S}t}A}M} J}. O}P}T}I}M}.
V}o}l}. 33, N}o}. 2, p}p}. 1061--1091



# POLICY MIRROR DESCENT FOR REGULARIZED REINFORCEMENT LEARNING: A GENERALIZED FRAMEWORK WITH LINEAR CONVERGENCE*

WENHAO ZHAN[r], SHICONG CEN[r], BAIHE HUANG[S], YUXIN CHEN[P], JASON D. LEE[r], AND YUEJIE CHI[r]

Abstract. Policy optimization, which learns the policy of interest by maximizing the value function via large-scale optimization techniques, lies at the heart of modern reinforcement learning (RL). In addition to value maximization, other practical considerations arise commonly as well, including the need of encouraging exploration, and that of ensuring certain structural properties of the learned policy due to safety, resource, and operational constraints. These considerations can often be accounted for by resorting to regularized RL, which augments the target value function with a structure-promoting regularization term. Focusing on an infinite-horizon discounted tabular Markov decision process, this paper proposes a generalized policy mirror descent (GPMD) algorithm for solving regularized RL. As a generalization of policy mirror descent [G. Lan, Math. Program., 198 (2023), pp. 1059--1106], the proposed algorithm accommodates a general class of convex regularizers as well as a broad family of Bregman divergence in cognizance of the regularizer in use. We demon-strate that our algorithm converges linearly to the global solution over an entire range of learning rates, in a dimension-free fashion, even when the regularizer lacks strong convexity and smoothness. In addition, this linear convergence feature is provably stable in the face of inexact policy evaluation and imperfect policy updates. Numerical experiments are provided to corroborate the applicability and appealing performance of GPMD.

Key words. policy mirror descent, Bregman divergence, regularization, policy optimization

MSC codes. 68Q25, 68Q32, 90C26

DOI. 10.1137/21M1456789

## 1. Introduction.

Policy optimization lies at the heart of recent successes of reinforcement learning (RL) [39]. In its basic form, the optimal policy of interest, or a suitably parameterized version, is learned by attempting to maximize the value

*Received by the editors November 2, 2021; accepted for publication (in revised form) January 6, 2023; published electronically June 22, 2023. A preliminary version of this work was presented at NeurIPS 2021 Workshop on Optimization for Machine Learning. The first two authors contributed equally.

https://doi.org/10.1137/21M1456789

Funding: The research of the second and sixth authors was supported in part by grants ONR N00014-19-1-2404, NSF CCF-2106778, DMS-2134080, CCF-1901199, CCF-2007911, and CNS-2148212. The research of the second author was also supported by Wei Shen and Xuehong Zhang`s Presidential Fellowship and Nicholas Minnici Dean's Graduate Fellowship in Electrical and Computer Engineering at Carnegie Mellon University. The research of the first and fourth authors was supported in part by the Google Research Scholar Award, the Alfred P. Sloan Research Fellowship, and grants AFOSR FA9550-22-1-0198, ONR N00014-22-1-2354, NSF CCF-2221009, CCF-1907661, IIS-2218713, and IIS-2218773. The research of the first and fifth authors was supported in part by the ARO under MURI award W911NF-11-1-0304, the Sloan Research Fellowship, NSF CCF 2002272, NSF IIS 2107304, and an ONR Young Investigator Award.

[r] Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (wz3993@princeton.edu, jasondl@princeton.edu).

[r] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (shicongc@andrew.cmu.edu, yuejiechi@cmu.edu).

[S] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720 USA (baihehuang@pku.edu.cn).

[P] Department of Statistics and Data Science, Wharton School, University of Pennsylvania, Philadelphia, PA 19104 USA (yuxinc@wharton.upenn.edu).

function in a Markov decision process (MDP). For the most part, the maximization step is carried out by means of first-order optimization algorithms amenable to large-scale applications, whose foundations were set forth in the early works of [53, 47]. A partial list of widely adopted variants in modern practice includes policy gradient (PG) methods [47], natural policy gradient (NPG) methods [24], TRPO [44], PPO [45], and soft actor-critic methods [22], to name just a few. In comparison with model-based and value-based approaches, this family of policy-based algorithms offers a remarkably flexible framework that accommodates both continuous and discrete action spaces and lends itself well to the incorporation of powerful function approximation schemes like neural networks. In stark contrast to its practical success, however, theoretical understanding of policy optimization remains severely limited even for the tabular case, largely owing to the ubiquitous nonconvexity issue underlying the objective function.

1.1. The role of regularization. In practice, there are often competing objectives and additional constraints that the agent has to deal with in conjunction with maximizing values, which motivate the studies of regularization techniques in RL. In what follows, we isolate a few representative examples.

- Promoting exploration. In the face of large problem dimensions and complex dynamics, it is often desirable to maintain a suitable degree of randomness in the policy iterates, in order to encourage exploration and discourage premature convergence to suboptimal policies. A popular strategy of this kind is to enforce entropy regularization [54], which penalizes policies that are not suficiently stochastic. Along similar lines, the Tsallis entropy regularization [16, 31] further promotes sparsity of the learned policy while encouraging exploration, ensuring that the resulting policy does not assign non-negligible probabilities to too many suboptimal actions.
- Safe RL. In a variety of application scenarios such as industrial robot arms and self-driving vehicles, the agents are required to operate safely both to themselves and the surroundings [5, 40]; for example, certain actions might be strictly forbidden in some states. One way to incorporate such prescribed operational constraints is through adding a regularizer (e.g., a properly chosen log barrier or indicator function tailored to the constraints) to explicitly account for the constraints.
- Cost-sensitive RL. In reality, different actions of an agent might incur drastically different costs even for the same state. This motivates the design of new objective functions that properly trade off the cumulative rewards against the accumulated cost, which often take the form of certain regularized value functions.

Viewed in this light, it is of imminent value to develop a unified framework towards understanding the capability and limitations of regularized policy optimization. While a recent line of works [3, 38, 12] has looked into specific types of regularization techniques such as entropy regularization, existing convergence theory remains highly inadequate when it comes to a more general family of regularizers.

1.2. Main contributions. The current paper focuses on policy optimization for regularized RL in a $\gamma$-discounted infinite horizon Markov decision process (MDP) with state space $\mathcal{S}$, action space $\mathcal{A}$, and reward function $r(s,a)$. The goal is to find an optimal policy that maximizes a regularized value function. Informally speaking, the regularized value function associated with a given policy $\pi$ takes the following form:

$$V_\tau^\pi = V^\pi - \tau \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t h_s(\pi(a_t|s_t))\right],$$

where $V^i$ denotes the original (unregularized) value function, $u > 0$ is the regularization parameter, $h_s(t)$ denotes a convex regularizer employed to regularize the policy in state $s$, and the expectation is taken over certain marginal state distribution w.r.t. the MDP (to be made precise in section 2.1). It is noteworthy that this paper does not require the regularizer $h_s$ to be either strongly convex or smooth.

In order to maximize the regularized value function (2.8b), Lan [27] exhibited a seminal algorithm called Policy Mirror Descent (PMD), which can be viewed as an adaptation of the mirror descent algorithm [41, 7] to the realm of policy optimization. In particular, PMD subsumes the natural policy gradient (NPG) method [24] as a special case. To further generalize PMD [27], we propose an algorithm called Generalized Policy Mirror Descent (GPMD). In each iteration, the policy is updated for each state in parallel via a mirror-descent style update rule. In sharp contrast to [27], which considered a generic Bregman divergence, our algorithm selects the Bregman divergence adaptively in cognizance of the regularizer, which leads to complementary perspectives and insights. Several important features and theoretical appeal of GPMD are summarized as follows.

- GPMD substantially broadens the range of (provably effective) algorithmic choices for regularized RL and subsumes several well-known algorithms as special cases. For example, it reduces to regularized policy iteration [21] when the learning rate tends to infinity and subsumes entropy-regularized NPG methods as special cases if we take the Bregman divergence to be the Kullback--Leibler (KL) divergence [12].
- Assuming exact policy evaluation and perfect policy update in each iteration, GPMD converges linearly--in a dimension-free fashion--over the entire range of the learning rate $a > 0$. More precisely, it converges to an $n$-optimal regularized Q-function in no more than an order of

$$\frac{1+au}{au(1-a)}\log\frac{1}{n}$$

  iterations (up to some logarithmic factor). Encouragingly, this appealing feature is valid for a broad family of convex and possibly nonsmooth regularizers.
- The intriguing convergence guarantees are robust in the face of inexact policy evaluation and imperfect policy updates; namely, the algorithm is guaranteed to converge linearly at the same rate until an error floor is hit. See section 3.2 for details.
- Numerical experiments are provided in section 5 to demonstrate the practical applicability and appealing performance of the proposed GPMD algorithm.

Finally, we find it helpful to briefly compare the above findings with prior works. As soon as the learning rate exceeds $a \geq 1/u$, the iteration complexity of our algo-rithm is at most on the order of $\frac{1}{1-a}\log\frac{1}{\epsilon}$, thus matching that of regularized policy iteration [21]. In comparison to [27], our work sets forth a different framework to an-alyze mirror-descent-type algorithms for regularized policy optimization, generalizing and refining the approach in [12] far beyond entropy regularization. When constant learning rates are employed, the linear convergence of PMD [27] critically requires the regularizer to be strongly convex, with only sublinear convergence theory established for convex regularizers. In contrast, we establish the linear convergence of GPMD under constant learning rates even in the absence of strong convexity. Furthermore, for the special case of entropy regularization, the stability analysis of GPMD also significantly improves over the prior art in [12], preventing the error floor from blowing up when the learning rate approaches zero, as well as incorporating the impact of

optimization error that was previously uncaptured. More detailed comparisons with [27, 12] can be found in section 3.

1.3. Related works. Before embarking on our algorithmic and theoretic developments, we briefly review a small sample of other related works.

Global convergence of policy gradient methods. Recent years have witnessed a surge of activities towards understanding the global convergence properties of policy gradient methods and their variants for both continuous and discrete RL problems, examples including [20, 9, 3, 63, 51, 37, 10, 25, 35, 37, 4, 56, 51, 12, 36, 34, 52, 60, 62, 61, 46], among other things. The authors of [42] provided the first interpretation of NPG methods as mirror descent [41], thereby enabling the adaptation of techniques for analyzing mirror descent to the studies of NPG-type algorithms such as TRPO [46, 48]. It has been shown that the NPG method converges sublinearly for unregularized MDPs with a fixed learning rate [3], and converges linearly if the learning rate is set adaptively [25], via exact line search [10], or following a geometrically increasing schedule [55]. The global linear convergence of NPG holds more generally for an arbitrary fixed learning rate when entropy regularization is enforced [12]. Noteworthily, the authors of [33] established a lower bound indicating that softmax PG methods can take an exponential time--in the size of the state space--to converge, while the convergence rates of NPG-type methods are almost independent of the problem dimension. In addition, another line of recent works [1, 23, 30] established regret bounds for approximate NPG methods--termed as KL-regularized approximate policy iteration therein--for infinite-horizen undiscounted MDPs, which are beyond the scope of the current paper.

Regularization in RL. Regularization has been suggested to the RL literature either through the lens of optimization [17, 3] or through the lens of dynamic programming [21, 50]. Our work is clearly an instance of the former type. Several recent results in the literature merit particular attention: The authors of [3] demonstrated sublinear convergence guarantees for PG methods in the presence of relative entropy regularization and the authors of [38] established linear convergence of entropy-regularized PG methods, whereas the authors of [12] derived an almost dimension-free linear convergence theory for NPG methods with entropy regularization. Most of the existing literature focused on the entropy regularization or KL-type regularization, and the studies of general regularizers had been quite limited until the recent work [27]. The regularized MDP problems are also closely related to the studies of constrained MDPs, as both types of problems can be employed to model/promote constraint satisfaction in RL, as recently investigated in, e.g., [15, 19, 18, 58, 57]. Note, however, that it is dificult to directly compare our algorithm with these methods, due to drastically different formulations and settings.

