\frac{r_{sa} + \gamma \sum_{s'} P_{ass'} v_{s'}}{\tau} \right) \right).$$

Figure 1 illustrates the primal formulations of a randomly generated MDP with $|S| = |A| = 2$, where the yellow region represents the feasible set and the red dot represents the optimal value v^*. Due to the key assumption $r_{sa} \geq 0$, the feasible set lies in the first quadrant. From the contour plots of the objective function $e^\mathsf{T} v$ and $\|v\|^2$ shown by the dotted curves, it is clear that both objective functions are minimized at v^* when constrained to the feasible set.

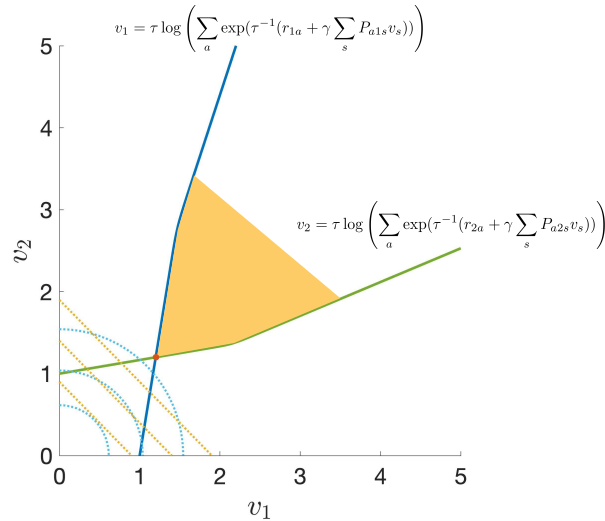The following theorem states this equivalence formally, with its proof given in section 6.

FIG. 1. *This plot heuristically demonstrates the correctness of the quadratically convexified primal-dual formulation on a randomly generated MDP with $|S| = |A| = 2$. The yellow region represents the feasible set of the primal problem (2.3), whose boundary corresponds to the solution to (1.7) and is shown by the blue and green curves. The red dot denotes the optimal value v^*. The cyan and orange dotted curves are contour lines of $\|v\|^2$ and $e^\mathsf{T} v$, respectively. It can be seen from this plot that the solution to the quadratically convexified formulation (2.4) is also v^*. (Color figure available online.)*

THEOREM 2.1. *For an infinite-horizon discounted MDP with finite state space S, finite action space A, and nonnegative reward r, we have the following properties:*

    (a) *There is a unique solution (v^*, u°) to the primal-dual problem:*

$$\min_v \max_u E_0(v, u) = \sum_s e_s v_s + \sum_{sa} u_{sa} \left(r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right)$$
$$- \tau \sum_{sa} u_{sa} \log \frac{u_{sa}}{\tilde{u}_s},$$

           where v^* is the optimal value function defined by (1.5) *and $\frac{u^\circ_{sa}}{\tilde{u}^\circ_s}$ gives the optimal policy π^*_{sa}.*

    (b) *There is a unique solution (v^*, u^*) to the quadratically convexified problem:*

$$\min_v \max_u E(v, u) = \frac{\alpha}{2} \sum_s v_s^2 + \sum_{sa} u_{sa} \left(r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right)$$
$$- \tau \sum_{sa} u_{sa} \log \frac{u_{sa}}{\tilde{u}_s},$$

           where v^* is the optimal value function, and $\frac{u^*_{sa}}{\tilde{u}^*_s}$ coincides with the optimal policy π^*_{sa}.

Remark 2.2. With the same method as the one used for the proof of Theorem 2.1, one can show that the conclusions of Theorem 2.1 still hold if the term $\frac{\alpha}{2} \sum_s v_s^2$ in the formulation (2.2) is replaced with a strictly increasing convex function of v. The intuition provided in Figure 1 also applies.

2.2. Natural gradient ascent descent. As mentioned earlier, the gradient-based methods for the primal-dual formulation (2.1) suffer from slow convergence, partly due to the linearity of $E_0(v, u)$ in v. Since the quadratically convexified scheme (2.2) gives the same value function v^* and policy π^* as the original primal-dual problem (2.1), we work instead with (2.2) and propose an NGAD algorithm.

The first-order derivatives of the new objective function $E(v, u)$ are

(2.5)
$$\frac{\partial E}{\partial v_{s'}} = \alpha v_{s'} - \sum_{sa} K_{ass'} u_{sa}, \quad s' \in S,$$
$$\frac{\partial E}{\partial u_{sa}} = \left(r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right) - \tau \log \frac{u_{sa}}{\tilde{u}_s}, \quad (s, a) \in S \times A.$$

The diagonal blocks of the second-order derivatives $\frac{\partial^2 E}{\partial v^2}$ and $\frac{\partial^2 E}{\partial u^2}$ are

(2.6)
$$\frac{\partial^2 E}{\partial v_s \partial v_{s'}} = \alpha \delta_{ss'}, \quad (s, s') \in S \times S,$$
$$\frac{\partial^2 E}{\partial u_{sa} \partial u_{s'a'}} = -\tau \delta_{ss'} \left(\frac{\delta_{aa'}}{u_{sa}} - \frac{1}{\tilde{u}_s} \right), \quad (s, s', a, a') \in S^2 \times A^2.$$

Of the two diagonal blocks above, $\frac{\partial^2 E}{\partial v^2}$ is easy to invert since it is diagonal with positive diagonal entries, whereas $\frac{\partial^2 E}{\partial u^2}$ is the sum of a diagonal part and a low-rank part. In the natural gradient dynamics below, we only keep the first part of $\frac{\partial^2 E}{\partial u^2}$,

namely $-\tau\delta_{ss'}\delta_{aa'}/u_{sa}$ (or more compactly $-\tau\text{diag}(1/u)$ in the matrix form). The resulting NGAD flow is

$$\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = -\frac{1}{\alpha}\left(\alpha v_{s'} - \sum_{sa}K_{ass'}u_{sa}\right), \quad s' \in S,$$

$$\frac{\mathrm{d}u_{sa}}{\mathrm{d}t} = -\frac{1}{\tau}u_{sa}\left(\tau\log\frac{u_{sa}}{\tilde{u}_s} - \left(r_{sa} - \sum_{s'}K_{ass'}v_{s'}\right)\right), \quad (s,a) \in S \times A,$$

or equivalently,

$$(2.7) \qquad \begin{aligned} \frac{\mathrm{d}v_{s'}}{\mathrm{d}t} &= -\left(v_{s'} - \frac{1}{\alpha}\sum_{sa}K_{ass'}u_{sa}\right), \quad s' \in S, \\ \frac{\mathrm{d}u_{sa}}{\mathrm{d}t} &= -u_{sa}\left(\log\frac{u_{sa}}{\tilde{u}_s} - \frac{1}{\tau}\left(r_{sa} - \sum_{s'}K_{ass'}v_{s'}\right)\right), \quad (s,a) \in S \times A. \end{aligned}$$

To analyze its convergence, we start by identifying a Lyapunov function of this dynamics. By Theorem 2.1 there is a unique solution (v^*, u^*) to problem (2.2). Based on the solution (v^*, u^*), define

$$(2.8) \qquad L(v,u) = \frac{\alpha}{2}\sum_{s \in S}|v_s - v_s^*|^2 + \tau\sum_{s \in S, a \in A}\left(u_{sa}^*\log\frac{u_{sa}^*}{u_{sa}} + u_{sa} - u_{sa}^*\right).$$

The following lemma summarizes some key properties of $L(v,u)$.

LEMMA 2.3. $L(v,u) \geq 0$ is strictly convex, and the unique minimum is (v^*, u^*), which satisfies $L(v^*, u^*) = 0$. In addition, any sublevel set of L is bounded.

The next lemma states that $L(v,u)$ is a Lyapunov function of (2.7).

LEMMA 2.4. $L(v,u)$ is a Lyapunov function for the dynamics (2.7), i.e., $\frac{\mathrm{d}L}{\mathrm{d}t} \leq 0$ when $\frac{\mathrm{d}v}{\mathrm{d}t}$ and $\frac{\mathrm{d}u}{\mathrm{d}t}$ are defined in (2.7), and the only trajectory of the dynamics (2.7) satisfying $\frac{\mathrm{d}L}{\mathrm{d}t} = 0$ is $(v,u) = (v^*, u^*)$.

The proofs of these two lemmas are given in section 6.

THEOREM 2.5. The dynamics of (2.7) converges globally to (v^*, u^*).

Proof. By Lemma 2.3, Lemma 2.4, and the Barbashin–Krasovskii–LaSalle theorem [14], the dynamics of (2.7) is globally asymptotically stable, which means the NGAD dynamics converges globally to (v^*, u^*). □

To show the exponential convergence of (2.7), we follow Lyapunov's indirect method, i.e., analyzing the linearization of (2.7) at (v^*, u^*) and demonstrating that the real part of the eigenvalues of the corresponding matrix is negative. This result is the content of Theorem 2.6, with the proof given in section 6.

THEOREM 2.6. The dynamics of (2.7) converges at rate $O(e^{-ct})$ to (v^*, u^*) for some $c > 0$.

Below we discuss the implementation of (2.7). By introducing $u_{sa} = \exp(\theta_{sa})$, (2.7) can be rewritten as

(2.9)

$$\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = -\left(v_{s'} - \frac{1}{\alpha} \sum_{sa} K_{ass'} \exp(\theta_{sa}) \right), \quad s' \in S,$$

$$\frac{\mathrm{d}\theta_{sa}}{\mathrm{d}t} = -\left(\theta_{sa} - \log\left(\sum_{a} \exp(\theta_{sa}) \right) - \frac{1}{\tau}\left(r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right) \right), \quad (s,a) \in S \times A.$$

With a learning rate $\eta > 0$, this leads to the update rule

(2.10)

$$v_{s'} \leftarrow (1-\eta)v_{s'} + \frac{\eta}{\alpha} \sum_{sa} K_{ass'} \exp(\theta_{sa}), \quad s' \in S,$$

$$\theta_{sa} \leftarrow (1-\eta)\theta_{sa} + \eta \log\left(\sum_{a} \exp(\theta_{sa}) \right) + \frac{\eta}{\tau}\left(r_{sa} - \sum_{s'} K_{ass'} v_{s'} \right), \quad (s,a) \in S \times A.$$

The details of the algorithm are summarized in Algorithm 2.1.

3. Interpolating natural gradient method. In subsection 2.2, NGAD is introduced using the diagonal part of $\frac{\partial^2 E}{\partial u^2}$. A natural question is whether the whole matrix $\frac{\partial^2 E}{\partial u^2}$ can be used. Under the matrix notation, $\frac{\partial^2 E}{\partial u^2}$ in (2.6) takes the form

(3.1) $$\frac{\partial^2 E}{\partial u^2} = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_{|S|} \end{bmatrix}, \quad H_s = \mathrm{diag}\big((u_{s\cdot})^{-1}\big) - \frac{1}{\tilde{u}_s}\mathbf{1}_{|A|}\mathbf{1}_{|A|}^{\mathsf{T}}, \quad s \in S.$$

Since the Hessian matrix describes the local geometry of the problem, the standard NGAD in subsection 2.2 can be viewed as approximating the Hessian diagonally,

$$\frac{\partial^2 E}{\partial u^2} \approx \begin{bmatrix} \mathrm{diag}\big((u_{1\cdot})^{-1}\big) & & \\ & \ddots & \\ & & \mathrm{diag}\big((u_{|S|\cdot})^{-1}\big) \end{bmatrix},$$

Algorithm 2.1. Standard NGAD for quadratically convexified formulation.

Require: the MDP model $\mathcal{M} = (S, A, P, r, \gamma)$, initialization $(v_{\mathrm{init}}, \theta_{\mathrm{init}})$, convergence threshold ϵ_{tol}, coefficient $\alpha > 0$ for the quadratic term in (2.2), regularization coefficient τ, learning rate η.

1: Initialize the value and parameters $v = v_{\mathrm{init}}$, $\theta = \theta_{\mathrm{init}}$.
2: Calculate $u_{sa} = \exp(\theta_{sa})$, $(s,a) \in S \times A$.
3: Set $q = 1 + \epsilon_{\mathrm{tol}}$.
4: **while** $q > \epsilon_{\mathrm{tol}}$ **do**
5:     Calculate $(v_{\mathrm{new}})_{s'} = (1-\eta)v_{s'} + \frac{\eta}{\alpha} \sum_{sa} K_{ass'} u_{sa}, \quad s' \in S.$
6:     Update θ by

$$\theta_{sa} \leftarrow (1-\eta)\theta_{sa} + \eta \log \sum_{a} u_{sa} + \frac{\eta}{\tau}\left(r_{sa} - \sum_{s'} K_{ass'}(v_{\mathrm{new}})_{s'} \right), \ (s,a) \in S \times A.$$

7:     Calculate $(u_{\mathrm{new}})_{sa} = \exp(\theta_{sa})$, $(s,a) \in S \times A$.
8:     Calculate $q = \max\{\|v_{\mathrm{new}} - v\|/\|v\|, \|u_{\mathrm{new}} - u\|/\|u\|\}$.
9:     Update (v, u) by $v \leftarrow v_{\mathrm{new}}$, $u \leftarrow u_{\mathrm{new}}$.
10: **end while**

---

and using its inverse,

$$
\begin{bmatrix} \text{diag}(u_{1\cdot}) & & \\ & \ddots & \\ & & \text{diag}(u_{|S|\cdot}) \end{bmatrix} \equiv \begin{bmatrix} \tilde{u}_1(\text{diag}(\pi_{1\cdot})) & & \\ & \ddots & \\ & & \tilde{u}_{|S|}(\text{diag}(\pi_{|S|\cdot})) \end{bmatrix},
$$

to precondition the gradient. However, H_s is in fact singular with $\text{Null}(H_s) = \text{Span}(u_{s\cdot})$ and its pseudoinverse reads

$$
\begin{bmatrix} \tilde{u}_1(\text{diag}(\pi_{1\cdot}) - \pi_{1\cdot}\pi_{1\cdot}^{\mathsf{T}}) & & \\ & \ddots & \\ & & \tilde{u}_{|S|}(\text{diag}(\pi_{|S|\cdot}) - \pi_{|S|\cdot}\pi_{|S|\cdot}^{\mathsf{T}}) \end{bmatrix}.
$$

If we had constructed the natural gradient method with this pseudoinverse, the component in the $\mathbf{1}_{|A|}$ direction would not have been updated in the dynamics.

The key idea is that one can interpolate between these two extreme cases; i.e., we propose to use

$$
(3.2) \qquad \begin{bmatrix} \tilde{u}_1(\text{diag}(\pi_{1\cdot}) - c\pi_{1\cdot}\pi_{1\cdot}^{\mathsf{T}}) & & \\ & \ddots & \\ & & \tilde{u}_{|S|}(\text{diag}(\pi_{|S|\cdot}) - c\pi_{|S|\cdot}\pi_{|S|\cdot}^{\mathsf{T}}) \end{bmatrix}
$$

for $0 < c < 1$ to precondition the gradient.

Under this interpolating metric (3.2), the new *interpolating* NGAD (INGAD) is given by

$$
(3.3) \quad
\begin{aligned}
\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} &= -\left(v_{s'} - \frac{1}{\alpha}\sum_{sa} K_{ass'}u_{sa} \right), \quad s' \in S, \\
\frac{\mathrm{d}u_{s\cdot}}{\mathrm{d}t} &= -\tilde{u}_s\left(\text{diag}(\pi_{s\cdot}) - c\pi_{s\cdot}\pi_{s\cdot}^{\mathsf{T}} \right)\left(\log\frac{u_{s\cdot}}{\tilde{u}_s} - \frac{1}{\tau}\left(r_s - \sum_{s'}K_{\cdot ss'}v_{s'} \right) \right), \quad s \in S,
\end{aligned}
$$

where $u_{s\cdot} \in \mathbb{R}^{|A|}$. When $c = 0$, this dynamics reduces to (2.7).

A Lyapunov function of this dynamics can also be identified. Using the unique solution (v^*, u^*) to (2.2), we define

(3.4)

$$
\begin{aligned}
L_c(v, u) = {} & \frac{\alpha}{2}\sum_s |v_s - v_s^*|^2 \\
& + \tau\left(\sum_{sa}\left(u_{sa}^*\log\frac{u_{sa}^*}{u_{sa}} + u_{sa} - u_{sa}^* \right) + \frac{c}{1-c}\sum_s\left(\tilde{u}_s^*\log\frac{\tilde{u}_s^*}{\tilde{u}_s} + \tilde{u}_s - \tilde{u}_s^* \right) \right),
\end{aligned}
$$

where the subscript c denotes the hyperparameter in the function. Some key properties of $L_c(v, u)$ are summarized in the following lemma.

LEMMA 3.1. $L_c(v, u)$ *is convex, and the unique minimum is* $L_c(v^*, u^*) = 0$. *The sublevel sets of* L_c *are bounded.*

The next lemma states that $L_c(v, u)$ is a Lyapunov function for (3.3).

LEMMA 3.2. $L_c(v, u)$ *is a Lyapunov function for the dynamics* (3.3), *i.e.,* $\frac{\mathrm{d}L_c}{\mathrm{d}t} \leq 0$ *when* $\frac{\mathrm{d}v}{\mathrm{d}t}$ *and* $\frac{\mathrm{d}u}{\mathrm{d}t}$ *are defined by* (3.3), *and the only trajectory of the dynamics* (3.3) *satisfying* $\frac{\mathrm{d}L_c}{\mathrm{d}t} = 0$ *is* $(v, u) = (v^*, u^*)$.

The proofs of these two lemmas can be found again in section 6.

THEOREM 3.3. *The dynamics of* (3.3) *converges globally to* (v^*, u^*).

Proof. Similar to Theorem 2.5, by Lemma 3.1, Lemma 3.2, and the Barbashin–Krasovskii–LaSalle theorem [14], the dynamics of (3.3) is globally asymptotically stable and hence converges globally to (v^*, u^*). ∎

The local exponential convergence of (3.3) can also be shown with Lyapunov's indirect method. This result is stated in Theorem 3.4.

THEOREM 3.4. *The dynamics of* (3.3) *converges at rate* $O(e^{-ct})$ *to* (v^*, u^*) *for some* $c > 0$.

Finally, we discuss the implementation of (2.7). By letting $u_{sa} = \exp(\theta_{sa})$, (3.3) can be written as

(3.5)

$$\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = -\left(v_{s'} - \frac{1}{\alpha}\sum_{sa} K_{ass'}\exp(\theta_{sa})\right), \quad s' \in S,$$

$$\frac{\mathrm{d}\theta_{s\cdot}}{\mathrm{d}t} = -\left(I - \frac{c\mathbf{1}\exp(\theta_{s\cdot})^{\mathsf{T}}}{\mathbf{1}^{\mathsf{T}}\exp(\theta_{s\cdot})}\right)\left(\theta_{s\cdot} - \log\sum_a \exp(\theta_{sa})\mathbf{1} - \frac{1}{\tau}\left(r_{s\cdot} - \sum_{s'} K_{\cdot ss'}v_{s'}\right)\right), \quad s \in S.$$

With a learning rate $\eta > 0$, this becomes

$$v_{s'} \leftarrow (1-\eta)v_{s'} + \frac{\eta}{\alpha}\sum_{sa} K_{ass'}\exp(\theta_{sa}), \quad s' \in S,$$

(3.6)
$$\theta_{s\cdot} \leftarrow \theta_{s\cdot} - \eta\left(I - \frac{c\mathbf{1}\exp(\theta_{s\cdot})^{\mathsf{T}}}{\mathbf{1}^{\mathsf{T}}\exp(\theta_{s\cdot})}\right)\left(\theta_{s\cdot} - \log\sum_a \exp(\theta_{sa})\mathbf{1}\right.$$

$$\left. - \frac{1}{\tau}\left(r_{s\cdot} - \sum_{s'} K_{\cdot ss'}v_{s'}\right)\right), \quad s \in S.$$

The details of the algorithm can be found in Algorithm 3.1 below.

4. Numerical results. In this section, we examine the performance of Algorithms 2.1 and 3.1 with several different examples. Subsection 4.1 compares Algorithms 2.1 and 3.1 in a complete-information case where the transition probabilities and the rewards are known exactly. A comparison with an existing method in [42] is showcased in this setting as well. The sample-based setting is investigated in subsection 4.2, where we give an adapted version of INGAD with sample access and test its performance on two different MDPs.

4.1. Experiments with complete information. Here we test the numerical performance of the standard natural gradient in Algorithm 2.1 and the interpolating natural gradient in Algorithm 3.1 in a complete-information situation. The MDP used is from [42], where $|S| = 200$, $|A| = 50$, and the transition probabilities and rewards are randomly generated. More specifically, the transition probabilities are set as $P_{ass'} = 1/20$ for any $s' \in S_{sa}$, where S_{sa} is a uniformly randomly chosen subset of S such that $|S_{sa}| = 20$, and the reward $r_{sa} = U_{sa}U_s$ for $(s, a) \in S \times A$, where U_{sa} and U_s are independently uniformly sampled from $[0, 1]$.

Algorithm 3.1. INGAD for quadratically convexified formulation.