1.4. Notation. Let us introduce several notations that will be adopted throughout. For any set $A$, we denote by $|A|$ the cardinality of a set $A$ and let $\Delta(A)$ indicate the probability simplex over the set $A$. For any convex and differentiable function $h(t)$, the Bregman divergence generated by $h(t)$ is defined as

$$D_h(z,x) := h(z) - h(x) - \langle \nabla h(x), z - x \rangle. \tag{1.1}$$

For any convex (but not necessarily differentiable) function $h(t)$, we denote by $\partial h$ the subdifferential of $h$. Given two probability distributions $\pi_1$ and $\pi_2$ over $A$, the KL divergence from $\pi_2$ to $\pi_1$ is defined as $KL(\pi_1 | \pi_2) := \sum_{a \in A} \pi_1(a) \log \frac{\pi_1(a)}{\pi_2(a)}$. For any vectors $a = [a_i]_{1 \le i \le n}$ and $b = [b_i]_{1 \le i \le n}$, the notation $a \le b$ (resp., $a < b$) means that

$a_i \le b_i$ ($a_i \ge b_i$) for every $1 \le i \le n$. We shall also use 1 (resp., 0) to denote the all-one (resp., all-zero) vector whenever it is clear from the context.

## 2. Model and algorithms.

**2.1. Problem settings.** The focus of this paper is a discounted infinite-horizon Markov decision process, as represented by $M = (S, A, P, r, \alpha)$ [8]. Here, $S := \{1, \dots, |S|\}$ is the state space, $A := \{1, \dots, |A|\}$ is the action space, $\alpha \in [0, 1)$ is the discount factor, and $P : S \times A \to \Delta(S)$ is the probability transition matrix (so that $P(t|s, a)$ is the transition probability from state $s$ upon execution of action $a$), whereas $r : S \times A \to [0, 1]$ is the reward function (so that $r(s, a)$ indicates the immediate reward received in state $s$ after action $a$ is executed). Here, we focus on finite-state and finite-action scenarios, meaning that both $|S|$ and $|A|$ are assumed to be finite. A policy $\pi : S \to \Delta(A)$ specifies a possibly randomized action selection rule; namely, $\pi(t|s)$ represents the action selection probability in state $s$.

For any policy $\pi$, we define the associated value function $V^\pi : S \to \mathbb{R}$ as follows:

$$\forall s \in S: \quad V^\pi(s) := \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(t|s_t),\\ s_{t+1} \sim P(t|s_t, a_t)\\ t \ge 0}} \left[ \sum_{t=0}^{\infty} \alpha^t r(s_t, a_t) \,\middle|\, s_0 = s \right], \tag{2.1}$$

which can be viewed as the utility function we wish to maximize. Here, the expectation is taken over the randomness of the MDP trajectory $\{(s_t, a_t)\}_{t \ge 0}$ induced by policy $\pi$. Similarly, when the initial action $a$ is fixed, we can define the action-value function (or Q-function) as follows:

$$\forall (s, a) \in S \times A: \quad Q^\pi(s, a) := \mathop{\mathbb{E}}_{\substack{s_{t+1} \sim P(t|\\ s_t, a_t), a_{t+1} \sim \pi(t|\\ s_{t+1}) \,|\, t \ge 0}} \left[ \sum_{t=0}^{\infty} \alpha^t r(s_t, a_t) \,\middle|\, s_0 = s, a_0 = a \right]. \tag{2.2}$$

A well-known fact is that the policy gradient of $V^\pi$ (w.r.t. the policy $\pi$) admits the following closed-form expression [47]:

$$\forall (s, a) \in S \times A: \quad \frac{\partial V^\pi(s_0)}{\partial \pi(a|s)} = \frac{1}{1-\alpha} d^\pi_{s_0}(s) Q^\pi(s, a). \tag{2.3}$$

Here, $d^\pi_{s_0} \in \Delta(S)$ is the so-called discounted state visitation distribution defined as follows:

$$d^\pi_{s_0}(s) := (1-\alpha) \sum_{t=0}^{\infty} \alpha^t P^\pi(s_t = s | s_0), \tag{2.4}$$

where $P^\pi(s_t = s | s_0)$ denotes the probability of $s_t = s$ when the MDP trajectory $\{s_t\}_{t \ge 0}$ is generated under policy $\pi$ given the initial state $s_0$.

Furthermore, the optimal value function and the optimal Q-function are defined and denoted by

$$\forall (s, a) \in S \times A: \quad V^*(s) := \max_\pi V^\pi(s), \qquad Q^*(s, a) := \max_\pi Q^\pi(s, a). \tag{2.5}$$

It is well known that there exists at least one optimal policy, denoted by $\pi^*$, that simultaneously maximizes the value function and the Q-function for all state-action pairs [2].

Regularized MDP. In practice, the agent is often asked to design policies that possess certain structural properties in order to be cognizant of system constraints such as safety and operational constraints, as well as to encourage exploration during the optimization/learning stage. A natural strategy to achieve these is to resort to the following regularized value function w.r.t. a given policy $\pi$ [42, 38, 12, 27]:

$$\text{for any } s \in S: \quad V_\tau^\pi(s) := \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t),\\ s_{t+1} \sim P(\cdot|s_t,a_t)\\ t \geq 0}} \left[ \sum_{t=0}^{\infty} \gamma^t \left\{ r(s_t,a_t) - \nu h_{s_t}\left(\pi(\cdot|s_t)\right) \right\} \,\middle|\, s_0 = s \right]$$

$$= V^\pi(s) - \frac{\nu}{1-\gamma} \sum_{s^e \in S} d_s^\pi(s^e) h_{s^e}\left(\pi(\cdot|s^e)\right), \tag{2.6}$$

where $h_s : \Delta_\alpha(A) \mapsto \mathbb{R}$ stands for a convex and possibly nonsmooth regularizer for state $s$, $\nu > 0$ denotes the regularization parameter, and $d_s^\pi(\cdot)$ is defined in (2.4). Here, for technical convenience, we assume throughout that $h_s(\cdot)$ $(s \in S)$ is well-defined over an "$\alpha$-neighborhood" of the probability simplex $\Delta(A)$ defined as follows:

$$\Delta_\alpha(A) := \left\{ x = [x_a]_{a \in A} \,\middle|\, x_a \geq 0 \text{ for all } a \in A;\ 1 - \alpha \leq \sum_{a \in A} x_a \leq 1 + \alpha \right\},$$

where $\alpha > 0$ can be an arbitrary constant. For instance, entropy regularization adopts the choice $h_s(p) = \sum_{i \in A} p_i \log p_i$ for all $s \in S$ and $p \in \Delta(A)$, which coincides with the negative Shannon entropy of a probability distribution. Similarly, a KL regularization adopts the choice $h_s(p) = KL(p \,|\, p_{\text{ref}})$, which penalizes the distribution $p$ that deviates from the reference $p_{\text{ref}}$. As another example, a weighted $l_1$ regularization adopts the choice $h_s(p) = \sum_{i \in A} w_{s,i} p_i$ for all $s \in S$ and $p \in \Delta(A)$, where $w_{s,i} \geq 0$ is the cost of taking action $i$ at state $s$, and the regularizer $h_s(\pi(\cdot|s))$ captures the expected cost of the policy $\pi$ in state $s$. Throughout this paper, we impose the following assumption.

Assumption 1. Consider an arbitrarily small constant $\alpha > 0$. For for any $s \in S$, suppose that $h_s(\cdot)$ is convex and

$$h_s(p) = \infty \quad \text{for any } p \notin \Delta_\alpha(A). \tag{2.7}$$

Following the convention in prior literature (see, e.g., [38]), we also define the corresponding regularized Q-function as follows:

$$\text{for } (s,a) \in S \times A: \quad Q_\tau^\pi(s,a) := r(s,a) + \gamma \mathop{\mathbb{E}}_{s^e \sim P(\cdot|s,a)} \left[ V_\tau^\pi(s^e) \right]. \tag{2.8a}$$

As can be straightforwardly verified, one can also express $V_\nu^\pi$ in terms of $Q_\nu^\pi$ as

$$\text{for } s \in S: \quad V_\nu^\pi(s) := \mathop{\mathbb{E}}_{a \sim \pi(\cdot|s)} \left[ Q_\nu^\pi(s,a) - \nu h_s\left(\pi(\cdot|s)\right) \right]. \tag{2.8b}$$

The optimal regularized value function $V_\nu^*$ and the corresponding optimal policy $\pi_\nu^*$ are defined, respectively, as follows:

$$\text{for } s \in S: \quad V_\nu^*(s) := V_\nu^{\pi_\nu^*}(s) = \max_\pi V_\nu^\pi(s), \qquad \pi_\nu^* := \arg\max_\pi V_\nu^\pi. \tag{2.9}$$

It is worth noting that the author of [43] asserts the existence of an optimal policy $\pi_\tau^*$ that achieves (2.9) simultaneously for all $s \in S$. Correspondingly, we shall also define the resulting optimal regularized Q-function as

$$\text{for } (s,a) \in S \times A: \quad Q_\nu^*(s,a) = Q_\nu^{\pi_\nu^*}(s,a). \tag{2.10}$$

2.2. Algorithm: Generalized policy mirror descent. Motivated by PMD [27], we put forward a generalization of PMD that selects the Bregman divergence in cognizance of the regularizer in use. A thorough comparison with [27] will be provided after introducing our generalized PMD algorithm.

For notational simplicity, we shall write

$$V_u^{(k)} := V_u^{i^{(k)}}, \qquad Q_u^{(k)}(s,a) := Q_u^{i^{(k)}}(s,a), \qquad \text{and} \qquad d_{s_0}^{(k)}(s) := d_{s_0}^{i^{(k)}}(s) \tag{2.11}$$

throughout the paper, where $i^{(k)}$ denotes our policy estimate in the $k$th iteration.

To begin with, suppose for simplicity that $h_s(t)$ is differentiable everywhere. In the $k$th iteration, a natural MD scheme that comes into mind for solving (2.6)--namely, maximize$_i$ $V_u^i(s_0)$ for a given initial state $s_0$--is the following update rule:

(2.12)

$$\begin{aligned} i^{(k+1)}(t\,|s) &= \arg\min_{p\in\mathcal{A}(\mathcal{A})} \Big\{ -\eta\, a_{i(t|s)} V_u^i(s_0)\big|_{i=i^{(k)}}, p\Big\rangle \\ &\quad + \frac{\iota}{1-\gamma} d_{s_0}^{(k)}(s) h_s(p) + \frac{1}{\eta} D_{h_s}\big(p, i^{(k)}(t\,|s)\big) \Big\} \\ &= \arg\min_{p\in\mathcal{A}(\mathcal{A})} \Big\{ \frac{1}{1-\gamma} d_{s_0}^{(k)}(s) \Big\{ -\eta\, Q_\iota^{(k)}(s,t), p\Big\rangle + \eta h_s(p) \Big\} + \frac{1}{\eta} D_{h_s}\big(p, i^{(k)}(t\,|s)\big) \Big\} \\ &= \arg\min_{p\in\mathcal{A}(\mathcal{A})} \Big\{ -\eta\, Q_\iota^{(k)}(s,t), p\Big\rangle + \eta h_s(p) + \frac{1}{\eta} D_{h_s}\big(p, i^{(k)}(t\,|s)\big) \Big\} \end{aligned}$$

for every state $s \in \mathcal{S}$. Here, we start with a learning rate $\eta$ and obtain simplification by replacing $\eta$ with $\eta(1-\gamma)/d_{s_0}^{(k)}(s)$. Notably, the update strategy (2.12) is invariant to the initial state $s_0$, akin to natural policy gradient methods [3].

This update rule is well-defined for, say, the case when $h_s$ is the negative entropy, since the algorithm guarantees $i^{(k)} > 0$ all the time and hence $h_s$ is always differentiable w.r.t. the $k$th iterate (see [12]). In general, however, it is possible to encounter situations when the gradient of $h_s$ does not exist on the boundary (e.g., when $h_s$ represents a certain indicator function). To cope with such cases, we resort to a generalized version of Bregman divergence (see, e.g., [26, 28, 29]). To be specific, we attempt to replace the usual Bregman divergence $D_{h_s}(p,q)$ by the following metric:

$$D_{h_s}(p,q;g_s) := h_s(p) - h_s(q) - \langle g_s, p-q\rangle \geq 0, \tag{2.13}$$

where $g_s$ can be any vector falling within the subdifferential $\partial h_s(q)$. Here, the nonnegativity condition in (2.13) follows directly from the definition of the subgradient for any convex function. The constraint on $g_s$ can be further relaxed by exploiting the requirement $p, q \in \mathcal{A}(\mathcal{A})$. In fact, for any vector $i_s = g_s - c_s\mathbf{1}$ (with $c_s \in \mathbb{R}$ some constant and $\mathbf{1}$ the all-one vector), one can readily see that

$$\begin{aligned} D_{h_s}(p,q;g_s) &= h_s(p) - h_s(q) - \langle g_s, p-q\rangle = h_s(p) - h_s(q) - \langle i_s, p-q\rangle + c_s\langle \mathbf{1}, p-q\rangle \\ &= h_s(p) - h_s(q) - \langle i_s, p-q\rangle = D_{h_s}(p,q;i_s), \end{aligned} \tag{2.14}$$

where the last line is valid since $\mathbf{1}^\top p = \mathbf{1}^\top q = 1$. As a result, everything boils down to identifying a vector $i_s$ that falls within $\partial h_s(q)$ upon global shift.

Towards this, we propose the following iterative rule for designing such a sequence of vectors as surrogates for the subgradient of $h_s$:

$$\pi^{(0)}(s,t) \in \partial h_s\left(\pi^{(0)}(t|s)\right); \tag{2.15a}$$

$$\pi^{(k+1)}(s,t) = \frac{1}{1+\alpha\tau}\pi^{(k)}(s,t) + \frac{\alpha}{1+\alpha\tau}Q_\tau^{(k)}(s,t), \qquad k \geq 0, \tag{2.15b}$$

where $\pi^{(k+1)}(s,t)$ is updated as a convex combination of the previous $\pi^{(k)}(s,t)$ and $Q_\tau^{(k)}(s,t)$, where more emphasis is put on $Q_\tau^{(k)}(s,t)$ when the learning rate $\alpha$ is large. As asserted by the following lemma, the above vectors $\pi^{(k)}(s,t)$ we construct satisfy the desired property, i.e., lying within the subdifferential of $h_s$ under suitable global shifts. It is worth mentioning that these global shifts $\{c_s^{(k)}\}$ only serve as an aid to better understand the construction but are not required during the algorithm updates.

Lemma 2.1. For all $k \geq 0$ and every $s \in \mathcal{S}$, there exists a quantity $c_s^{(k)} \in \mathbb{R}$ such that

$$\pi^{(k)}(s,t) - c_s^{(k)}\mathbf{1} \in \partial h_s\left(\pi^{(k)}(t|s)\right). \tag{2.16}$$

In addition, for every $s \in \mathcal{S}$, there exists a quantity $c_s^r \in \mathbb{R}$ such that

$$\tau^{-1}Q_\tau^r(s,t) - c_s^r\mathbf{1} \in \partial h_s\left(\pi_u^r(t|s)\right). \tag{2.17}$$

Proof. See Appendix A.1. □

Thus far, we have presented all crucial ingredients of our algorithm. The whole procedure is summarized in Algorithm 2.1 and will be referred to as Generalized Policy Mirror Descent (GPMD) throughout the paper. Interestingly, several well-known algorithms can be recovered as special cases of GPMD.

**Algorithm 2.1.** PMD with generalized Bregman divergence (GPMD).

**Input:** initial policy iterate $\pi^{(0)}$, learning rate $\alpha > 0$.

**Initialize** $\pi^{(0)}$ so that $\pi^{(0)}(s,t) \in \partial h_s(\pi^{(0)}(t|s))$ for all $s \in \mathcal{S}$.

**for** $k = 0, 1, \ldots,$ **do**

For every $s \in \mathcal{S}$, set

$$\pi^{(k+1)}(t|s) = \arg\min_{p \in \Delta(\mathcal{A})} \left\{ -\left\langle Q_\tau^{(k)}(s,t), p\right\rangle + \tau h_s(p) + \frac{1}{\alpha} D_{h_s}\left(p, \pi^{(k)}(t|s); \pi^{(k)}\right)\right\}, \tag{2.18a}$$

where

$$D_{h_s}\left(p, q; \pi\right) := h_s(p) - h_s(q) - \left\langle \pi(s,t), p - q\right\rangle. \tag{2.18b}$$

For every $(s,a) \in \mathcal{S} \times \mathcal{A}$, compute

$$\pi^{(k+1)}(s,a) = \frac{1}{1+\alpha\tau}\pi^{(k)}(s,a) + \frac{\alpha}{1+\alpha\tau}Q_\tau^{(k)}(s,a). \tag{2.18c}$$

**end for**

- When the Bregman divergence $D_{h_s}(t,t)$ is taken as the KL divergence, GPMD reduces to the well-renowned NPG algorithm [24] when $u = 0$ (no regularization) and to the NPG algorithm with entropy regularization analyzed in [12] when $h_s(t)$ is taken as the negative Shannon entropy.
- When $a = \gamma$ (no divergence), GPMD reduces to regularized policy itera-tion in [21]; in particular, GPMD reduces to the standard policy iteration algorithm if in addition $u$ is also 0.

Comparison with PMD [27]. Before continuing, let us take a moment to point out the key differences between our algorithm GPMD and the PMD algorithm proposed in [27] in terms of algorithm designs. Although the primary exposition of PMD in [27] fixes the Bregman divergence as the KL divergence, the algorithm also works in the presence of a generic Bregman divergence, whose relationship with the regularizer $h_s$ is, however, unspecified. Furthermore, GPMD adaptively sets this term to be the Bregman divergence generated by the regularizer $h_s$ in use, together with a carefully designed recursive update rule (cf. (2.15)) to compute surrogates for the subgradient of $h_s$ to facilitate implementation. Encouragingly, this specific choice leads to a tailored performance analysis of GPMD, which was not present in and instead complementary with that of PMD [27]. In truth, our theory offers linear convergence guarantees for more general scenarios by adapting to the geometry of the regularizer $h_s$; details will follow momentarily.

## 3. Main results.

This section presents our convergence guarantees for the GPMD method presented in Algorithm 2.1. We shall start with the idealized case, assuming that the update rule can be precisely implemented, and then discuss how to generalize it to the scenario with imperfect policy evaluation.

### 3.1. Convergence of exact GPMD.

To start with, let us pin down the convergence behavior of GPMD, assuming that accurate evaluation of the policy $Q_t^{(k)}$ is available and the subproblem (2.18a) can be solved perfectly. Here and below, we shall refer to the algorithm in this case as exact GPMD. Encouragingly, exact GPMD provably achieves global linear convergence from an arbitrary initialization, as asserted by the following theorem.

**Theorem 3.1** (exact GPMD). Suppose that Assumption 1 holds. Consider any learning rate $a > 0$, and set $a := \frac{1}{1+au}$. Then the iterates of Algorithm 2.1 satisfy

$$\left\|Q_t^r - Q_t^{(k+1)}\right\| \le a\left(1-(1-a)(1-a)\right)^k C_1, \tag{3.1a}$$

$$\left\|V_u^r - V_u^{(k+1)}\right\|_\gamma \le (a+2)\left(1-(1-a)(1-a)\right)^k C_1 \tag{3.1b}$$

for all $k \ge 0$, where $C_1 := \|Q_u^r - Q_t^{(0)}\|_\gamma + 2a\|Q_u^r - u\pi^{(0)}\|_\gamma$.

In addition, if $h_s$ is 1-strongly convex w.r.t. the $l_1$ norm for some $s \in S$, then one further has

$$\left\|\pi_u^r(s) - \pi_t^{(k+1)}(s)\right\|_1 \le u^{-1}\left(1-(1-a)(1-a)\right)^k C_1, \qquad k \ge 0. \tag{3.2}$$

Our theorem confirms the fast global convergence of the GPMD algorithm, in terms of both the resulting regularized Q-value (if $h_s(t)$ is convex) and the policy estimate (if $h_s(t)$ is strongly convex). In summary, it takes GPMD no more than

$$\frac{1}{(1-a)(1-a)}\log\frac{C_1}{\varepsilon} = \frac{1+au}{au(1-a)}\log\frac{C_1}{\varepsilon} \tag{3.3a}$$

iterations to converge to an n-optimal regularized Q-function (in the $l_y$ sense) or

$$\text{(3.3b)} \qquad \frac{1}{(1-a)(1-a)} \log \frac{C_1}{nu} = \frac{1+au}{au(1-a)} \log \frac{C_1}{nu}$$

iterations to yield an n-approximation (w.r.t. the $l_1$ norm error) of $i_t^r$. The iteration complexity (3.3) is nearly dimension-free--namely depending at most logarithmically on the dimension of the state-action space--making it scalable to large-dimensional problems.

To make clear our contributions, it is helpful to compare Theorem 3.1 with the theory for the state-of-the-art algorithm PMD in [27].

- Linear convergence for convex regularizers under constant learning rates. Suppose that constant learning rates are adopted for both GPMD and PMD. Our finding reveals that GPMD enjoys global linear convergence--in terms of both $|Q_t^r - Q_t^{(k+1)}|_y$ and $|V_t^r - V_u^{(k+1)}|_y$ --even when the regularizer $h_s(t)$ is only convex but not strongly convex. In contrast, [27, Theorem 2] provided only sublinear convergence guarantees (with an iteration complexity proportional to $1/n$) for the case with convex regularizers, provided that constant learning rates are adopted.[1]
- A full range of learning rates. Theorem 3.1 reveals linear convergence of GPMD for a full range of learning rates; namely, our result is applicable to any $a > 0$. In comparison, linear convergence was established in [27] only when the learning rates were suficiently large and when $h_s$ was 1-strongly convex w.r.t. the KL divergence. Consequently, the linear convergence results in [27] do not extend to several widely used regularizers, such as negative Tsallis entropy and log-barrier functions (even after scaling), which are, in contrast, covered by our theory. It is worth noting that the case with small-to-medium learning rates is often more challenging to cope with in theory, given that its dynamics could differ drastically from that of regularized policy iteration.
- Further comparison of rates under large learning rates. [27, Theorem 1] achieves a contraction rate of $a$ when the regularizer is strongly convex and the step size satisfies $a \geq \frac{1-a}{at}$, while the contraction rate of GPMD is $1 - \frac{au}{1+at}(1-a)$ under the full range of the step size, which is slower but approaches the contraction rate $a$ of PMD as $a$ goes to infinity. Therefore, in the limit $a \to \infty$, both GPMD and PMD achieve the contraction rate $a$. As soon as $a \geq 1/u$, their iteration complexities are on the same order.

3.2. Convergence of approximate GPMD. In reality, however, it is often the case that GPMD cannot be implemented in an exact manner, either because perfect policy evaluation is unavailable or because the subproblem (2.18a) cannot be solved exactly. To accommodate these practical considerations, this subsection generalizes our previous result by permitting inexact policy evaluation and non-zero optimization error in solving (2.18a). The following assumptions make precise this imperfect scenario.

Assumption 2 (policy evaluation error). Suppose, for any $k \geq 0$, we have access to an estimate $\widetilde{Q}_t^{(k)}$ obeying

$$\text{(3.4)} \qquad \left| \widetilde{Q}_u^{(k)} - Q_t^{(k)} \right|_y \leq \eta_{\text{eval}}.$$

[1] In fact, [27, Theorem 3] suggests using a vanishing strongly convex regularization, as well as a corresponding increasing sequence of learning rates, in order to enable linear convergence for non-strongly-convex regularizers.