Require: the MDP model $\mathcal{M} = (S, A, P, r, \gamma)$, initialization $(v_{\text{init}}, \theta_{\text{init}})$,
   convergence threshold ϵ_{tol}, coefficient $\alpha > 0$ for the quadratic term in (2.2),
   regularization coefficient τ, metric coefficient $0 \leq c < 1$, learning rate η.
1: Initialize the value and parameters $v = v_{\text{init}}$, $\theta = \theta_{\text{init}}$.
2: Calculate $u_{sa} = \exp(\theta_{sa})$, $(s, a) \in S \times A$.
3: Set $q = 1 + \epsilon_{\text{tol}}$.
4: **while** $q > \epsilon_{\text{tol}}$ **do**
5:    Calculate $(v_{\text{new}})_{s'} = (1 - \eta)v_{s'} + \frac{\eta}{\alpha} \sum_{sa} K_{ass'} u_{sa}, \quad s' \in S$.
6:    Update θ by

$$\theta_{s\cdot} \leftarrow \theta_{s\cdot} - \eta \left(I - \frac{c\mathbf{1}u_{s\cdot}^\mathsf{T}}{\mathbf{1}^\mathsf{T} u_{s\cdot}} \right) \left(\theta_{s\cdot} - \left(\log \sum_a u_{sa} \right) \mathbf{1} \right.$$
$$\left. - \frac{1}{\tau} \left(r_{s\cdot} - \sum_{s'} K_{\cdot ss'}(v_{\text{new}})_{s'} \right) \right), \, s \in S.$$

7:    Calculate $(u_{\text{new}})_{sa} = \exp(\theta_{sa})$, $(s, a) \in S \times A$.
8:    Calculate $q = \max\{\|v_{\text{new}} - v\|/\|v\|, \|u_{\text{new}} - u\|/\|u\|\}$.
9:    Update (v, u) by $v \leftarrow v_{\text{new}}$, $u \leftarrow u_{\text{new}}$.
10: **end while**

A comparison of Algorithms 2.1 and 3.1 is carried out using the same discount rate $\gamma = 0.99$ and hyperparameters $(\epsilon_{\text{tol}}, \alpha, \tau) = (1 \times 10^{-5}, 0.1, 0.01)$. Since both algorithms are explicit discretizations of the corresponding flow, a sufficiently small learning rate is needed to ensure convergence. In the tests, the learning rates are set as $\eta = 3 \times 10^{-4}$ for Algorithm 2.1 and $\eta = 8 \times 10^{-3}$ for Algorithm 3.1, which are both manually tuned to be close to the largest learning rates such that convergence is achieved. For Algorithm 3.1, we set $c = 0.98$.

As a result, Algorithm 2.1 takes 59296 iterations to converge while Algorithm 3.1 takes 2213 iterations, demonstrating that the interpolating metric introduced in section 3 gives rise to an acceleration of more than 1 magnitude. Plotted in Figures 2(a) and 2(c) are the errors of the value and policy with respect to the ground truth in the training process, which verifies that Algorithm 3.1 achieves the same precision more than a magnitude faster than Algorithm 2.1. Moreover, it can be observed from Figures 2(b) and 2(d) that the Lyapunov function decreases monotonically in both cases, confirming the theoretical analyses in sections 2 and 3.

Comparison with PMD [42]. Next, we compare the performance of Algorithm 3.1 (INGAD) with an existing method, namely the policy mirror descent (PMD) method used in [42]. The underlying MDP of the problem is the same as in subsection 4.1. For the hyperparameters of INGAD, we take $(N_{\text{iter}}, \alpha, c) = (2000, 0.1, 0.98)$. In order to make a fair comparison, the learning rate is set as $\eta = 8 \times 10^{-3}$, and the regularization coefficient is set as $\tau = 0.01$ for both methods. For the PMD method, we take the first 20000 iterations.

It can be seen from Figure 3 that Algorithm 3.1 admits a faster convergence than PMD. For both the value function and the policy, Algorithm 3.1 achieves a higher precision in 2000 iterations than PMD with 20000 iterations. The final errors
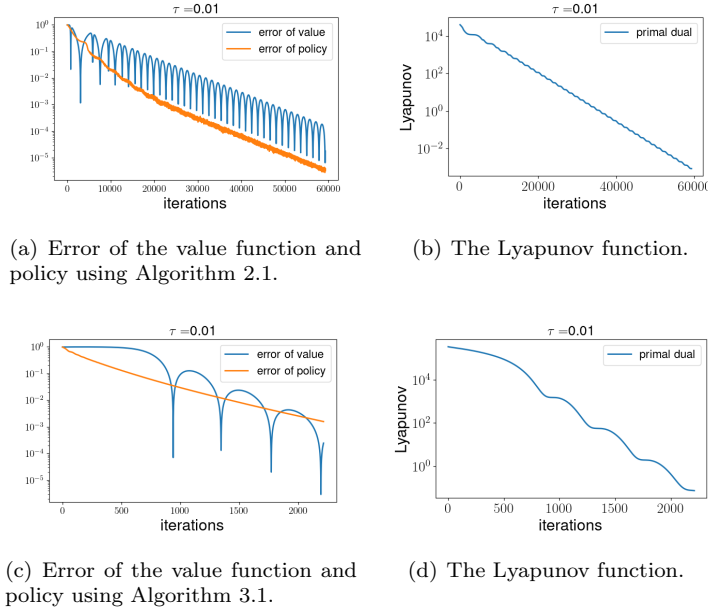
(a) Error of the value function and policy using Algorithm 2.1.

(b) The Lyapunov function.

(c) Error of the value function and policy using Algorithm 3.1.

(d) The Lyapunov function.

FIG. 2. *Comparison of Algorithms* 2.1 *and* 3.1. (a) *Convergence of the value and policy during training of Algorithm* 2.1; (b) *Lyapunov function* (2.8); (c) *Convergence of the value and policy during training of Algorithm* 3.1; (d) *Lyapunov function* (3.4). *Blue curves in* (a) *and* (c): *The convergence of* $\|\pi - \pi^*\|_F / \|\pi^*\|_F$ *in the training process. Orange curves in* (a) *and* (c): *The convergence of* $\|v - v^*\|_2 / \|v^*\|_2$ *in the training process. A logarithmic scale is used for all vertical axes. (Color figure available online.)*
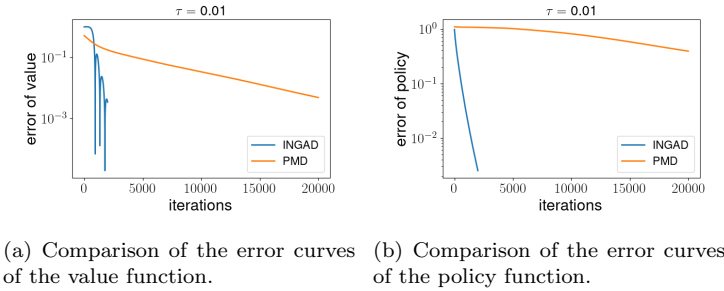
(a) Comparison of the error curves of the value function.

(b) Comparison of the error curves of the policy function.

FIG. 3. *Comparison of Algorithm* 3.1 *with PMD* [42]. (a) *Convergence of the value function;* (b) *Convergence of the policy. Blue curves: The convergence of* $\|\pi - \pi^*\|_F / \|\pi^*\|_F$ *in the training process. Orange curves: The convergence of* $\|v - v^*\|_2 / \|v^*\|_2$ *in the training process. A logarithmic scale is used for all vertical axes. (Color figure available online.)*

in the value function and policy are approximately $(0.0034, 0.0025)$ for INGAD and $(0.39, 0.0049)$ for PMD.

4.2. Experiments with random samples. Finally, we test the INGAD algorithm on the case where the transition probabilities are unknown. In each iteration, a size-N_b batch of samples is used to estimate the transition probabilities and used for the INGAD update, as presented in Algorithm 4.1. In order to stabilize the training dynamics, we use a decaying learning rate starting with η_{init} and ending with η_{end}. If $\eta_{\text{init}} = \eta_{\text{end}}$, then the algorithm reduces to the constant learning rate case. We first use the MDP introduced in subsection 4.1.

Algorithm 4.1. INGAD for quadratically convexified formulation (sample version).

Require: the discount rate γ, initialization $(v_{\text{init}}, \theta_{\text{init}})$, convergence threshold ϵ_{tol}, maximum number of iterations N_{iter}, coefficient $\alpha > 0$ for the quadratic term in (2.2), regularization coefficient τ, metric coefficient $0 \leq c < 1$, the initial and the final learning rate $(\eta_{\text{init}}, \eta_{\text{end}})$.

1: Initialize the value and parameters $v = v_{\text{init}}$, $\theta = \theta_{\text{init}}$.
2: Calculate $u_{sa} = \exp(\theta_{sa})$, $(s, a) \in S \times A$.
3: Set $q = 1 + \epsilon_{\text{tol}}$ and $i = 0$.
4: Initialize a buffer $\mathcal{B}$ with N transition samples (s, a, s', r).
5: **while** $q > \epsilon_{\text{tol}}$ and $i < N_{\text{iter}}$ **do**
6:    Calculate $\eta_i = (1 + i(N_{\text{iter}}\eta_{\text{end}})^{-1}(\eta_{\text{init}} - \eta_{\text{end}}))^{-1}\eta_{\text{init}}$.
7:    Randomly sample a batch of samples from $\mathcal{B}$ with size N_b.
8:    Estimate $\hat{K}^{(i)}$ from the samples.
9:    Calculate $(v_{\text{new}})_{s'} = (1 - \eta)v_{s'} + \frac{\eta}{\alpha}\sum_{sa}\hat{K}^{(i)}_{ass'}u_{sa}, \quad s' \in S$.
10:   Update θ by

$$\theta_{s\cdot} \leftarrow \theta_{s\cdot} - \eta\left(I - \frac{c\mathbf{1}u_{s\cdot}^{\mathsf{T}}}{\mathbf{1}^{\mathsf{T}}u_{s\cdot}}\right)\left(\theta_{s\cdot} - (\log\sum_a u_{sa})\mathbf{1}\right.$$

$$\left. - \frac{1}{\tau}\left(r_{s\cdot} - \sum_{s'}\hat{K}^{(i)}_{\cdot ss'}(v_{\text{new}})_{s'}\right)\right), \; s \in S.$$

11:   Calculate $(u_{\text{new}})_{sa} = \exp((\theta_{\text{new}})_{sa})$, $(s, a) \in S \times A$.
12:   Calculate $q = \max\{\|v_{\text{new}} - v\|/\|v\|, \|u_{\text{new}} - u\|/\|u\|\}$.
13:   Update (v, u) by $v \leftarrow v_{\text{new}}$, $u \leftarrow u_{\text{new}}$.
14:   $i \leftarrow i + 1$.
15: **end while**

(a) Error of the value function and policy.
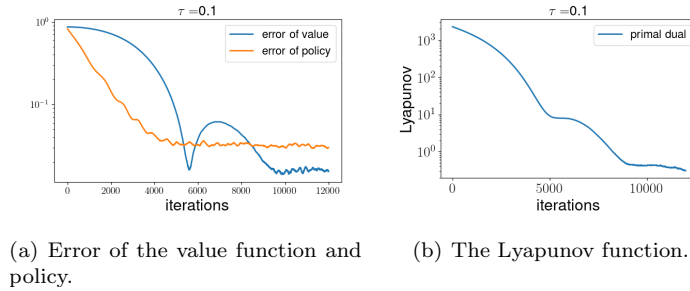
(b) The Lyapunov function.

FIG. 4. *Performance of Algorithm 4.1 for the MDP problem described in subsection 4.1. (a) Convergence of the value and policy during training of Algorithm 4.1; blue curve: the convergence of $\|\pi - \pi^*\|_F/\|\pi^*\|_F$ in the training process; orange curve: the convergence of $\|v - v^*\|_2/\|v^*\|_2$ in the training process. (b) Lyapunov function (3.4). A logarithmic scale is used for all vertical axes. (Color figure available online.)*

In this experiment, we adopt $(\gamma, N_{\text{iter}}, N_b, \alpha, \tau, c) = (0.9, 12000, 1\times10^5, 0.1, 0.1, 0.9)$ and $(\eta_{\text{init}}, \eta_{\text{end}}) = (0.001, 0.001)$. Altogether 1×10^8 samples are used in the training process.