**Algorithm 3.1.** Approximate PMD with generalized Bregman divergence (approximate GPMD).

Input: initial policy $\pi^{(0)}$, learning rate $\alpha > 0$.
Initialize $\widehat{Q}^{(0)}(s) \in \Pi h_s(\pi^{(0)}(\cdot|s))$ for all $s \in S$.
for $k = 0, 1, \ldots,$ do
For every $s \in S$, invoke the oracle to obtain (cf. (3.5))

$$\pi^{(k+1)}(s) = G_{s,\eta_{\mathrm{opt}}}\left(\widehat{Q}^{(k)}, \pi^{(k)}, \widehat{\pi}^{(k)}\right). \tag{3.6}$$

For every $(s,a) \in S \times A$, compute

$$\widehat{Q}^{(k+1)}(s,a) = \frac{1}{1+\alpha\mu}\widehat{Q}^{(k)}(s,a) + \frac{\alpha}{1+\alpha\mu}Q_{\iota}^{(k)}(s,a). \tag{3.7}$$

end for

Assumption 3 (subproblem optimization error). Consider any policy $\pi$ and any vector $\pi \in \mathbb{R}^{|S||A|}$. Define

$$f_s(p;\pi,\pi) := -\left\langle Q(s,\cdot), p\right\rangle + \mu h_s(p) + \frac{1}{\alpha}D_{h_s}\left(p, \pi(\cdot|s); \pi(s,\cdot)\right),$$

where $D_{h_s}(p,q;\pi)$ is defined in (2.13). Suppose there exists an oracle $G_{s,\eta_{\mathrm{opt}}}(Q,\pi,\pi)$ that is capable of returning $\pi^{e}(\cdot|s)$ such that

$$f_s\left(\pi^{e}(\cdot|s);\pi,\pi\right) \le \min_{p\in\Delta(A)} f_s(p;\pi,\pi) + \eta_{\mathrm{opt}}. \tag{3.5}$$

Note that the oracle in Assumption 3 can be implemented efficiently in practice via various first-order methods [6]. Under Assumptions 2 and 3, we can modify Algorithm 2.1 by replacing $\{Q_u^{(k)}\}$ with the estimate $\{\widehat{Q}_u^{(k)}\}$ and invoking the oracle $G_{s,\eta_{\mathrm{opt}}}(Q,\pi,\pi)$ to solve the subproblem (2.18a) approximately. The whole procedure, which we shall refer to as approximate GPMD, is summarized in Algorithm 3.1.

The following theorem uncovers that approximate GPMD converges linearly--at the same rate as exact GPMD--before an error floor is hit.

**Theorem 3.2** (approximate GPMD). Suppose that Assumptions 1, 2, and 3 hold. Consider any learning rate $\alpha > 0$. Then the iterates of Algorithm 3.1 satisfy

$$\|Q_u^{r} - Q_u^{(k+1)}\|_\infty \le \alpha\left[\left(1-(1-\alpha)(1-\alpha\mu)\right)^{k}C_1 + C_2\right], \tag{3.8a}$$

$$\|V_u^{r} - V_u^{(k+1)}\|_\infty \le (\alpha+2)\left[\left(1-(1-\alpha)(1-\alpha\mu)\right)^{k}C_1 + C_2\right] + (1-\alpha)\eta_{\mathrm{opt}}, \tag{3.8b}$$

where $\alpha := \frac{1}{1+\alpha\mu}$, $C_1$ is defined in Theorem 3.1, and

$$C_2 := \frac{1}{1-\alpha}\left[\left(2+\frac{2\alpha}{(1-\alpha)(1-\alpha\mu)}\right)\eta_{\mathrm{eval}} + \left(1+\frac{2\alpha}{(1-\alpha)(1-\alpha\mu)}\right)\eta_{\mathrm{opt}}\right].$$

In addition, if $h_s$ is 1-strongly convex w.r.t. the $l_1$ norm for any $s \in S$, then we can further obtain

$$\|Q_u^r - Q_u^{(k+1)}\|_\infty \le \alpha^k \left[\left(1 - (1-\alpha)(1-\tau)^k\right) C_1 + C_3\right], \tag{3.9a}$$

$$\|V_u^r - V_u^{(k+1)}\|_\infty \le (\alpha+2)\left[\left(1 - (1-\alpha)(1-\tau)^k\right) C_1 + C_3\right] + (1-\tau)^k n_{\text{opt}}, \tag{3.9b}$$

$$\left\|i_u^r(t\,|s) - i^{(k+1)}(t\,|s)\right\|_1 \le u^{-1}\left\{\left[\left(1 - (1-\alpha)(1-\tau)^k\right) C_1 + C_3\right] + \frac{2\alpha r_{\text{opt}}}{1+\alpha u}\right\}, \tag{3.9c}$$

where

$$C_3 := \frac{1}{1-\alpha}\left[\left(2 + \frac{n_{\text{eval}}\alpha}{u(1-\alpha)}\right) n_{\text{eval}} + \left(1 + \frac{4\alpha}{(1-\alpha)(1-\tau)}\right) n_{\text{opt}}\right]. \tag{3.10}$$

In the special case where $n_{\text{opt}} = 0$ and $\alpha = \gamma$, Algorithm 3.1 reduces to regularized policy iteration, and the convergence result can be simplified as follows:

$$\left\|Q_u^r - Q_u^{(k)}\right\|_\infty \le \gamma^k \left\|Q_u^r - Q_u^{(0)}\right\|_\infty + \frac{2\alpha\, n_{\text{eval}}}{(1-\alpha)^2}.$$

In particular, when $h_s$ is taken as the negative entropy, our result strengthens the prior result established in [12] for the approximate entropy-regularized NPG method with $n_{\text{opt}} = 0$ over a wide range of learning rates. Specifically, the error bound in [12] reads as $\frac{n_{\text{eval}}}{1-\alpha}\left(2 + \frac{2\alpha}{\alpha u}\right)$, where the second term in the brackets scales inversely with respect to $\alpha$ and therefore grows unboundedly as $\alpha$ approaches 0. In contrast, (3.9) and (3.10) suggest a bound $\frac{n_{\text{eval}}}{1-\alpha}\left(2 + \frac{n_{\text{eval}}\alpha}{u(1-\alpha)}\right)$, which is independent of the learning rate $\alpha$ in use and thus prevents the error bound from blowing up when the learning rate approaches 0. Indeed, our result improves over the prior art [12] whenever $\alpha \le \frac{2(1-\alpha)}{n_{\text{eval}}}$.

Remark 1 (sample complexities). One might naturally ask how many samples are sufficient to learn an $\varepsilon$-optimal regularized Q-function by leveraging sample-based policy evaluation algorithms in GPMD. Notice that it is straightforward to consider an expected version of Assumption 2 as follows:

$$\begin{cases} \mathbb{E}\left[\left\|\widehat{Q}_u^{(k)} - Q_u^{(k)}\right\|_\infty\right] \le n_{\text{eval}}; \\ \mathbb{E}\left[\left\|\widehat{Q}_u^{(k)} - Q_u^{(k)}\right\|_\infty^2\right] \le n_{\text{eval}}^2, \end{cases}$$

where the expectation is with respect to the randomness in policy evaluation; then the convergence results in Theorem 3.2 apply to $\mathbb{E}[\|Q_u^r - Q_u^{(k+1)}\|_\infty]$ and $\mathbb{E}[\|i_u^r(t\,|s) - i_u^{(k+1)}(t\,|s)\|_1]$ instead. This randomized version makes it immediately amenable to combine with, e.g., the rollout-based policy evaluators in [27, section 5.1], to obtain (possibly crude) bounds on the sample complexity. We omit these straightforward developments. ∎

Roughly speaking, approximate GPMD is guaranteed to converge linearly to an error bound that scales linearly in both the policy evaluation error $n_{\text{eval}}$ and the optimization error $n_{\text{opt}}$, thus confirming the stability of our algorithm vis-à-vis imperfect implementation of the algorithm. As before, our theory improves upon prior works by demonstrating linear convergence for a full range of learning rates even in the absence of strong convexity and smoothness.

4. Analysis for exact GPMD (Theorem 3.1). In this section, we present the analysis for our main result in Theorem 3.1, which follows a framework different from

[27]. Here and throughout, we shall often employ the following shorthand notation when it is clear from the context:

$$\begin{aligned} &\pi^{(k)}(s) := \pi^{(k)}(\cdot|s) \in \Delta(\mathcal{A}), && Q^{\pi}(s) := Q^{\pi}(s,\cdot) \in \mathbb{R}^{|\mathcal{A}|}, \\ &\pi^{(k)}(s) := \pi^{(k)}(s,\cdot) \in \mathbb{R}^{|\mathcal{A}|}, && Q^{\pi}_{u}(s) := Q^{\pi}_{u}(s,\cdot) \in \mathbb{R}^{|\mathcal{A}|}, \end{aligned} \tag{4.1}$$

in addition to those already defined in (2.11).

4.1. Preparation: Basic facts. In this subsection, we single out a few basic results that underlie the proof of our main theorems.

Performance improvement. To begin with, we demonstrate that GPMD enjoys a sort of monotonic improvement concerning the updates of both the value function and the Q-function, as stated in the following lemma. This lemma can be viewed as a generalization of the well-established policy improvement lemma in the analysis of NPG [3, 12] as well as PMD [27]. The proof can be found in [59].

Lemma 4.1 (pointwise monotonicity). For any $(s,a) \in \mathcal{S} \times \mathcal{A}$ and any $k \geq 0$, Algorithm 2.1 achieves

$$V_{u}^{(k+1)}(s) \geq V_{u}^{(k)}(s) \qquad \text{and} \qquad Q_{\iota}^{(k+1)}(s,a) \geq Q_{\iota}^{(k)}(s,a). \tag{4.2}$$

Interestingly, the above monotonicity holds simultaneously for all state-action pairs and hence can be understood as a kind of pointwise monotonicity.

Generalized Bellman operator. Another key ingredient of our proof lies in the use of a generalized Bellman operator $\mathcal{T}_{u,h} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ associated with the regularizer $h = \{h_s\}_{s \in \mathcal{S}}$. Specifically, for any state-action pair $(s,a)$ and any vector $Q \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we define

$$\mathcal{T}_{u,h}(Q)(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[ \max_{p \in \Delta(\mathcal{A})} \left\{ \langle Q(s'), p \rangle - u h_{s'}(p) \right\} \right]. \tag{4.3}$$

It is worth noting that this definition shares similarity with the regularized Bellman operator proposed in [21], where the operator defined there is targeted at $V_u$, while ours is defined w.r.t. $Q_u$.

The importance of this generalized Bellman operator is two-fold: it enjoys a desired contraction property, and its fixed point corresponds to the optimal regularized Q-function. These are generalizations of the properties for the classical Bellman operator, and are formally stated in the following lemma, whose proof can be found in [59].

Lemma 4.2 (properties of the generalized Bellman operator). For any $u > 0$, the operator $\mathcal{T}_{u,h}$ defined in (4.3) satisfies the following properties:

- $\mathcal{T}_{u,h}$ is a contraction operator w.r.t. the $\ell_\infty$ norm; namely, for any $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, one has

$$\left\| \mathcal{T}_{u,h}(Q_1) - \mathcal{T}_{u,h}(Q_2) \right\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty . \tag{4.4}$$

- The optimal regularized Q-function $Q_u^r$ is a fixed point of $\mathcal{T}_{u,h}$, that is,

$$\mathcal{T}_{u,h}(Q_u^r) = Q_u^r . \tag{4.5}$$

4.2. Proof of Theorem 3.1. Inspired by [12], our proof consists of (i) characterizing the dynamics of $\ell_\infty$ errors and establishing a connection to a useful linear system with two variables and (ii) analyzing the dynamics of this linear system directly. In what follows, we elaborate on each of these steps.

Step 1: Error contraction and its connection to a linear system. With the assistance of the above preparations, we are ready to elucidate how to characterize the convergence behavior of $|Q_\iota^r - Q_\iota^{(k+1)}|_y$. Recalling the update rule of $i^{(k+1)}$ (cf. (2.18c)), we can deduce that

$$Q_\iota^r - ui^{(k+1)} = a\left(Q_u^r - ui^{(k)}\right) + (1-a)\left(Q_u^r - Q_u^{(k)}\right)$$

with $a = \frac{1}{1+\bar{a}-\underline{u}}$, thus indicating that

$$|Q_\iota^r - ui^{(k+1)}|_y \le a|Q_\iota^r - ui^{(k)}|_y + (1-a)|Q_u^r - Q_\iota^{(k)}|_y. \tag{4.6}$$

Interestingly, there exists an intimate connection between $|Q_\iota^r - Q_\iota^{(k+1)}|_y$ and $|Q_\iota^r - ui^{(k+1)}|_y$ that allows us to bound the former term by the latter. This is stated in the following lemma, with the proof postponed to Appendix A.2.