It can be seen from Figure 4 that the approximate value function and policy given by Algorithm 4.1 converge to the ground truth and oscillate around it at the final stage. The final errors in the value function and policy are approximately 0.015 and 0.030, respectively. It can also be seen from Figure 4(b) that the Lyapunov function

mostly decreases in the training process even though the transition probabilities used are just unbiased estimators of the ground truth.

Experiment with the FrozenLake environment. In this part, the MDP we consider is from the FrozenLake environment (see [6]). The environment describes the problem where the player aims to walk on a frozen lake from one corner to another without falling into the holes. In the example we use below, the map is an 8×8 square grid with 10 randomly generated holes. Therefore, the size of the state space is 64, and there are 4 actions, corresponding to the 4 directions one can choose at each position. In order to model the low-friction property of ice, the transition is not deterministic. More specifically, the agent has a $1/3$ probability of moving in the intended direction or the two perpendicular directions. An illustration of the lake map is given in Figure 5.

In the numerical experiment, we set $(\gamma, N_{\text{iter}}, N_b, \alpha, \tau, c) = (0.9, 80000, 2000, 0.1, 0.1, 0.9)$ and $(\eta_{\text{init}}, \eta_{\text{end}}) = (0.002, 0.0002)$. The buffer size N and the batch-size N_b are chosen as 2×10^6 and 2000, respectively.

Similar to the previous example, both the error of the value function and the error of the policy function reduce in the training process, indicating the effectiveness of Algorithm 4.1 given sample access to the MDP (see Figure 6). The oscillations represent the randomness in the samples gathered in each batch. The final errors for the value and policy are 0.012 and 0.026, respectively. The Lyapunov function also shows a clear decreasing trend along the training process.
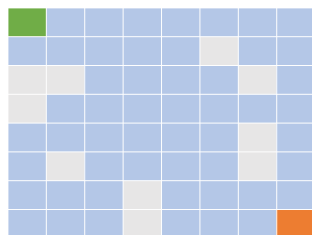

FIG. 5. *Map of the FrozenLake environment with size* 8×8 *and* 10 *randomly generated holes. The green and the orange boxes represent the starting position and the target position, respectively. The blue area represents the positions with ice, while the gray spots indicate the positions of holes. (Color figure available online.)*
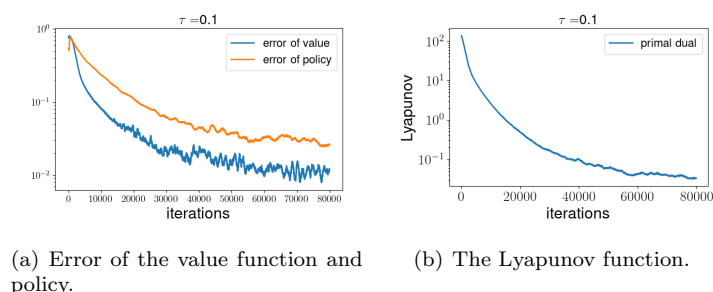

(a) Error of the value function and policy.

(b) The Lyapunov function.

FIG. 6. *Performance of Algorithm* 4.1 *for the* 8×8 *FrozenLake problem.* (a) *Convergence of the value and policy during training of Algorithm 4.1; blue curve: the convergence of* $\|\pi - \pi^*\|_F / \|\pi^*\|_F$ *in the training process; orange curve: the convergence of* $\|v - v^*\|_2 / \|v^*\|_2$ *in the training process.* (b) *Lyapunov function* (3.4). *A logarithmic scale is used for all vertical axes. (Color figure available online.)*

5. Conclusion and discussion. In this paper, we focused on the primal-dual formulation of entropy regularized Markov decision problems. We proposed a quadratically convexified primal-dual formulation that makes the landscape of the objective function smoother and enables faster numerical algorithms. We proved the equivalence of the quadratically convexified primal-dual formulation with the original primal-dual formulation. Leveraging the enhanced convexity of the objective function, we proposed an NGAD method and proved its convergence properties using the Lyapunov methods. We further introduced an INGAD algorithm that accelerates convergence significantly. The efficiency and robustness of the proposed algorithms are demonstrated through multiple numerical experiments.

For future directions, one can potentially extend the convergence analysis here to the finite sample case with standard statistical methods. Another interesting direction to explore is the application of other optimization techniques to the convexified formulation proposed here.

6. Proofs.

6.1. Proof of Theorem 2.1. First, we show that there exists a unique solution to (2.1). By [12], there exist a unique optimal policy π^* and a unique optimal value function $v^* = v_{\pi^*}$ such that (1.5) or, equivalently, (1.7) and (1.8) hold. From [41], we know that this optimal value function and policy (v^*, π^*) also yield a solution (v^*, u°) to the primal-dual problem by $u_{sa}^\circ = \pi_{sa}^*(K_{\pi^*}^{-\mathsf{T}}e)_s$. Also from [41] we know that any solution to the primal-dual formulation (2.1) satisfies $v = v^*$, $u_{sa}/\tilde{u}_s = \pi_{sa}^*, (s,a) \in S \times A$, and $\tilde{u} = K_{\pi^*}^{-\mathsf{T}}e$, which combined with the uniqueness of (v^*, π^*) shows that the solution (v^*, u°) to (2.1) is unique.

Next, we show that (v^*, u^*) satisfies the first-order condition of (2.2), where

$$(6.1) \qquad u_{sa}^* := \frac{w_s}{\tilde{u}_s^\circ} u_{sa}^\circ, \quad w := \alpha K_{\pi^*}^{-\mathsf{T}}v^*.$$

The first-order condition of (2.1) gives

$$(6.2) \qquad \begin{aligned} e_{s'} - \sum_{sa} K_{ass'} u_{sa}^\circ &= 0 \quad \forall s' \in S, \\ r_{sa} - \sum_{s'} K_{ass'} v_{s'}^* - \tau \log(u_{sa}^\circ/\tilde{u}_s^\circ) &= 0 \quad \forall (s,a) \in S \times A. \end{aligned}$$

Since $e_{s'} = \sum_{sa} K_{ass'} u_{sa}^\circ = \sum_s \sum_a K_{ass'} \pi_{sa}^* \tilde{u}_s^\circ = \sum_s K_{\pi^* ss'} \tilde{u}_s^\circ$, we have $\tilde{u}^\circ = K_{\pi^*}^{-\mathsf{T}}e = e + \sum_{k=1}^\infty \gamma^k (P_{\pi^*}^{\mathsf{T}})^k e$, and thus $\tilde{u}_s^\circ \geq e_s > 0$ for all $s \in S$. Similarly, $w_s \geq \alpha v_s^*$ for all $s \in S$ since $w = \alpha K_{\pi^*}^{-\mathsf{T}}v^*$. By (1.8), it is also known that $\pi_{sa}^* > 0$ for all $(s,a) \in S \times A$, so $(r_{\pi^*} - \tau h_{\pi^*})_s > 0$ for all $s \in S$ since r is nonnegative. Again by an expansion of $K_{\pi^*}^{-1}$, one can show that $v_s^* = K_{\pi^*}^{-1}(r_{\pi^*} - \tau h_{\pi^*}) \geq (r_{\pi^*} - \tau h_{\pi^*})_s > 0$. Hence $u_{sa}^* = \frac{w_s}{\tilde{u}_s^\circ} u_{sa}^\circ > 0$ is well defined. In addition, $\tilde{u}^* = w$ and

$$(6.3) \qquad \frac{u_{sa}^*}{\tilde{u}_{sa}^*} = \frac{u_{sa}^\circ}{\tilde{u}_s^\circ} = \pi_{sa}^*.$$

As a result,

$$(6.4) \qquad r_{sa} - \sum_{s'} K_{ass'} v_{s'}^* = \tau \log \frac{u_{sa}^\circ}{\tilde{u}_s^\circ} = \tau \log \frac{u_{sa}^*}{\tilde{u}_{sa}^*} \quad \forall (s,a) \in S \times A.$$

Moreover, one can show that

(6.5)

$$\alpha v_{s'}^* = K_{\pi^*s s'}^\mathsf{T} w = \sum_s K_{\pi^* s s'} \frac{u_{sa}^*}{\pi_{sa}^*} = \sum_s \left(\sum_a K_{ass'} \pi_{sa}^* \right) \frac{u_{sa}^*}{\pi_{sa}^*} = \sum_{sa} K_{ass'} u_{sa}^* \quad \forall s' \in S.$$

Combining (6.4) and (6.5), we conclude that (v^*, u^*) is a solution to

(6.6)
$$\alpha v_{s'} - \sum_{sa} K_{ass'} u_{sa} = 0 \quad \forall s' \in S,$$

$$r_{sa} - \sum_{s'} K_{ass'} v_{s'} - \tau \log \frac{u_{sa}}{\tilde{u}_s} = 0 \quad \forall (s,a) \in S \times A.$$

This is the first-order stationary condition for the problem (2.2).

Finally, we show that (v^*, u^*) is the unique solution to (6.6). Assume that (v^1, u^1) and (v^2, u^2) are both solutions to (6.6). If $v^1 \neq v^2$, then $E(v^1, u^1) < E(v^2, u^1)$ and $E(v^2, u^2) < E(v^1, u^2)$ since for any u, $E(v, u)$ is strictly convex in v. On the other hand, for any v, $E(v, u)$ is concave in u (see, for example, [27] or [41]). So $E(v^1, u^1) \geq E(v^1, u^2)$ and $E(v^2, u^2) \geq E(v^2, u^1)$ and

$$E(v^1, u^1) \geq E(v^1, u^2) > E(v^2, u^2) \geq E(v^2, u^1) > E(v^1, u^1),$$

which is a contradiction, so we must have $v^1 = v^2$ instead. By the second equation in (6.6),

$$\frac{u_{sa}^1}{\tilde{u}_s^1} = \exp\left(\tau^{-1} \left(r_{sa} - \sum_{s'} K_{ass'} v_{s'}^1 \right) \right) = \exp\left(\tau^{-1} \left(r_{sa} - \sum_{s'} K_{ass'} v_{s'}^2 \right) \right) = \frac{u_{sa}^1}{\tilde{u}_s^1};$$

thus $\pi^1 = \pi^2$, where $\pi_{sa}^1 = \frac{u_{sa}^1}{\tilde{u}_s^1}$, $\pi_{sa}^2 = \frac{u_{sa}^2}{\tilde{u}_s^2}$. Since $(\pi^1, v^1) = (\pi^2, v^2)$, by the first equation in (6.6),

$$\tilde{u}^1 = \alpha K_{\pi_1}^{-\mathsf{T}} v^1 = \alpha K_{\pi_2}^{-\mathsf{T}} v^2 = \tilde{u}^2.$$

As a result,

$$u_{sa}^1 = \tilde{u}_s^1 \cdot \frac{u_{sa}^1}{\tilde{u}_s^1} = \tilde{u}_s^2 \cdot \frac{u_{sa}^2}{\tilde{u}_s^2} = u_{sa}^2 \quad \forall (s,a) \in S \times A, \qquad \square$$

and $(v^1, u^1) = (v^2, u^2)$. Hence the solution to (6.6) is unique. Therefore, (v^*, u^*) is the unique solution to (6.6). By (6.3), the policy yielded by u^* coincides with the optimal policy π^*, which finishes the proof.