Lemma 4.3. Set $a = \frac{1}{1+\bar{a}-\underline{u}}$. The iterates of Algorithm 2.1 satisfy

$$|Q_u^r - Q_\iota^{(k+1)}|_y \le a|Q_\iota^r - ui^{(k+1)}|_y + a a^{k+1}|Q_\iota^{(0)} - ui^{(0)}|_y. \tag{4.7}$$

The above inequalities (4.6) and (4.7) can be succinctly described via a useful linear system with two variables $|Q_u^r - Q_\iota^{(k)}|_y$ and $|Q_u^r - ui^{(k)}|_y$, that is,

$$x_{k+1} \le Ax_k + a a^{k+1} y, \tag{4.8}$$

where

$$A := \begin{bmatrix} a(1-a) & aa \\ 1-a & a \end{bmatrix}, \quad x_k := \begin{bmatrix} |Q_\iota^r - Q_\iota^{(k)}|_y \\ |Q_\iota^r - ui^{(k)}|_y \end{bmatrix}, \quad \text{and} \quad y := \begin{bmatrix} |Q_\iota^{(0)} - ui^{(0)}|_y \\ 0 \end{bmatrix}. \tag{4.9}$$

This forms the basis for proving Theorem 3.1.

Step 2: Analyzing the dynamics of the linear system (4.8). Before proceeding, we note that a linear system similar to (4.8) has been analyzed in [12, section 4.2.2]. We intend to apply the following properties that have been derived therein:

$$x_{k+1} \le A^{k+1}\left[x_0 + a(a^{-1}A - I)^{-1}y\right], \tag{4.10a}$$

$$a(a^{-1}A - I)^{-1}y = \begin{bmatrix} 0 \\ |Q_\iota^{(0)} - ui^{(0)}|_y \end{bmatrix}, \tag{4.10b}$$

$$A^{k+1} = \left((1-a)a + a\right)^k \begin{bmatrix} a \\ 1 \end{bmatrix}\begin{bmatrix} 1-a & a \end{bmatrix}. \tag{4.10c}$$

Substituting (4.10c) and (4.10b) into (4.10a) and rearranging terms, we reach

$$\begin{aligned} x_{k+1} &\le \left((1-a)a + a\right)^k\left((1-a)|Q_u^r - Q_\iota^{(0)}|_y\right) + a|Q_u^r - ui^{(0)}|_y + a|Q_u^{(0)} - ui^{(0)}|_y \end{aligned}\begin{bmatrix} a \\ 1 \end{bmatrix}$$
$$\le \left((1-a)a + a\right)^k\left(|Q_u^r - Q_u^{(0)}|_y + 2a|Q_u^r - ui^{(0)}|_y\right)\begin{bmatrix} a \\ 1 \end{bmatrix},$$

which taken together with the definition of $x_{k+1}$ gives

$$|Q_u^r - Q_u^{(k+1)}|_y \le a\left((1-a)a + a\right)^k\left(|Q_u^r - Q_u^{(0)}|_y + 2a|Q_u^r - ui^{(0)}|_y\right), \tag{4.11a}$$

$$|Q_\iota^r - ui^{(k+1)}|_y \le \left((1-a)a + a\right)^k\left(|Q_u^r - Q_u^{(0)}|_y + 2a|Q_u^r - ui^{(0)}|_y\right). \tag{4.11b}$$

Step 3: Controlling $\|\pi_u^r(s) - \pi^{(k+1)}(s)\|_1$ and $\|V_u^r - V_\iota^{(k+1)}\|_\infty$. It remains to convert this result to an upper bound on $\|\pi_u^r(s) - \pi^{(k+1)}(s)\|_1$ and $\|V_u^r - V_u^{(k+1)}\|_\infty$. By virtue of Lemma 2.1, there exist two vectors $g_u^r(s) \in \partial h_s(\pi_u^r(s))$, $g^{(k+1)}(s) \in \partial h_s(\pi^{(k+1)}(s))$ and two scalars $c_s^r, c_s^{(k+1)} \in \mathbb{R}$ that satisfy

$$\begin{cases} u^{-1}Q_u^r(s) - c_s^r \mathbf{1} &= g_u^r(s), \\ \pi^{(k+1)}(s,t) - c_s^{(k+1)}\mathbf{1} &= g^{(k+1)}(s). \end{cases}$$

It holds for all $s \in \mathcal{S}$ that

$$\begin{aligned} & V_u^r(s) - V_u^{(k+1)}(s) \\ &= \big\langle Q_u^r(s), \pi_u^r(s)\big\rangle - uh_s(\pi_u^r(s)) - \big\langle Q_u^{(k+1)}(s), \pi_u^{(k+1)}(s)\big\rangle + uh_s(\pi_u^{(k+1)}(s)) \\ &= \big\langle Q_u^r(s) - Q_u^{(k+1)}(s), \pi_u^{(k+1)}(s)\big\rangle \\ &\quad + u(h_s(\pi_u^{(k+1)}(s)) - h_s(\pi_u^r(s))) - \big\langle Q_u^r(s), \pi_u^{(k+1)}(s) - \pi_u^r(s)\big\rangle \qquad (4.12) \\ &\le \big\langle Q_u^r(s) - Q_\iota^{(k+1)}(s), \pi_\iota^{(k+1)}(s)\big\rangle + u\big\langle g^{(k+1)}(s) - Q_\iota^r(s), \pi_\iota^{(k+1)}(s) - \pi_\iota^r(s)\big\rangle \\ &= \big\langle Q_u^r(s) - Q_\iota^{(k+1)}(s), \pi_\iota^{(k+1)}(s)\big\rangle + u\big\langle \pi^{(k+1)}(s) - Q_\iota^r(s), \pi_\iota^{(k+1)}(s) - \pi_\iota^r(s)\big\rangle \\ &\le \|Q_u^r(s) - Q_u^{(k+1)}(s)\|_\infty + 2\|Q_u^r(s) - u\pi^{(k+1)}(s)\|_\infty, \qquad (4.13) \end{aligned}$$

where (i) results from $h_s(\pi_u^{(k+1)}(s)) - h_s(\pi_u^r(s)) \ge \langle g^{(k+1)}(s), \pi_u^{(k+1)}(s) - \pi_u^r(s)\rangle$. Plugging (4.11) into (4.13) completes the proof for (3.1b).

When $h_s$ is 1-strongly convex w.r.t. the $\ell_1$ norm, we can invoke the strong monotonicity property of a strongly convex function [6, Theorem 5.24] to obtain

$$\begin{aligned} \|\pi_u^r(s) - \pi^{(k+1)}(s)\|_1^2 &\le \big\langle \pi_\iota^r(s) - \pi^{(k+1)}(s), g_\iota^r(s) - g^{(k+1)}(s)\big\rangle \\ &= \big\langle \pi_u^r(s) - \pi^{(k+1)}(s), g_\iota^r(s) + c_s^r\mathbf{1} - g^{(k+1)}(s) - c_s^{(k+1)}\mathbf{1}\big\rangle \\ &\le \|\pi_u^r(s) - \pi^{(k+1)}(s)\|_1 \|g_\iota^r(s) + c_s^r\mathbf{1} - g^{(k+1)}(s) - c_s^{(k+1)}\mathbf{1}\|_\infty \\ &= u^{-1}\|\pi_u^r(s) - \pi^{(k+1)}(s)\|_1 \|Q_\iota^r(s) - u\pi^{(k+1)}(s)\|_\infty, \qquad (4.14) \end{aligned}$$

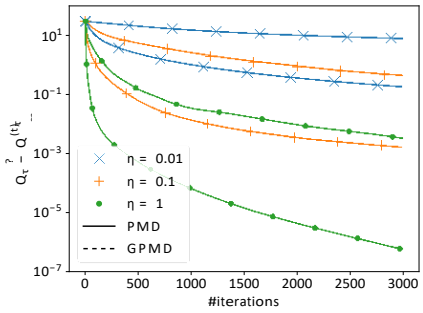
where the second line is valid since $\langle \pi_\iota^r(s), \mathbf{1}\rangle = \langle \pi^{(k+1)}(s), \mathbf{1}\rangle = 1$. This taken together with (4.11) gives rise to the advertised bound

$$\begin{aligned} \|\pi_\iota^r(s) - \pi^{(k+1)}(s)\|_1 &\le u^{-1}\|Q_\iota^r(s) - u\pi^{(k+1)}(s)\|_\infty \\ &\le u^{-1}(1-\alpha)^k\big(\|Q_u^r - Q_u^{(0)}\|_\infty + 2\alpha\|Q_u^r - u\pi^{(0)}\|_\infty\big). \end{aligned}$$

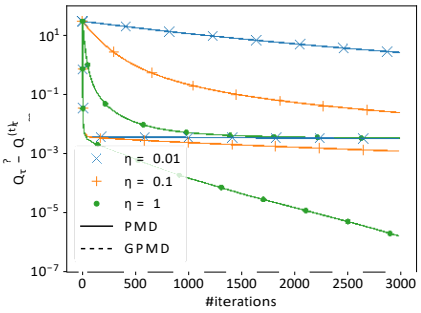
5. Numerical experiments. In this section, we provide some simple numerical experiments to corroborate the effectiveness of the GPMD algorithm.

5.1. Tsallis entropy. While Shannon entropy is a popular choice of regularization, the discrepancy between the value function of the regularized MDP and the unregularized counterpart scales as $O(\frac{u}{1-\alpha}\log|\mathcal{A}|)$. In addition, the optimal policy under Shannon entropy regularization assigns positive mass to all actions and is hence nonsparse. To promote sparsity and obtain better control of the bias induced by regularization, the authors of [31, 32] proposed to employ the Tsallis entropy [49] as an alternative. To be precise, for any vector $p \in \Delta(\mathcal{A})$, the associated Tsallis entropy is defined as

$$\mathrm{Tsallis}_q(p) = \frac{1}{q-1}\Big(1 - \sum_{a\in\mathcal{A}} (p(a))^q\Big) = \frac{1}{q-1}\mathbb{E}_{a\sim p}\big[1 - (p(a))^{q-1}\big],$$




(a) Tsallis entropy regularization




(b) Log-barrier regularization

Fig. 1. $|Q_\iota^r - Q_\iota^{(t)}|_\gamma$ versus the iteration count for both PMD and GPMD, for multiple choices of the learning rate a. The left plot (a) is concerned with Tsallis entropy regularization, whereas the right plot (b) concerns log-barrier regularization used in our constrained RL example. The error curves are averaged over five independent runs.

where $q > 0$ is often referred to as the entropic index. When $q \to 1$, the Tsallis entropy reduces to the Shannon entropy.

We now evaluate numerically the performance of PMD and GPMD when applied to a randomly generated MDP with $|S| = 200$ and $|A| = 50$. Here, the transition probability kernel and the reward function are generated as follows. For each state-action pair $(s, a)$, we randomly select 20 states to form a set $S_{s,a}$, and set $P(s^e|s, a) = 1/20$ if $s^e \in S_{s,a}$, and 0 otherwise. The reward function is generated by $r(s, a) = U_{s,a} \cdot U_s$, where $U_{s,a}$ and $U_s$ are independent uniform random variables over $[0, 1]$. We shall set the regularizer as $h_s(p) = -\text{Tsallis}_2(p)$ for all $s \in S$ with a regularization parameter $\tau = 0.001$. As can be seen from the numerical results displayed in Figure 1(a), GPMD enjoys a faster convergence rate compared to PMD.

5.2. Constrained RL. In reality, an agent with the sole aim of maximizing cumulative rewards might sometimes end up with unintended or even harmful behavior, due to, say, improper design of the reward function or nonperfect simulation of physical laws. Therefore, it is sometimes necessary to enforce proper constraints on the policy in order to prevent it from taking certain actions too frequently.