6.2. Proof of Lemma 2.3. From the definition of L we know that $\frac{\partial^2 L}{\partial u_{sa} \partial v_{s'}} = 0$. Moreover,

$$\frac{\partial^2 L}{\partial v_s \partial v_{s'}} = \alpha \delta_{ss'}, \quad \frac{\partial^2 L}{\partial u_{sa} \partial u_{s'a'}} = \tau \delta_{(s,a),(s',a')} \frac{u_{sa}^*}{u_{sa}^2}, \quad (s, s', a, a') \in S^2 \times A^2,$$

which means that the Hessian matrix of L is a diagonal matrix with positive diagonal elements on the domain $\mathbb{R}^{|S|} \times \mathbb{R}_+^{|S| \times |A|}$. Hence L is strictly convex. Since the first-order condition

(6.7) $$\frac{\partial L}{\partial v_s} = \alpha(v_s - v_s^*) = 0, \quad \frac{\partial L}{\partial u_{sa}} = \tau \left(1 - \frac{u_{sa}^*}{u_{sa}} \right) = 0, \quad (s,a) \in S \times A,$$

has a unique solution $(v, u) = (v^*, u^*)$, it is also the unique global minimum of L. Let

$$\varphi_s(x) = \frac{1}{2}\alpha|x - v_s^*|^2, \quad \psi_{sa}(x) = \tau(u_{sa}^* \log u_{sa}^*/x + x - u_{sa}^*).$$

By the calculation above, one can also show that φ_s and ψ_{sa} are strictly convex and nonnegative. Moreover, since $\lim_{x \to +\infty} \psi_{sa}(x) = +\infty$, we have $M(C) = \max_{sa} \sup\{x > 0 \,|\, \psi_{sa}(x) \le C\} < +\infty$. As a result, the sublevel set

$$\{(v, u) \,|\, L(v, u) \le C\} \subset \{(v, u) \,|\, |v_{s'} - v_{s'}^*|$$
$$\le \sqrt{2C/\alpha}, 0 < u_{sa} < M(C), s' \in S, (s, a) \in S \times A\}$$

is bounded. $\qquad\square$

6.3. Proof of Lemma 2.4. We first prove the following lemma.

LEMMA 6.1. *Define $H : \mathbb{R}_+^{|A|} \to \mathbb{R}$ by $H(z) = \sum_a z_a \log z_a - \bar{z} \log \bar{z}$, where $\bar{z} = \sum_a z_a$, then H is convex. Moreover, $(z_1 - z_2) \cdot (\nabla H(z_1) - \nabla H(z_2)) \ge 0$, and the equality is achieved if and only if $z_2 = cz_1$ for some $c > 0$.*

Proof. The second-order derivatives of H read $\frac{\partial^2 H}{\partial z_a \partial z_{a'}} = \frac{\delta_{aa'}}{z_a} - \frac{1}{\bar{z}}$. By the Cauchy–Schwarz inequality, for any $x \in \mathbb{R}^{|A|}$

$$\sum_{aa'} x_a \frac{\partial^2 H}{\partial z_a \partial z_{a'}} x_{a'} = \sum_{aa'} x_a \left(\frac{\delta_{aa'}}{z_a} - \frac{1}{\bar{z}}\right) x_{a'} = \sum_a \frac{x_a^2}{z_a} - \frac{1}{\bar{z}} \left(\sum_a x_a\right)^2 \ge 0.$$

Hence the Hessian matrix of H is positive semidefinite and H is convex. By convexity $(z_1 - z_2) \cdot (\nabla H(z_1) - \nabla H(z_2)) \ge 0$. Suppose now that equality holds. If $z_1 = z_2$, then clearly $z_2 = cz_1$ for $c = 1$. If $z_1 \ne z_2$, let $h(t) = H(z_1 + t(z_2 - z_1))$; then h is also convex and $h'(0) = (z_2 - z_1) \cdot \nabla H(z_1) = (z_2 - z_1) \cdot \nabla H(z_2) = h'(1)$, so $h'(t) = h'(0)$ for any $t \in [0, 1]$, and thus

$$0 = h''(0) = (z_2 - z_1)^\mathsf{T} \nabla^2 H(z_1)(z_2 - z_1).$$

Hence from the equality condition of the Cauchy–Schwarz inequality, we conclude $z_2 - z_1 = \tilde{c}z_1$ and thus $z_2 = cz_1$ for some c, and we have $c > 0$ since $z_1, z_2 \in \mathbb{R}_+^{|A|}$. $\qquad\square$

Proof of Lemma 2.4. By Theorem 2.1, (v^*, u^*) is also the unique solution to (6.6), so

$$\alpha v_{s'}^* - \sum_{sa} K_{ass'} u_{sa}^* = 0, \quad s' \in S,$$

(6.8)

$$\left(r_{sa} - \sum_{s'} K_{ass'} v_{s'}^*\right) - \tau \log \frac{u_{sa}^*}{\tilde{u}_s^*} = 0, \quad (s, a) \in S \times A.$$

Subtracting this from the dynamics (2.7) leads to

(6.9)

$$\frac{dv_{s'}}{dt} = -\left((v_{s'} - v_{s'}^*) - \frac{1}{\alpha} \sum_{sa} K_{ass'}(u_{sa} - u_{sa}^*)\right), \quad s' \in S,$$

$$\frac{du_{sa}}{dt} = -u_{sa}\left(\left(\log \frac{u_{sa}}{\tilde{u}_s} - \log \frac{u_{sa}^*}{\tilde{u}_s^*}\right) + \frac{1}{\tau} \sum_{s'} K_{ass'}(v_{s'} - v_{s'}^*)\right), \quad (s, a) \in S \times A.$$

Taking the derivative of L gives

$$
\begin{aligned}
\frac{\mathrm{d}L}{\mathrm{d}t} = &-\alpha \sum_{s'}(v_{s'}-v_{s'}^*)\left((v_{s'}-v_{s'}^*)-\frac{1}{\alpha}\sum_{sa}K_{ass'}(u_{as}-u_{as}^*)\right)\\
(6.10)\qquad &-\tau\sum_{sa}\frac{u_{sa}-u_{sa}^*}{u_{sa}}\cdot u_{sa}\left(\left(\log\frac{u_{sa}}{\tilde{u}_s}-\log\frac{u_{sa}^*}{\tilde{u}_s^*}\right)+\frac{1}{\tau}\sum_{s'}K_{ass'}(v_{s'}-v_{s'}^*)\right)\\
= &-\alpha\sum_{s'}(v_{s'}-v_{s'}^*)^2-\tau\sum_{sa}(u_{sa}-u_{sa}^*)\left(\log\frac{u_{sa}}{\tilde{u}_s}-\log\frac{u_{sa}^*}{\tilde{u}_s^*}\right),
\end{aligned}
$$

where we have used (6.7). By Lemma 6.1,

$$
\sum_{sa}(u_{sa}-u_{sa}^*)(\log u_{sa}/\tilde{u}_s-\log u_{sa}^*/\tilde{u}_s^*)=\sum_s(u_s-u_s^*)\cdot(\nabla H(u_s)-\nabla H(u_s^*))\geq 0,
$$

where H is defined in Lemma 6.1. Therefore,

$$
-\alpha\sum_{s'}(v_{s'}-v_{s'}^*)^2-\tau\sum_{sa}(u_{sa}-u_{sa}^*)\left(\log\frac{u_{sa}}{\tilde{u}_s}-\log\frac{u_{sa}^*}{\tilde{u}_s^*}\right)\leq 0.
$$

By Lemma 6.1 the equality holds only when $v=v^*$ and $u_{sa}=c_s u_{sa}^*$ for $c_s>0$, $(s,a)\in S\times A$. Let

$$
\mathcal{R}=\left\{(v,u)\mid-\alpha\sum_{s'}(v_{s'}-v_{s'}^*)^2-\tau\sum_{sa}(u_{sa}-u_{sa}^*)(\log u_{sa}/\tilde{u}_s-\log u_{sa}^*/\tilde{u}_s^*)=0\right\};
$$

then $\mathcal{R}=\{(v,u)\mid v=v^*,u_{sa}=c_s u_{sa}^*,c_s\in\mathbb{R}_+,s\in S\}$. We proceed to prove that the only trajectory of (6.9) in $\mathcal{R}$ is $(v,u)=(v^*,u^*)$. Since $v=v^*$ for any $(v,u)\in\mathcal{R}$, $\frac{\mathrm{d}v_{s'}}{\mathrm{d}t}=0$ for any $s'\in S$. The equality

$$
\begin{aligned}
(6.11)\qquad 0=&\sum_{sa}K_{ass'}(u_{sa}-u_{sa}^*)=\sum_{sa}K_{ass'}(c_s-1)u_{sa}^*\\
=&\sum_{sa}K_{ass'}(c_s-1)\tilde{u}_s^*\pi_{sa}^*=\sum_s K_{\pi^*ss'}(c_s-1)\tilde{u}_s^*
\end{aligned}
$$

means that, for any point (v,u) on the trajectory of (6.9) in $\mathcal{R}$, $K_{\pi^*}^{\mathsf{T}}((c-1)\tilde{u}^*)=0$. Here $(c-1)\tilde{u}^*$ is the vector with length $|S|$ whose sth element is $(c_s-1)\tilde{u}_s^*$. Thus $c_s=1$ for any $s\in S$, and the trajectory is a single point $(v,u)=(v^*,u^*)$. $\qquad\square$

6.4. Proof of Theorem 2.6. The linearized dynamic of the standard natural gradient (2.7) is

(6.12)

$$
\begin{aligned}
\frac{\mathrm{d}v_{s'}}{\mathrm{d}t}=&-\left((v_{s'}-v_{s'}^*)-\frac{1}{\alpha}\sum_{sa}K_{ass'}(u_{sa}-u_{sa}^*)\right),\quad s'\in S,\\
\frac{\mathrm{d}u_{sa}}{\mathrm{d}t}=&-u_{sa}^*\left(\frac{1}{\tau}\sum_{s'}K_{ass'}(v_{s'}-v_{s'}^*)+\frac{u_{sa}-u_{sa}^*}{u_{sa}^*}-\frac{\tilde{u}_s-\tilde{u}_s^*}{\tilde{u}_s^*}\right),\quad (s,a)\in S\times A.
\end{aligned}
$$

Define matrix $\tilde{K}$ by $\tilde{K}_{(s-1)|A|+a,s'}=K_{ass'}$, and let $\delta v=v-v^*$, $\delta u=u-u^*$. Then (6.12) becomes

$$
(6.13)\qquad\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\delta v\\\delta u\end{bmatrix}=-\begin{bmatrix}I_{|S|}&-\frac{1}{\alpha}\tilde{K}^{\mathsf{T}}\\\frac{1}{\tau}\mathrm{diag}(u^*)\tilde{K}&\mathrm{diag}(u^*)M\end{bmatrix}\begin{bmatrix}\delta v\\\delta u\end{bmatrix},
$$

where $\mathrm{diag}(u^*)$ is a diagonal matrix whose $((s-1)|A|+a)$th diagonal element is u^*_{sa}. Here M is a block-diagonal matrix defined as

$$(6.14) \qquad M = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_{|S|} \end{bmatrix}, \quad M_s = \mathrm{diag}\big((u^*_s)^{-1}\big) - \frac{1}{\tilde{u}^*_s}\mathbf{1}_{|A|}\mathbf{1}^{\mathsf{T}}_{|A|}, \quad s \in S,$$

where $\mathrm{diag}\big((u^*_s)^{-1}\big)$ is a diagonal $|A| \times |A|$ matrix with the ath diagonal element equal to $1/u^*_{sa}$. Notice that M is symmetric and by the Cauchy–Schwarz inequality, for any $x \in \mathbb{R}^{|A|}$

$$(6.15) \qquad \sum_{aa'} x_a M_s x_{a'} = \sum_a \frac{x_a^2}{u^*_{sa}} - \frac{1}{\bar{u}^*_s}\left(\sum_a x_a\right)^2 \geq 0, \quad s \in S.$$

Hence M_s is positive semidefinite for all s and thus M is also positive semidefinite. Define invertible matrix P as

$$P = \begin{bmatrix} \sqrt{\tau}I_{|S|} & \\ & \sqrt{\alpha}\mathrm{diag}\big(\sqrt{u^*}\big) \end{bmatrix},$$

where $\mathrm{diag}\big(\sqrt{u^*}\big)$ is a diagonal $|S||A| \times |S||A|$ matrix with the $((s-1)|A|+a)$th diagonal element equal to $\sqrt{u^*_{sa}}$. Denote the matrix in the linearized dynamics (6.13) as $-J$, i.e.,

$$J = \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha}\tilde{K}^{\mathsf{T}} \\ \frac{1}{\tau}\mathrm{diag}(u^*)\tilde{K} & \mathrm{diag}(u^*)M \end{bmatrix}.$$

Then

$$P^{-1}JP$$
$$= \begin{bmatrix} \frac{1}{\sqrt{\tau}}I_{|S|} & \\ & \frac{1}{\sqrt{\alpha}}\mathrm{diag}\big((\sqrt{u^*})^{-1}\big) \end{bmatrix} \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha}\tilde{K}^{\mathsf{T}} \\ \frac{1}{\tau}\mathrm{diag}(u^*)\tilde{K} & \mathrm{diag}(u^*)M \end{bmatrix} \begin{bmatrix} \sqrt{\tau}I_{|S|} & \\ & \sqrt{\alpha}\mathrm{diag}\big(\sqrt{u^*}\big) \end{bmatrix}$$
$$= \begin{bmatrix} I_{|S|} & -\frac{1}{\sqrt{\alpha\tau}}\tilde{K}^{\mathsf{T}}\mathrm{diag}\big(\sqrt{u^*}\big) \\ \frac{1}{\sqrt{\alpha\tau}}\mathrm{diag}\big(\sqrt{u^*}\big)\tilde{K} & \mathrm{diag}\big(\sqrt{u^*}\big)M\mathrm{diag}\big(\sqrt{u^*}\big) \end{bmatrix}.$$

It suffices to show that the real part of the eigenvalues of $P^{-1}JP$ is positive. Denote $P^{-1}JP$ by $\tilde{J}$. Using the positive semidefiniteness of M, for any eigenpair (λ, x) of $\tilde{J}$ we can deduce

$$
\begin{aligned}
(6.16) \quad \mathrm{Re}(\lambda) &= \frac{1}{2}\left(\frac{x^{\mathrm{H}}\tilde{J}x}{x^{\mathrm{H}}x} + \frac{x^{\mathrm{H}}\tilde{J}^{\mathrm{H}}x}{x^{\mathrm{H}}x}\right) \\
&= \frac{1}{2x^{\mathrm{H}}x}x^{\mathrm{H}}\Bigg(\begin{bmatrix} I_{|S|} & -\frac{1}{\sqrt{\alpha\tau}}\tilde{K}^{\mathsf{T}}\mathrm{diag}\big(\sqrt{u^*}\big) \\ \frac{1}{\sqrt{\alpha\tau}}\mathrm{diag}\big(\sqrt{u^*}\big)\tilde{K} & \mathrm{diag}\big(\sqrt{u^*}\big)M\mathrm{diag}\big(\sqrt{u^*}\big) \end{bmatrix} \\
&\qquad + \begin{bmatrix} I_{|S|} & \frac{1}{\sqrt{\alpha\tau}}\tilde{K}^{\mathsf{T}}\mathrm{diag}\big(\sqrt{u^*}\big) \\ -\frac{1}{\sqrt{\alpha\tau}}\mathrm{diag}\big(\sqrt{u^*}\big)\tilde{K} & \mathrm{diag}\big(\sqrt{u^*}\big)M\mathrm{diag}\big(\sqrt{u^*}\big) \end{bmatrix}\Bigg)x \\
&= \frac{1}{x^{\mathrm{H}}x}x^{\mathrm{H}}\begin{bmatrix} I_{|S|} & \\ & \mathrm{diag}\big(\sqrt{u^*}\big)M\mathrm{diag}\big(\sqrt{u^*}\big) \end{bmatrix}x \\
&\geq 0,
\end{aligned}
$$

where the superscript H denotes the Hermitian transpose. Now we proceed to show $\mathrm{Re}(\lambda) \neq 0$. Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_1 \in \mathbb{R}^{|S|}$, $x_2 \in \mathbb{R}^{|S||A|}$. If $\mathrm{Re}(\lambda) = 0$, then

$$0 = x^{\mathrm{H}} \begin{bmatrix} I_{|S|} & \\ & \mathrm{diag}(\sqrt{u^*}) M \mathrm{diag}(\sqrt{u^*}) \end{bmatrix} x = x_1^{\mathrm{H}} x_1$$
$$+ (\mathrm{diag}(\sqrt{u^*})x_2)^{\mathrm{H}} M (\mathrm{diag}(\sqrt{u^*})x_2) \geq 0;$$

thus $x_1 = 0$ and the equality condition of the Cauchy–Schwarz inequality (6.15) must hold. Hence $(x_2)_{sa} = c_s \sqrt{u_{sa}^*}$ for some $c_s \in \mathbb{R}$, $s \in S$. We also know that c_s is not all zero for $s \in S$; otherwise, $x_2 = 0$ and $x = 0$ is not an eigenvector. Thus

$$\tilde{J}x = \begin{bmatrix} I_{|S|} & -\frac{1}{\sqrt{\alpha\tau}}\tilde{K}^{\mathsf{T}}\mathrm{diag}(\sqrt{u^*}) \\ \frac{1}{\sqrt{\alpha\tau}}\mathrm{diag}(\sqrt{u^*})\tilde{K} & \mathrm{diag}(\sqrt{u^*})M\mathrm{diag}(\sqrt{u^*}) \end{bmatrix} x = \frac{-1}{\sqrt{\alpha\tau}} \begin{bmatrix} \tilde{K}^{\mathsf{T}}\mathrm{diag}(\sqrt{u^*})x_2 \\ 0 \end{bmatrix},$$

which is not a scalar multiple of x unless $\tilde{K}^{\mathsf{T}}\mathrm{diag}(\sqrt{u^*})x_2 = 0$. However, as

$$\left(\tilde{K}^{\mathsf{T}}\mathrm{diag}\left(\sqrt{u^*}\right)x_2\right)_{s'} = \sum_{sa} K_{ass'} c_s u_{sa}^* = \sum_s K_{\pi^* ss'} c_s \tilde{u}_s^*, \quad s' \in S,$$

$\tilde{K}^{\mathsf{T}}\mathrm{diag}\left(\sqrt{u^*}\right)x_2 = K_{\pi^*}^{\mathsf{T}}c\tilde{u}^*$, where $c\tilde{u}^*$ denotes the elementwise product. Thus $K_{\pi^*}^{\mathsf{T}}c\tilde{u}^* = 0$ and then $c\tilde{u}^* = 0$, contradicting the fact that c_s is not all zero. The contradiction means that $\mathrm{Re}(\lambda) \neq 0$. Together with the inequality (6.16) we have $\mathrm{Re}(\lambda) > 0$ for any eigenvalue λ of J. Hence $\mathrm{Re}(\lambda) < 0$ for any eigenvalue λ of $-J$, the matrix in the linearized dynamics (6.13). By Lyapunov's indirect theorem [14], (2.7) has locally exponential convergence. □

6.5. Proof of Lemma 3.1. Similar to Lemma 2.3, we first note that $\frac{\partial^2 L}{\partial u_{sa}\partial v_{s'}} = 0$. Moreover,

$$\frac{\partial^2 L_c}{\partial v_s \partial v_{s'}} = \alpha\delta_{ss'}, \quad \frac{\partial^2 L_c}{\partial u_{sa}\partial u_{s'a'}} = \tau\delta_{ss'}\left(\delta_{aa'}\frac{u_{sa}^*}{u_{sa}^2} + \frac{c\tilde{u}_s^*}{(1-c)\tilde{u}_s^2}\right), \quad (s, s', a, a') \in S^2 \times A^2.$$

Hence the Hessian matrix of L_c is

$$\begin{bmatrix} \alpha I_{|S|} & 0 \\ 0 & \tau\mathrm{diag}(u^*/u^2) + \frac{c\tau}{1-c}B \end{bmatrix},$$

where $(u^*/u^2)_{sa} = u_{sa}^*/u_{sa}^2$ and B is a positive definite block-diagonal matrix:

$$(6.17) \qquad B := \begin{bmatrix} \frac{\tilde{u}_1^*}{\tilde{u}_1^2}\mathbf{1}_{|A|}\mathbf{1}_{|A|}^{\mathsf{T}} & & \\ & \ddots & \\ & & \frac{\tilde{u}_{|S|}^*}{\tilde{u}_{|S|}^2}\mathbf{1}_{|A|}\mathbf{1}_{|A|}^{\mathsf{T}} \end{bmatrix}.$$

Thus the Hessian of L_c is positive definite and L_c is strictly convex. The derivatives of L_c are

$$(6.18) \qquad \begin{aligned} \frac{\partial L_c}{\partial v_s} &= \alpha(v_s - v_s^*), \quad s \in S, \\ \frac{\partial L_c}{\partial u_{sa}} &= \tau\left(\frac{u_{sa} - u_{sa}^*}{u_{sa}} + \frac{c}{1-c}\frac{\tilde{u}_s - \tilde{u}_s^*}{\tilde{u}_s}\right), \quad (s, a) \in S \times A, \end{aligned}$$

from which we can see that (v^*, u^*) is a solution to the first-order condition $\frac{\partial L_c}{\partial v} = 0$, $\frac{\partial L_c}{\partial u} = 0$. Since L_c is strictly convex, it is also the unique minimizer of L_c. Now we prove that L_c has bounded sublevel sets. Let $\ell(u) = \sum_s (\tilde{u}_s^* \log \tilde{u}_s^* / \tilde{u}_s + \tilde{u}_s - \tilde{u}_s^*)$. Then $L_c(v, u) = L_0(v, u) + \frac{c\tau}{1-c} \ell(u)$. Since $\frac{\partial^2 \ell}{\partial u^2} = B$ is positive definite, ℓ is strictly convex. Moreover, $\frac{\partial \ell}{\partial u_{sa}} = (\tilde{u}_s - \tilde{u}_s^*) / \tilde{u}_s$ equals 0 when $u = u^*$, so by the strict convexity of ℓ, u^* is the unique minimizer of ℓ, and thus $\ell(u) \geq \ell(u^*) = 0$. Hence the sublevel set $\{(v, u) \mid L_c(v, u) \leq C\} \subset \{(v, u) \mid L_0(v, u) \leq C\}$. Since the latter is bounded according to Lemma 2.3, the sublevel set of L_c is also bounded. □