To simulate this problem, we first solve an MDP with $|S| = 200$ and $|A| = 50$, generated in the same way as in the previous subsection. We then pick 10 state-action pairs from the support of the optimal policy at random to form a set $\mathcal{I}$. We can ensure that $\pi^r(a|_\iota s) < \pi_{\max} = 0.1$ for all $(s, a) \in \mathcal{I}$ by adding the following log-barrier regularization with $\tau = 0.001$:

$$h_s(p) = \begin{cases} \infty & \text{if } (s, a) \in \mathcal{I} \text{ and } p(a) \geq \pi_{\max}, \\ -\log\left(\pi_{\max} - p(a)\right) & \text{if } (s, a) \in \mathcal{I} \text{ and } p(a) < \pi_{\max}, \\ 0 & \text{otherwise.} \end{cases}$$

Numerical comparisons of PMD and GPMD when applied this problem are plotted in Figure 1(b). It is observed that PMD methods stall after reaching an error floor on the order of $10^{-2}$, while GPMD methods are able to converge to the optimal policy eficiently.

6. Discussion. The present paper has introduced a generalized framework of policy optimization tailored to regularized RL problems. We have proposed a

Generalized Policy Mirror Descent (GPMD) algorithm that achieves dimension-free linear convergence, which covers an entire range of learning rates and accommodates convex and possibly nonsmooth regularizers. Numerical experiments have been conducted to demonstrate the utility of the proposed GPMD algorithm. Our approach opens up a couple of future directions that are worthy of further exploration. For example, the current work restricts its attention to convex regularizers and tabular MDPs; it is of paramount interest to develop policy optimization algorithms when the regularizers are nonconvex and when sophisticated policy parameterization--including function approximation--is adopted. Understanding the sample complexities of the proposed algorithm--when the policies are evaluated using samples collected over an online trajectory--is crucial in sample-constrained scenarios and is left for future investigation. Furthermore, it might be worthwhile to extend the proposed algorithm to accommodate multi-agent RL, with a representative example being regularized multi-agent Markov games [14, 64, 11, 13].

Appendix A. Proof of key lemmas. In this section, we collect the proof of several key lemmas. Here and throughout, we use $E_{\pi}[\cdot]$ to denote the expectation over the randomness of the MDP induced by policy $\pi$. We shall follow the notation convention in (4.1) throughout. In addition, to further simplify notation, we shall abuse the notation by letting

$$D_{h_s}(\dot{e}, \pi; \pi) := D_{h_s}\big(\dot{e}(\cdot|s), \pi(\cdot|s); \pi(s,t)\big), \tag{A.1a}$$

$$D_{h_s}(p, \pi; \pi) := D_{h_s}\big(p, \pi(\cdot|s); \pi(s,t)\big), \tag{A.1b}$$

$$D_{h_s}(\pi, p; \pi) := D_{h_s}\big(\pi(\cdot|s), p; \pi(s,t)\big) \tag{A.1c}$$

for any policy $\pi$ and $\dot{e}$ and any $p \in \Delta(\mathcal{A})$, whenever it is clear from the context.

A.1. Proof of Lemma 2.1 We start by relaxing the probability simplex constraint (i.e., $p \in \Delta(\mathcal{A})$) in (2.18a) with a simpler linear constraint $\sum_{a\in\mathcal{A}} p(a) = 1$ as follows:

$$\begin{aligned} &\text{minimize}_{p\in\mathbb{R}^{|\mathcal{A}|}} && -\big\langle Q_{\iota}^{(k)}(s), p\big\rangle + \alpha\iota h_s(p) + D_{h_s}\big(p, \pi^{(k)}; \pi^{(k)}\big) \\ &\text{subject to} && \sum_{a\in\mathcal{A}} p(a) = 1. \end{aligned} \tag{A.2}$$

To justify the validity of dropping the non-negative constraint, we note that for any $p$ obeying $p(a) < 0$ for some $a \in \mathcal{A}$, our assumption on $h_s$ (see Assumption 1) leads to $h_s(p) = \infty$, which cannot possibly be the optimal solution. This confirms the equivalence between (2.18a) and (A.2).

Observe that the Lagrangian w.r.t. (A.2) is given by

$$\begin{aligned} L_s\big(p, a_s^{(k)}\big) &= -\big\langle Q_{\iota}^{(k)}(s), p\big\rangle + \alpha\iota h_s(p) + h_s(p) - h_s\big(\pi^{(k)}(s)\big) - \big\langle p - \pi^{(k)}(s), \nabla h_s(\pi^{(k)}(s))\big\rangle \\ &\quad + a_s^{(k)}\Big(\sum_{a\in\mathcal{A}} p(a) - 1\Big), \end{aligned}$$

where $a_s^{(k)} \in \mathbb{R}$ denotes the Lagrange multiplier associated with the constraint $\sum_{a\in\mathcal{A}} p(a) = 1$. Given that $\pi^{(k+1)}(s)$ is the solution to (2.18a) and hence (A.2), the optimality condition requires that

$$0 \in \partial_p L_s\big(p, a_s^{(k)}\big)\Big|_{p=\pi^{(k+1)}(s)} = -\alpha Q_u^{(k)}(s) + (1+\alpha u)\partial h_s\big(\pi^{(k+1)}(s)\big) - \pi^{(k)}(s) + a_s^{(k)}\mathbf{1}.$$

Rearranging terms and making use of the construction (2.15), we are left with

$$i^{(k+1)}(s) - \frac{\hat{a}_s^{(k)}}{1+au}1 = \frac{1}{1+au}\left[aQ_u^{(k)}(s) + i^{(k)}(s) - a_s^{(k)}1\right] \in \partial h_s\left(i^{(k+1)}(s)\right),$$

thus concluding the proof of the first claim (2.16).

We now turn to the second claim (2.17). In view of the property (4.5), we have

$$i_u^r(s) = \arg\min_{p \in \mathcal{A}} -\left\langle Q_u^r(s), p\right\rangle + uh_s(p).$$

This optimization problem is equivalent to

$$\begin{aligned} &\text{minimize}_{p \in \mathbb{R}^{|\mathcal{A}|}} && -\left\langle Q_u^r(s), p\right\rangle + uh_s(p) \\ &\text{subject to} && \sum_{a \in \mathcal{A}} p(a) = 1, \end{aligned} \tag{A.3}$$

which can be verified by repeating a similar argument for (A.2). The Lagrangian associated with (A.3) is

$$L_s\left(p, a_s^r\right) = -\left\langle Q_u^r(s), p\right\rangle + uh_s(p) + a_s^r\left(\sum_{a \in \mathcal{A}} p(a) - 1\right),$$

where $a_s^r \in \mathbb{R}$ denotes the Lagrange multiplier. Therefore, the first-order optimality condition requires that

$$0 \in \partial_p L_s\left(p, a_s^r\right)\Big|_{p = i_u^r(s)} = -Q_u^r(s) + u\partial h_s\left(i_t^r(s)\right) + a_s^r 1,$$

which immediately finishes the proof.

A.2. Proof of Lemma 4.3. Recall that $Q_t^{(k+1)} = Q_t^{i^{(k+1)}}$. In view of the relation (2.8), one obtains

$$\begin{aligned} Q_t^{(k+1)}(s,a) &= r(s,a) + a \mathop{\mathbb{E}}_{s^e \sim P(t|s,a)}\left[V_t^{(k+1)}(s^e)\right] \\ &= r(s,a) + a \mathop{\mathbb{E}}_{s^e \sim P(t|s,a)}\left[\mathop{\mathbb{E}}_{a^e \sim i^{(k+1)}(s^e)}\left[Q_t^{(k+1)}(s^e,a^e) - uh_{s^e}\left(i^{(k+1)}(s^e)\right)\right]\right] \\ &= r(s,a) + a \mathop{\mathbb{E}}_{s^e \sim P(t|s,a)}\left\langle Q_t^{(k+1)}(s^e), i^{(k+1)}(s^e)\right\rangle - uh_{s^e}\left(i^{(k+1)}(s^e)\right). \end{aligned}$$

This combined with the fixed-point condition (4.5) allows us to derive

$$\begin{aligned} &Q_u^r(s,a) - Q_t^{(k+1)}(s,a) \\ &\quad = T_{u,h}(Q_u^r)(s,a) \\ &\qquad - \left\{r(s,a) + a \mathop{\mathbb{E}}_{s^e \sim P(t|s,a)}\left[\left\langle Q_t^{(k+1)}(s^e), i^{(k+1)}(s^e)\right\rangle - uh_{s^e}\left(i^{(k+1)}(s^e)\right)\right]\right\} \\ &\quad = T_{u,h}(Q_u^r)(s,a) \\ &\qquad - \left\{r(s,a) + a \mathop{\mathbb{E}}_{s^e \sim P(t|s,a)}\left[\left\langle ui^{(k+1)}(s^e), i^{(k+1)}(s^e)\right\rangle - uh_{s^e}\left(i^{(k+1)}(s^e)\right)\right]\right\} \\ &\qquad - a \mathop{\mathbb{E}}_{s^e \sim P(t|s,a), a^e \sim i^{(k+1)}(s^e)}\left[Q_t^{(k+1)}(s^e,a^e) - ui^{(k+1)}(s^e,a^e)\right]. \end{aligned} \tag{A.4}$$

In what follows, we control each term on the right-hand side of (A.4) separately.

Step 1: Bounding the 1st term on the right-hand side of (A.4). Lemma 2.1 tells us that

$$i^{(k+1)}(s) - c_s^{(k+1)} 1 \in \partial h_s(i^{(k+1)}(s))$$

for some scalar $c_s^{(k+1)} \in \mathbb{R}$. This important property allows one to derive

$$0 \in -i^{(k+1)}(s) + c_s^{(k+1)} 1 + \partial h_s\left(i^{(k+1)}(s)\right) = \partial L_{k+1,s}\left(i^{(k+1)}(s); c_s^{(k+1)}\right), \tag{A.5}$$

where

$$L_{k+1,s}(p;\alpha) := \underbrace{\langle i^{(k+1)}(s), p\rangle + h_s(p)}_{=: f_{k+1,s}(p)} + \alpha 1^\top p.$$

Recognizing that the function $f_{k+1,s}(t)$ is convex in $p$, we can view $L_{k+1,s}(p;\alpha)$ as the Lagrangian of the following constrained convex problem with Lagrangian multiplier $\alpha \in \mathbb{R}$:

$$\underset{p:1^\top p=1}{\text{minimize}} \quad f_{k+1,s}(p) = -\langle i^{(k+1)}(s), p\rangle + h_s(p). \tag{A.6}$$

The condition (A.5) can then be interpreted as the optimality condition w.r.t. the program (A.6) and $i^{(k+1)}(s)$, meaning that

$$f_{k+1,s}\left(i^{(k+1)}(s)\right) = \min_{p:1^\top p=1} f_{k+1,s}(p)$$

or, equivalently,

$$\langle i^{(k+1)}(s), i^{(k+1)}(s)\rangle - h_s\left(i^{(k+1)}(s)\right) = \max_{p:1^\top p=1} \langle i^{(k+1)}(s), p\rangle - h_s(p). \tag{A.7}$$

In addition, for any vector $p$ that does not obey $p \geq 0$, Assumption 1 implies that $h_s(p) = \infty$, and hence $p$ cannot possibly be the optimal solution to $\max_{p\in\mathrm{dom}(h_s)} \langle i^{(k+1)}(s), p\rangle - h_s(p)$. This together with (A.7) essentially implies that

$$\langle i^{(k+1)}(s), i^{(k+1)}(s)\rangle - h_s\left(i^{(k+1)}(s)\right) = \max_{p\in\mathrm{dom}(h_s)} \langle i^{(k+1)}(s), p\rangle - h_s(p). \tag{A.8}$$

As a consequence, we arrive at

(A.9)

$$\begin{aligned}
&T_{u,h}(Q_u^r)(s,a) \\
&\quad = \Big\{ r(s,a) + \alpha \mathop{\mathbb{E}}_{s'\sim P(t|s,a)} \big[ \langle u_i^{(k+1)}(s'), i^{(k+1)}(s')\rangle - u h_{s'}\left(i^{(k+1)}(s')\right) \big] \Big\} \\
&\quad = T_{u,h}(Q_\iota^r)(s,a) - \Big\{ r(s,a) + \alpha \mathop{\mathbb{E}}_{s'\sim P(t|s,a)} \Big[ \max_{p\in\mathrm{dom}(h_s)} \big\{ \langle u_i^{(k+1)}(s'), p\rangle - u h_{s'}(p) \big\} \Big] \Big\} \\
&\quad = T_{u,h}(Q_\iota^r)(s,a) - T_{u,h}(u_i^{(k+1)})(s,a) \\
&\quad \leq \alpha \left\| Q_u^r - u_i^{(k+1)} \right\|_\infty,
\end{aligned}$$

where the last step results from the contraction property (4.4) in Lemma 4.2.