6.6. Proof of Lemma 3.2. Plugging the first-order condition (6.8) for the exact solution (v^*, u^*) into the interpolating natural gradient (3.3) results in

(6.19)
$$
\frac{dv_{s'}}{dt} = -\left((v_{s'} - v_{s'}^*) - \frac{1}{\alpha} \sum_{sa} K_{ass'} (u_{sa} - u_{sa}^*) \right), \quad s' \in S,
$$
$$
\frac{du_{s\cdot}}{dt} = -\tilde{u}_s \left(\text{diag}(\pi_{s\cdot}) - c\pi_{s\cdot}\pi_{s\cdot}^\mathsf{T} \right) \left(\left(\log \frac{u_{s\cdot}}{\tilde{u}_s} - \log \frac{u_{s\cdot}^*}{\tilde{u}_s^*} \right) + \frac{1}{\tau} \sum_{s'} K_{\cdot ss'} (v_{s'} - v_{s'}^*) \right), \quad s \in S,
$$

where π_{sa} is defined as $u_{sa}/\tilde{u}_s$. A direct calculation shows that

(6.20)
$$
(u_{s\cdot} - u_{s\cdot}^*)/u_{s\cdot} + \frac{c}{1-c} (\tilde{u}_s - \tilde{u}_s^*)/\tilde{u}_s \mathbf{1}_{|A|} = \left(\text{diag}(1/\pi_{s\cdot}) + \frac{c}{1-c} \mathbf{1}_{|A|} \mathbf{1}_{|A|}^\mathsf{T} \right) \left(\frac{u_{s\cdot} - u_{s\cdot}^*}{\tilde{u}_s} \right).
$$

Then

(6.21)
$$
\frac{dL}{dt} = -\alpha \sum_{s'} (v_{s'} - v_{s'}^*) \left((v_{s'} - v_{s'}^*) - \frac{1}{\alpha} \sum_{sa} K_{ass'} (u_{sa} - u_{sa}^*) \right)
$$
$$
-\tau \sum_s \left[\left(\frac{u_{s\cdot} - u_{s\cdot}^*}{\tilde{u}_s} \right)^\mathsf{T} \left(\text{diag}(1/\pi_{s\cdot}) + \frac{c}{1-c} \mathbf{1}_{|A|} \mathbf{1}_{|A|}^\mathsf{T} \right) \tilde{u}_s \left(\text{diag}(\pi_{s\cdot}) - c\pi_{s\cdot}\pi_{s\cdot}^\mathsf{T} \right) \right.
$$
$$
\left. \left(\left(\log \frac{u_{s\cdot}}{\tilde{u}_s} - \log \frac{u_{s\cdot}^*}{\tilde{u}_s^*} \right) + \frac{1}{\tau} \sum_{s'} K_{\cdot ss'} (v_{s'} - v_{s'}^*) \right) \right]
$$
$$
= -\alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*) \left(\log \frac{u_{as}}{\tilde{u}_s} - \log \frac{u_{as}^*}{\tilde{u}_s^*} \right),
$$

where we have used the fact that

$$
\left(\text{diag}(1/\pi_{s\cdot}) + \frac{c}{1-c} \mathbf{1}_{|A|} \mathbf{1}_{|A|}^\mathsf{T} \right) \left(\text{diag}(\pi_{s\cdot}) - c\pi_{s\cdot}\pi_{s\cdot}^\mathsf{T} \right)
$$
$$
= \text{diag}(1/\pi_{s\cdot})\text{diag}(\pi_{s\cdot}) + \frac{c}{1-c} \mathbf{1}_{|A|} \mathbf{1}_{|A|}^\mathsf{T} \text{diag}(\pi_{s\cdot}) - \frac{c^2}{1-c} \mathbf{1}_{|A|} \mathbf{1}_{|A|}^\mathsf{T} \pi_{s\cdot}\pi_{s\cdot}^\mathsf{T}
$$
$$
- c\,\text{diag}(1/\pi_{s\cdot})\pi_{s\cdot}\pi_{s\cdot}^\mathsf{T} = I + \left(\frac{c}{1-c} - \frac{c^2}{1-c} - c \right) \mathbf{1}_{|A|} \pi_{s\cdot}^\mathsf{T} = I.
$$

Therefore,

$$
\frac{dL_c}{dt} = -\alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*) \left(\log \frac{u_{as}}{\tilde{u}_s} - \log \frac{u_{as}^*}{\tilde{u}_s^*} \right),
$$

where the right-hand side coincides with that of (6.10). Hence $\frac{\mathrm{d}L_c}{\mathrm{d}t} = \frac{\mathrm{d}L_0}{\mathrm{d}t} \leq 0$ by the proof of Lemma 2.4. Let

$$\mathcal{R} = \left\{ (v, u) \mid -\alpha \sum_{s'} (v_{s'} - v_{s'}^*)^2 - \tau \sum_{sa} (u_{sa} - u_{sa}^*)(\log u_{sa}/\tilde{u}_s - \log u_{sa}^*/\tilde{u}_s^*) = 0 \right\}.$$

Then by the proof of Lemma 2.4, $\mathcal{R} = \{(v, u) \mid v = v^*, u_{sa} = c_s u_{sa}^*, c_s \in \mathbb{R}_+, s \in S\}$. We proceed to prove that the only trajectory of (6.19) in $\mathcal{R}$ is $(v, u) = (v^*, u^*)$. Since $v = v^*$ for any $(v, u) \in \mathcal{R}$, $\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = 0$ for $s' \in S$. In addition, for any $s' \in S$ we have

$$0 = \sum_{sa} K_{ass'}(u_{sa} - u_{sa}^*) = \sum_{s} K_{\pi^* ss'}(c_s - 1)\tilde{u}_s^*,$$

by the same calculation as (6.11). This means that for point (v, u) on the trajectory of (6.19) in $\mathcal{R}$, $K_{\pi^*}^\mathsf{T}((c - 1)\tilde{u}^*) = 0$; thus $(c - 1)\tilde{u}^* = 0$ and $c_s = 1$ for any $s \in S$. Since this is true for any (v, u) on the trajectory, the trajectory is a single point $(v, u) = (v^*, u^*)$. □

6.7. Proof of Theorem 3.4. The linearized dynamic of the interpolating natural gradient (3.3) is

(6.22)

$$\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = -\left((v_{s'} - v_{s'}^*) - \frac{1}{\alpha} \sum_{sa} K_{ass'y}(u_{sa} - u_{sa}^*) \right), \ s' \in S,$$

$$\frac{\mathrm{d}u_{s\cdot}}{\mathrm{d}t} = -\left(\mathrm{diag}(u_{s\cdot}^*) - \frac{c}{\tilde{u}_s^*} u_{s\cdot}^*(u_{s\cdot}^*)^\mathsf{T} \right) \left(\frac{1}{\tau} \sum_{s'} K_{\cdot ss'}(v_{s'} - v_{s'}^*) + \frac{u_{s\cdot} - u_{s\cdot}^*}{u_{s\cdot}^*} - \frac{\tilde{u}_s - \tilde{u}_s^*}{\tilde{u}_s^*} \mathbf{1} \right), \ s \in S.$$

Define $\tilde{K}$ by $\tilde{K}_{(s-1)|A|+a,s'} = K_{ass'}$ and let $\delta v = v - v^*$, $\delta u = u - u^*$. Then (6.22) becomes

(6.23)

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \delta v \\ \delta u \end{bmatrix} = -\begin{bmatrix} I_{|S|} & -\frac{1}{\alpha}\tilde{K}^\mathsf{T} \\ \frac{1}{\tau}G\tilde{K} & GM \end{bmatrix} \begin{bmatrix} \delta v \\ \delta u \end{bmatrix},$$

where M is a block-diagonal matrix defined as in (6.14) and G is a block-diagonal matrix

(6.24)

$$\begin{bmatrix} G_1 & & \\ & \ddots & \\ & & G_{|S|} \end{bmatrix},$$

with $G_s = \mathrm{diag}(u_{s\cdot}^*) - \frac{c}{\tilde{u}_s^*}u_{s\cdot}^*(u_{s\cdot}^*)^\mathsf{T}$. Notice that G_s is symmetric. By the Cauchy–Schwarz inequality

$$x^\mathsf{T} G_s x = \sum_a u_{sa}^* x_a^2 - \frac{c}{\tilde{u}_s^*} \left(\sum_a u_{sa}^* x_a \right)^2 \geq \frac{1}{\tilde{u}_s^*} \left(\sum_a u_{sa}^* x_a \right)^2 - \frac{c}{\tilde{u}_s^*} \left(\sum_a u_{sa}^* x_a \right)^2$$

$$= \frac{1 - c}{\tilde{u}_s^*} \left(\sum_a u_{sa}^* x_a \right)^2 > 0 \quad \forall x \in \mathbb{R}^{|A|}, x \neq 0, \quad \forall s \in S.$$

Thus G is positive definite, and we can define the positive definite square root F of G, i.e., $F^2 = G$. Define an invertible matrix Q as

$$Q = \begin{bmatrix} \sqrt{\tau} I_{|S|} & \\ & \sqrt{\alpha} F \end{bmatrix},$$

and denote the matrix in the linearized dynamics (6.23) as $-J$, i.e.,

$$J = \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha} \tilde{K}^{\mathsf{T}} \\ \frac{1}{\tau} G \tilde{K} & GM \end{bmatrix}.$$

Then

$$Q^{-1} J Q = \begin{bmatrix} \frac{1}{\sqrt{\tau}} I_{|S|} & \\ & \frac{1}{\sqrt{\alpha}} F^{-1} \end{bmatrix} \begin{bmatrix} I_{|S|} & -\frac{1}{\alpha} \tilde{K}^{\mathsf{T}} \\ \frac{1}{\tau} G \tilde{K} & GM \end{bmatrix} \begin{bmatrix} \sqrt{\tau} I_{|S|} & \\ & \sqrt{\alpha} F \end{bmatrix}$$

$$= \begin{bmatrix} I_{|S|} & -\frac{1}{\sqrt{\alpha\tau}} \tilde{K}^{\mathsf{T}} F \\ \frac{1}{\sqrt{\alpha\tau}} F \tilde{K} & FMF \end{bmatrix}.$$