Step 2: Bounding the 2nd term on the right-hand side of (A.4). Recall that $a = \frac{1}{1+a\iota}$. Invoking the monotonicity property in Lemma 4.1 and the update rule (2.18c), we obtain

$$\begin{aligned} &Q_\iota^{(k+1)}(s,a) - \mathrm{ui}^{(k+1)}(s,a) \\ &\quad = a\left\{Q_u^{(k+1)}(s,a) - \mathrm{ui}^{(k)}(s,a)\right\} + (1-a)\left\{Q_u^{(k+1)}(s,a) - Q_u^{(k)}(s,a)\right\} \\ &\quad \le a\left\{Q_\iota^{(k)}(s,a) - \mathrm{ui}^{(k)}(s,a)\right\}. \end{aligned}$$

Repeating this lower bound argument then yields

$$\begin{aligned} Q_u^{(k+1)}(s,a) - \mathrm{ui}^{(k+1)}(s,a) &\le a^{k+1}\left\{Q_u^{(0)}(s,a) - \mathrm{ui}^{(0)}(s,a)\right\} \\ &\le a^{k+1}\left|Q_\iota^{(0)} - \mathrm{ui}^{(0)}\right|_\infty, \end{aligned}$$

thus revealing that

$$\text{(A.10)}\qquad -\mathop{\mathbb{E}}_{s^e\sim P(\cdot|s,a),a^e\sim\mu^{k+1}(s^e)}\left[Q_\iota^{(k+1)}(s^e,a^e) - \mathrm{ui}^{(k+1)}(s^e,a^e)\right] \le a^{k+1}\left|Q_\iota^{(0)} - \mathrm{ui}^{(0)}\right|_\infty.$$

Step 3: Putting all this together. Substituting (A.9) and (A.10) into (A.4) gives

$$\text{(A.11)}\qquad 0 \le Q_\iota^r(s,a) - Q_\iota^{(k+1)}(s,a) \le a\left|Q_\iota^r - \mathrm{ui}^{(k+1)}\right|_\infty + a^{k+1}\left|Q_\iota^{(0)} - \mathrm{ui}^{(0)}\right|_\infty$$

for all $(s,a) \in S \times A$, thus concluding the proof.

Appendix B. Analysis for approximate GPMD (Theorem 3.2). The proof consists of three steps: (i) evaluating the performance difference between $\mu^{(k)}$ and $\mu^{(k+1)}$, (ii) establishing a linear system to characterize the error dynamic, and (iii) analyzing this linear system to derive global convergence guarantees. We shall describe the details of each step in what follows. As before, we adopt the notational convention (A.1) whenever it is clear from the context.

B.1. Step 1: Bounding performance difference between consecutive iterates. When only approximate policy evaluation is available, we are no longer guaranteed to have pointwise monotonicity as in the case of Lemma 4.1. Fortunately, we are still able to establish an approximate versioin of Lemma 4.1, as stated below.

Lemma B.1 (performance improvement for approximate GPMD). For all $s \in S$ and all $k \ge 0$, we have

$$V_u^{(k+1)}(s) \le V_u^{(k)}(s) - \frac{1+a}{1-a}\eta_{\mathrm{opt}} - \frac{2}{1-a}\eta_{\mathrm{eval}}.$$

In addition, if $h_s$ is 1-strongly convex w.r.t. the $l_1$ norm for all $s \in S$, then one further has

$$V_u^{(k+1)}(s) \le V_u^{(k)}(s) - \frac{3+a}{1-a}\eta_{\mathrm{opt}} - \frac{\varepsilon}{(2+au)(1-a)}\eta_{\mathrm{eval}}^2.$$

In words, while monotonicity is not guaranteed, this lemma precludes the possibility of $V_u^{(k+1)}(s)$ being much smaller than $V_u^{(k)}(s)$, as long as both $\eta_{\mathrm{eval}}$ and $\eta_{\mathrm{opt}}$ are reasonably small.

B.1.1. Proof of Lemma B.1. We divide the proof into two cases based on whether $h_s$ is convex or strongly convex.

The case when $h_s$ is convex. Let $\dot{e}^{(k+1)}$ be the exact solution of the following problem:

$$\dot{e}^{(k+1)}(s) = \arg\min_{p \in \Delta(A)} \left\{ \langle Q_\iota^{(k)}(s), p \rangle + h_s(p) + \frac{1}{\tilde{c}} D_{h_s}\left(p, \imath^{(k)}; \imath_t^{(k)}\right) \right\}. \tag{B.1}$$

With this auxiliary policy iterate $\dot{e}^{(k+1)}$ in mind, we start by decomposing $V_\iota^{(k+1)}(s) - V_u^{(k)}(s)$ into the following three parts:

$$\begin{aligned}
&(1-\alpha)\left(V_u^{(k+1)}(s) - V_\iota^{(k)}(s)\right) \\
&= \mathop{\mathbb{E}}_{s^e \cdots \mathcal{d}_s^{(k+1)}} \left[ \langle Q_\iota^{(k)}(s^e), \imath^{(k+1)}(s^e) - \imath^{(k)}(s^e) \rangle - h_{s^e}\left(\imath^{(k+1)}(s^e)\right) + h_{s^e}\left(\imath^{(k)}(s^e)\right) \right] \\
&= \mathop{\mathbb{E}}_{s^e \cdots \mathcal{d}_s^{(k+1)}} \left[ \langle Q_\iota^{(k)}(s^e), \dot{e}^{(k+1)}(s^e) - \imath^{(k)}(s^e) \rangle - h_{s^e}\left(\dot{e}^{(k+1)}(s^e)\right) + h_{s^e}\left(\imath^{(k)}(s^e)\right) \right] \\
&+ \mathop{\mathbb{E}}_{s^e \mathrm{m} d_s^{(k+1)}} \left[ \langle Q_\iota^{(k)}(s^e), \imath^{(k+1)}(s^e) - \dot{e}^{(k+1)}(s^e) \rangle - h_{s^e}\left(\imath^{(k+1)}(s^e)\right) + h_{s^e}\left(\dot{e}^{(k+1)}(s^e)\right) \right] \\
&+ \mathop{\mathbb{E}}_{s^e \mathrm{m} d_s^{(k+1)}} \left[ \epsilon Q_\iota^{(k)}(s^e) - Q_\iota^{(k)}(s^e), \imath^{(k+1)}(s^e) - \imath^{(k)}(s^e) \epsilon \right],
\end{aligned} \tag{B.2}$$

where the first identity arises from the performance difference lemma for the regularized setting [27]. To continue, we seek to control each part of (B.2) separately.

- Regarding the first term of (B.2), straightforward computation indicates that

$$\begin{aligned}
&\epsilon \langle Q_\iota^{(k)}(s^e), \imath^{(k+1)}(s^e) - \imath^{(k)}(s^e) \rangle - h_{s^e}\left(\imath^{(k+1)}(s^e)\right) + h_{s^e}\left(\imath^{(k)}(s^e)\right) \epsilon \\
&= \frac{1}{\tilde{c}} (1+\alpha) D_{h_{s^e}}\left(\imath^{(k)}, \dot{e}^{(k+1)}(s^e); \imath_t^{(k+1)}\right) + D_{h_{s^e}}\left(\dot{e}^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right)
\end{aligned} \tag{B.3}$$

  for all $s^e \in S$.

- As for the second term of (B.2), the definition of the oracle $G_{s,n_{\mathrm{opt}}}$ (see Assumption 3) guarantees that

$$\begin{aligned}
&- \epsilon \langle Q_\iota^{(k)}(s^e), \imath^{(k+1)}(s^e) \rangle + h_{s^e}\left(\imath^{(k+1)}(s^e)\right) + \frac{1}{\tilde{c}} D_{h_{s^e}}\left(\imath^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) \\
&\quad \leq \epsilon \langle Q_\iota^{(k)}(s^e), \dot{e}^{(k+1)}(s^e) \rangle + h_{s^e}\left(\dot{e}^{(k+1)}(s^e)\right) + \frac{1}{\tilde{c}} D_{h_{s^e}}\left(\dot{e}^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) \\
&\qquad + n_{\mathrm{opt}}
\end{aligned}$$

  for any $s^e \in S$. Rearranging terms, we are left with

$$\begin{aligned}
&\epsilon \langle Q_\iota^{(k)}(s^e), \imath^{(k+1)}(s^e) - \dot{e}^{(k+1)}(s^e) \rangle - h_{s^e}\left(\imath^{(k+1)}(s^e)\right) + h_{s^e}\left(\dot{e}^{(k+1)}(s^e)\right) \\
&\quad \leq -\frac{1}{\alpha} D_{h_{s^e}}\left(\dot{e}^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) + \frac{1}{\alpha} D_{h_{s^e}}\left(\imath^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) - n_{\mathrm{opt}} \\
&\quad = -\frac{1}{\tilde{c}} \left( D_{h_{s^e}}\left(\dot{e}^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) - D_{h_{s^e}}\left(\imath^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) \right) \\
&\qquad + \frac{1}{\tilde{c}} \left( D_{h_{s^e}}\left(\imath^{(k+1)}, \imath^{(k)}; \imath_t^{(k)}\right) - D_{h_{s^e}}\left(\imath^{(k+1)}, \dot{e}^{(k)}; \imath_t^{(k)}\right) \right) - n_{\mathrm{opt}}.
\end{aligned} \tag{B.5}$$

In addition, we note that the term

$$D_{h_{s^e}}(\dot{e}^{(k+1)}, i^{(k)}; i\dot{t}^{(k)})$$

appears in both (B.3) and (B.5), which can be canceled out when summing these two equalities. Specifically, adding (B.3) and (B.5) gives

$$\begin{aligned}
&\langle \dot{Q}_t^{(k)}(s^e), \dot{e}^{(k+1)}(s^e) - i^{(k)}(s^e)\rangle_e - u h_{s^e}(\dot{e}^{(k+1)}(s^e)) + u h_{s^e}(i^{(k)}(s^e)) \\
&\quad + [\dot{Q}_t^{(k)}(s^e), i^{(k+1)}(s^e) - \dot{e}^{(k+1)}(s^e)\rangle_e - u h_{s^e}(i^{(k+1)}(s^e)) + u h_{s^e}(\dot{e}^{(k+1)}(s^e)) \\
&\quad \le \frac{1}{\tau}(1+au) D_{h_{s^e}}(i^{(k)}, \dot{e}^{(k+1)}; i\dot{t}^{(k+1)}) + D_{h_{s^e}}(i^{(k+1)}, i\dot{e}^{(k)}; i\dot{t}^{(k)}) \\
&\quad + \frac{1}{\tau}\left( D_{h_{s^e}}(i^{(k+1)}, i^{(k)}; i\dot{t}^{(k)}) - D_{h_{s^e}}(i^{(k+1)}, \dot{e}^{(k)}; i\dot{t}^{(k)}) \right) - n_{opt}.
\end{aligned}$$

Substituting this into (B.2) and invoking the elementary inequality $|\langle a, b\rangle| \le |a|_1 |b|_\infty$ thus lead to

(B.6)

$$\begin{aligned}
&V_u^{(k+1)}(s) - V_u^{(k)}(s) \\
&\le \frac{1}{1-a} E_{s^e \sim d_s^{(k+1)}}\left[ \frac{1}{\tau}\left[ (1+au) D_{h_{s^e}}(i^{(k)}, \dot{e}^{(k+1)}; i\dot{t}^{(k+1)}) + D_{h_{s^e}}(i^{(k+1)}, \dot{e}^{(k)}; i\dot{t}^{(k)}) \right]\right] \\
&\quad + \frac{1}{1-a} E_{s^e \sim d_s^{(k+1)}}\left[ \frac{1}{\tau}\left( D_{h_{s^e}}(i^{(k+1)}, i^{(k)}; i\dot{t}^{(k)}) - D_{h_{s^e}}(i^{(k+1)}, \dot{e}^{(k)}; i\dot{t}^{(k)}) \right)\right] \\
&\quad - \frac{n_{opt}}{1-a} - \frac{1}{1-a} E_{s^e \sim d_s^{(k+1)}}\left[ \left| \dot{Q}_t^{(k)}(s^e) - \dot{Q}_t^{(k)}(s^e) \right|_\infty \left| i^{(k+1)}(s^e) - i^{(k)}(s^e) \right|_1 \right],
\end{aligned}$$

where the last line makes use of Assumption 2 and the fact that $|i^{(k+1)}(s)|_1 = |i^{(k)}(s)|_1 = 1$.