It suffices to show that the real part of the eigenvalues of $Q^{-1}JQ$ is positive. Denote $Q^{-1}JQ$ by $\tilde{J}$. Using the positive semidefiniteness of FMF, for any eigenpair (λ, x) of $\tilde{J}$ we can deduce

$$\operatorname{Re}(\lambda) = \frac{1}{2} \left(\frac{x^{\mathsf{H}} \tilde{J} x}{x^{\mathsf{H}} x} + \frac{x^{\mathsf{H}} \tilde{J}^{\mathsf{H}} x}{x^{\mathsf{H}} x} \right)$$

(6.25)
$$= \frac{1}{2 x^{\mathsf{H}} x} x^{\mathsf{H}} \left(\begin{bmatrix} I_{|S|} & -\frac{1}{\sqrt{\alpha\tau}} \tilde{K}^{\mathsf{T}} F \\ \frac{1}{\sqrt{\alpha\tau}} F \tilde{K} & FMF \end{bmatrix} + \begin{bmatrix} I_{|S|} & \frac{1}{\sqrt{\alpha\tau}} \tilde{K}^{\mathsf{T}} F \\ -\frac{1}{\sqrt{\alpha\tau}} F \tilde{K} & FMF \end{bmatrix} \right) x$$

$$= \frac{1}{x^{\mathsf{H}} x} x^{\mathsf{H}} \begin{bmatrix} I_{|S|} & \\ & FMF \end{bmatrix} x \geq 0.$$

It remains to show $\operatorname{Re}(\lambda) \neq 0$. Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_1 \in \mathbb{R}^{|S|}$, $x_2 \in \mathbb{R}^{|S||A|}$. Then if $\operatorname{Re}(\lambda) = 0$,

$$0 = x^{\mathsf{H}} \begin{bmatrix} I_{|S|} & \\ & FMF \end{bmatrix} x = x_1^{\mathsf{H}} x_1 + (Fx_2)^{\mathsf{H}} M (Fx_2) \geq 0.$$

Thus $x_1 = 0$, and the equality condition of the Cauchy–Schwarz inequality (6.15) must hold. Hence $(Fx_2)_{sa} = c_s u_{sa}^*$ for some $c_s \in \mathbb{R}$, $s \in S$. We also know that c_s is not all zero for $s \in S$; otherwise, $x_2 = 0$, so $x = 0$ is not an eigenvector. Thus

$$\tilde{J} x = \begin{bmatrix} I_{|S|} & -\frac{1}{\sqrt{\alpha\tau}} \tilde{K}^{\mathsf{T}} F \\ \frac{1}{\sqrt{\alpha\tau}} F \tilde{K} & FMF \end{bmatrix} x = \frac{-1}{\sqrt{\alpha\tau}} \begin{bmatrix} \tilde{K}^{\mathsf{T}} F x_2 \\ 0 \end{bmatrix},$$

which is not a scalar multiple of x unless $\tilde{K}^{\mathsf{T}} F x_2 = 0$. Since

$$\left(\tilde{K}^{\mathsf{T}} F x_2 \right)_{s'} = \sum_{sa} K_{ass'} c_s u_{sa}^* = \sum_s K_{\pi^* ss'} c_s \tilde{u}_s^*, \quad s' \in S,$$

$\tilde{K}^{\mathsf{T}} F x_2 = K_{\pi^*}^{\mathsf{T}} c \tilde{u}^*$. Thus $c \tilde{u}^* = 0$, contradicting the fact that c_s is not all zero. This contradiction means that $\operatorname{Re}(\lambda) \neq 0$. Together with the inequality (6.16), $\operatorname{Re}(\lambda) > 0$ for any eigenvalue λ of J. Hence $\operatorname{Re}(\lambda) < 0$ for any eigenvalue λ of $-J$, the matrix

in the linearized dynamics (6.23). Finally, by Lyapunov's indirect theorem [14], (3.3) has locally exponential convergence. □

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 Thirty-Third Conference on Learning Theory, PMLR 125, 2020, pp. 64–66.

[2] Z. AHMED, N. LE ROUX, M. NOROUZI, AND D. SCHUURMANS, *Understanding the impact of entropy on policy optimization*, in 36th International Conference on Machine Learning, PMLR 97, 2019, pp. 151–160.

[3] K. ASADI AND M. L. LITTMAN, *An alternative softmax operator for reinforcement learning*, in 34th International Conference on Machine Learning, PMLR 70, 2017, pp. 243–252.

[4] R. E. BELLMAN AND S. E. DREYFUS, *Applied Dynamic Programming*, Princeton University Press, 2015.

[5] F. BERKENKAMP, M. TURCHETTA, A. P. SCHOELLIG, AND A. KRAUSE, *Safe Model-based Reinforcement Learning with Stability Guarantees*, preprint, https://arxiv.org/abs/1705.08551, 2017.

[6] G. BROCKMAN, V. CHEUNG, L. PETTERSSON, J. SCHNEIDER, J. SCHULMAN, J. TANG, AND W. ZAREMBA, *OpenAI Gym*, preprint, arXiv:1606.01540, 2016.

[7] S. CEN, C. CHENG, Y. CHEN, Y. WEI, AND Y. CHI, *Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization*, preprint, https://arxiv.org/abs/2007.06558, 2020.

[8] W. S. CHO AND M. WANG, *Deep Primal-Dual Reinforcement Learning: Accelerating Actor-Critic Using Bellman Duality*, preprint, https://arxiv.org/abs/1712.02467, 2017.

[9] Y. CHOW, O. NACHUM, E. DUENEZ-GUZMAN, AND M. GHAVAMZADEH, *A Lyapunov-based Approach to Safe Reinforcement Learning*, preprint, https://arxiv.org/abs/1805.07708, 2018.

[10] B. DAI, A. SHAW, L. LI, L. XIAO, N. HE, Z. LIU, J. CHEN, AND L. SONG, *SBEED: Convergent reinforcement learning with nonlinear function approximation*, in 35th International Conference on Machine Learning, PMLR 80, 2018, pp. 1125–1134.

[11] D. DING, K. ZHANG, T. BASAR, AND M. JOVANOVIC, *Natural policy gradient primal-dual method for constrained Markov decision processes*, in Advances in Neural Information Processing Systems 33, Curran Associates, 2020, pp. 8378–8390.

[12] M. GEIST, B. SCHERRER, AND O. PIETQUIN, *A theory of regularized Markov decision processes*, in 36th International Conference on Machine Learning, PMLR 97, 2019, pp. 2160–2169.

[13] H. GONG, *Primal-Dual Method for Reinforcement Learning and Markov Decision Processes*, Ph.D. thesis, Princeton University, 2021.

[14] W. M. HADDAD AND V. CHELLABOINA, *Nonlinear Dynamical Systems and Control*, Princeton University Press, 2011.

[15] S. M. KAKADE, *A natural policy gradient*, in Advances in Neural Information Processing Systems 14, MIT Press, 2001, pp. 1531–1538.

[16] R. KALMAN AND J. BERTRAM, *Control system analysis and design via the second method of Lyapunov: (I) continuous-time systems (II) discrete time systems*, IRE Trans. Automat. Control, 4 (1959), pp. 112–112.

[17] R. E. KALMAN AND J. E. BERTRAM, *Control system analysis and design via the "second method" of Lyapunov: I. Continuous-time systems*, Trans. ASME Ser. D. J. Basic Engrg., 82 (1960), pp. 371–393.

[18] S. KHODADADIAN, P. R. JHUNJHUNWALA, S. M. VARMA, AND S. T. MAGULURI, *On the linear convergence of natural policy gradient algorithm*, in 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 3794–3799.

[19] G. LAN, *Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes*, Math. Program., 198 (2023), pp. 1059–1106.

[20] D. LEE AND N. HE, *Stochastic Primal-Dual Q-Learning*, preprint, https://arxiv.org/abs/1810.08298, 2018.

[21] G. LI, Y. WEI, Y. CHI, Y. GU, AND Y. CHEN, *Softmax Policy Gradient Methods Can Take Exponential Time to Converge*, preprint, https://arxiv.org/abs/2102.11270v2, 2021.

[22] H. LI, S. GUPTA, H. YU, L. YING, AND I. DHILLON, Approximate Newton policy gradient algorithms, SIAM J. Sci. Comput., 45 (2023), pp. A2585–A2609.

[23] A. M. LYAPUNOV, *The general problem of the stability of motion*, Internat. J. Control, 55 (1992), pp. 531–534.

[24] J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans, *On the global convergence rates of softmax policy gradient methods*, in 37th International Conference on Machine Learning, PMLR 119, 2020, pp. 6820–6829.

[25] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, Springer Science & Business Media, 2012.

[26] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, *Trust-PCL: An Off-Policy Trust Region Method for Continuous Control*, preprint, https://arxiv.org/abs/1707.01891, 2017.

[27] G. Neu, A. Jonsson, and V. Gómez, *A Unified View of Entropy-Regularized Markov Decision Processes*, preprint, https://arxiv.org/abs/1705.07798, 2017.

[28] T. J. Perkins and A. G. Barto, *Lyapunov design for safe reinforcement learning*, J. Mach. Learn. Res., 3 (2002), pp. 803–832.

[29] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, 2014.

[30] K. Rawlik, M. Toussaint, and S. Vijayakumar, *On stochastic optimal control and reinforcement learning by approximate inference (extended abstract)*, in Twenty-Third International Joint Conference on Artificial Intelligence, AAAI Press, 2013, pp. 3052–3056.

[31] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, *Trust region policy optimization*, in 32nd International Conference on Machine Learning, PMLR 37, 2015, pp. 1889–1897.

[32] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, *High-Dimensional Continuous Control Using Generalized Advantage Estimation*, preprint, https://arxiv.org/abs/1506.02438, 2015.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal Policy Optimization Algorithms*, preprint, https://arxiv.org/abs/1707.06347, 2017.

[34] J. B. Serrano and G. Neu, *Faster saddle-point optimization for solving large-scale Markov decision processes*, in Conference on Learning for Dynamics and Control, PMLR 120, 2020, pp. 413–423.

[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

[36] R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour, *Policy gradient methods for reinforcement learning with function approximation*, in Advances in Neural Information Processing Systems 12, MIT Press, 2000, pp. 1057–1063.

[37] M. Wang, *Primal-Dual π Learning: Sample Complexity and Sublinear Run Time for Ergodic Markov Decision Problems*, preprint, https://arxiv.org/abs/1710.06100, 2017.

[38] M. Wang, *Randomized linear programming solves the Markov decision problem in nearly linear (sometimes sublinear) time*, Math. Oper. Res., 45 (2020), pp. 517–546.

[39] M. Wang and Y. Chen, *An online primal-dual method for discounted Markov decision processes*, in IEEE 55th Conference on Decision and Control, IEEE, 2016, pp. 4516–4521.

[40] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, Mach. Learn., 8 (1992), pp. 229–256.

[41] L. Ying and Y. Zhu, *A note on optimization formulations of Markov decision processes*, Commun. Math. Sci., 20 (2022), pp. 727–745.

[42] 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*, preprint, https://arxiv.org/abs/2105.11066, 2021.

[43] J. Zhang, A. S. Bedi, M. Wang, and A. Koppel, *Cautious reinforcement learning via distributional risk in the dual domain*, IEEE J. Sel. Areas Inform. Theory, 2 (2021), pp. 611–626.