Following the discussion in Lemma 2.1, we can see that $i\dot{t}^{(k)}(s) = c_s^{(k)} \nabla h_s(\dot{e}^{(k)}(s))$ with some constant $c_s^{(k)}$ for all $k$. This together with the convexity of $h_s$ (see (2.13)) guarantees that

$$D_{h_s}(i^{(k)}, \dot{e}^{(k+1)}; i\dot{t}^{(k+1)}) \ge 0 \quad \text{and} \quad D_{h_s}(i^{(k+1)}, i\dot{e}^{(k)}; i\dot{t}^{(k)}) \ge 0 \tag{B.7}$$

for any $s \in S$, thus implying that the first term of (B.6) is non-negative. It remains to control the second term in (B.6). Towards this, a little algebra gives

$$\begin{aligned}
&D_{h_s}(i^{(k+1)}, i^{(k)}; i\dot{t}^{(k)}) - D_{h_s}(i^{(k+1)}, i\dot{e}^{(k)}; i\dot{t}^{(k)}) \\
&= -\{ h_s(i^{(k)}(s)) - h_s(\dot{e}^{(k)}(s)) - \langle i\dot{t}^{(k)}(s), i^{(k)}(s) - \dot{e}^{(k)}(s)\rangle_e \} \\
&= -h_s(i^{(k)}(s)) + h_s(\dot{e}^{(k)}(s)) \\
&\quad + \left\langle \frac{1}{1+au} i\dot{t}^{(k-1)}(s) + \frac{\tau}{1+au}\dot{Q}_t^{(k-1)}(s), i^{(k)}(s) - \dot{e}^{(k)}(s) \right\rangle_e \\
&= \frac{a}{1+au}\left\{ \langle \dot{Q}_t^{(k-1)}(s), \dot{e}^{(k)}(s)\rangle_e + u h_s(\dot{e}^{(k)}(s)) + \frac{1}{a} D_{h_s}(\dot{e}^{(k)}, i^{(k-1)}; i\dot{t}^{(k-1)}) \right. \\
&\quad \left. - \left[ \langle \dot{Q}_t^{(k-1)}(s), i^{(k)}(s)\rangle_e + u h_s(i^{(k)}(s)) + \frac{1}{a} D_{h_s}(i^{(k)}, i^{(k-1)}; i\dot{t}^{(k-1)}) \right] \right\} \\
&\le \frac{an_{opt}}{1+au}.
\end{aligned} \tag{B.8}$$

Here, the first and the third lines follow from the definition (2.13) and the second inequality comes from the construction (3.7), whereas the last step invokes the definition of the oracle (3.5). Substitution of (B.7) and (B.8) into (B.6) gives

$$\begin{aligned}
(\text{B.9}) \qquad & V_u^{(k+1)}(s) - V_u^{(k)}(s) \\
& \le \frac{1+a}{1-a}\varepsilon_{\text{opt}}, \\
(\text{B.10}) \qquad & \le \frac{1}{1-a}\mathop{\mathbb{E}}_{s^e \sim d_s^{(k+1)}}\left[\left|Q_u^{(k)}(s^e) - \widehat{Q}_u^{(k)}(s^e)\right|_\infty \left|\pi^{(k+1)}(s^e) - \pi^{(k)}(s^e)\right|_1\right] \\
& \le \frac{1+a}{1-a}\varepsilon_{\text{opt}} + \frac{2}{1-a}\varepsilon_{\text{eval}}.
\end{aligned}$$

The case when $h_s$ is strongly convex. When $h_{s^e}$ is 1-strongly convex w.r.t. the $l_1$ norm, the objective function of subproblem (B.1) is $\frac{1+a\tau}{\tau}$-strongly convex w.r.t. the $l_1$ norm. Taking this together with the $\varepsilon_{\text{opt}}$-approximation guarantee in Assumption 3, we can demonstrate that

$$(\text{B.11}) \qquad \frac{1+a\tau}{2\tau}\left|\widehat{\pi}^{(k+1)}(s^e) - \pi^{(k+1)}(s^e)\right|_1^2 \le \varepsilon_{\text{opt}} \qquad \text{for all } k \ge 0 \text{ and } s^e \in S.$$

Additionally, the strong convexity assumption also implies that

$$\begin{aligned}
& D_{h_{s^e}}\left(\pi^{(k+1)}(s^e) - \widehat{\pi}^{(k)}(s^e); \pi^{(k)}(s^e)\right) \ge \frac{1}{2}\left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k+1)}(s^e)\right|_1^2 \\
& = \frac{1}{2}\left(\left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k+1)}(s^e)\right|_1^2 + \left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k)}(s^e)\right|_1^2\right) - \frac{1}{2}\left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k)}(s^e)\right|_1^2 \\
& \ge \frac{1}{4}\left(\left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k+1)}(s^e)\right|_1 + \left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k)}(s^e)\right|_1\right)^2 - \frac{1}{2}\left|\widehat{\pi}^{(k)}(s^e) - \pi^{(k)}(s^e)\right|_1^2 \\
& \ge \frac{1}{4}\left|\pi^{(k)}(s^e) - \pi^{(k+1)}(s^e)\right|_1^2 - \frac{a\varepsilon_{\text{opt}}}{1+a\tau},
\end{aligned}$$

where the third line results from Young's inequality, and the final step follows from (B.11). We can develop a similar lower bound on $D_{h_{s^e}}(\pi^{(k)}, \widehat{\pi}^{(k+1)}; \pi^{(k+1)})$ as well. Taken together, these lower bounds give

$$\begin{aligned}
& \frac{1}{\tau}\left[(1+a\tau)D_{h_{s^e}}\left(\pi^{(k)}(s^e), \widehat{\pi}^{(k+1)}(s^e); \pi^{(k+1)}(s^e)\right) + D_{h_{s^e}}\left(\pi^{(k+1)}(s^e), \widehat{\pi}^{(k)}(s^e); \pi^{(k)}(s^e)\right)\right] \\
& \ge \frac{2+a\tau}{a}\left(\frac{1}{4}\left|\pi^{(k)}(s^e) - \pi^{(k+1)}(s^e)\right|_1^2 - \frac{a\varepsilon_{\text{opt}}}{1+a\tau}\right) \\
& \ge \frac{2+a\tau}{4a}\left|\pi^{(k)}(s^e) - \pi^{(k+1)}(s^e)\right|_1^2 - 2\varepsilon_{\text{opt}}.
\end{aligned}$$

In addition, it is easily seen that

$$\begin{aligned}
& -\left|Q_u^{(k)}(s^e) - \widehat{Q}_u^{(k)}(s^e)\right|_\infty \left|\pi^{(k+1)}(s^e) - \pi^{(k)}(s^e)\right|_1 \\
& \le \left(\frac{1}{2}\frac{2a}{2+a\tau}\left|Q_u^{(k)}(s^e) - \widehat{Q}_u^{(k)}(s^e)\right|_\infty^2 + \frac{2+a\tau}{2a\tau}\left|\pi^{(k+1)}(s^e) - \pi^{(k)}(s^e)\right|_1^2\right) \\
& \le \frac{\tau}{2+a\tau}\varepsilon_{\text{eval}}^2 + \frac{2+a\tau}{4a}\left|\pi^{(k+1)}(s^e) - \pi^{(k)}(s^e)\right|_1^2
\end{aligned}$$

Combining the above two inequalities with (B.10), we arrive at the advertised bound

$$V_u^{(k+1)}(s) - V_u^{(k)}(s) \le \frac{3+a}{1-a}\varepsilon_{\text{opt}} + \frac{\tau}{(2+a\tau)(1-a)}\varepsilon_{\text{eval}}^2.$$

B.2. Step 2: Connecting the algorithm dynamic with a linear system. Now we are ready to discuss how to control $|Q_t^r - Q_t^{(k)}|_\infty$. In short, we intend to establish the connection among several intertwined quantities and identify a simple linear system that captures the algorithm dynamic.

Bounding $|Q_t^r - \mathrm{unit}^{(k+1)}|_\infty$. From the definition of $\mathrm{unit}^{(k+1)}$ in (3.7), we have

$$\begin{aligned}
&\left|Q_t^r - \mathrm{unit}^{(k+1)}\right|_\infty \\
&= \left|\alpha(Q_u^r - \mathrm{unit}^{(k)}) + (1-\alpha)(Q_u^r - Q_u^{(k)}) + (1-\alpha)(Q_u^{(k)} - Q_u^{(k)})\right|_\infty \\
&\le \alpha\left|Q_t^r - \mathrm{unit}^{(k)}\right|_\infty + (1-\alpha)\left|Q_u^r - Q_u^{(k)}\right|_\infty + (1-\alpha)\left|Q_u^{(k)} - Q_u^{(k)}\right|_\infty \\
&\le \alpha\left|Q_t^r - \mathrm{unit}^{(k)}\right|_\infty + (1-\alpha)\left|Q_u^r - Q_t^{(k)}\right|_\infty + (1-\alpha)\eta_{\mathrm{eval}}, \qquad \text{(B.12)}
\end{aligned}$$

where the last inequality is a consequence of Assumption 2.

Bounding $-\min_{s,a}(Q_t^{(k+1)}(s,a) - \mathrm{unit}^{(k+1)}(s,a))$. Applying the definition in (3.7) once again, we obtain

$$\begin{aligned}
&-\left(Q_t^{(k+1)}(s,a) - \mathrm{unit}^{(k+1)}(s,a)\right) \\
&= -\alpha\left(Q_t^{(k)}(s,a) - \mathrm{unit}^{(k)}(s,a)\right) + (1-\alpha)\left(Q_t^{(k)}(s,a) - Q_t^{(k)}(s,a)\right) \\
&\quad + \left(Q_t^{(k)}(s,a) - Q_t^{(k+1)}(s,a)\right) \\
&\le -\alpha\left(Q_t^{(k)}(s,a) - \mathrm{unit}^{(k)}(s,a)\right) + (1-\alpha+c_1)\eta_{\mathrm{eval}} + c_2\eta_{\mathrm{opt}}, \qquad \text{(B.13)}
\end{aligned}$$

where

$$c_1 = \begin{cases} \dfrac{2\alpha}{1-\alpha} & \text{if } h_s \text{ is convex but not strongly convex,} \\ \dfrac{\alpha\eta_{\mathrm{eval}}\alpha}{(2+\alpha u)(1-\alpha)} & \text{if } h_s \text{ is 1-strongly convex w.r.t. the } l_1 \text{ norm,} \end{cases} \qquad \text{(B.14)}$$

$$c_2 = \begin{cases} \dfrac{(\alpha+1)\alpha}{1-\alpha} & \text{if } h_s \text{ is convex but not strongly convex,} \\ \dfrac{(\alpha+3)\alpha}{1-\alpha} & \text{if } h_s \text{ is 1-strongly convex w.r.t. the } l_1 \text{ norm.} \end{cases}$$

Here, the last step of (B.13) follows from Assumption 2 as well as the following relation:

$$Q_t^{(k)}(s,a) - Q_t^{(k+1)}(s,a) = \alpha \mathop{\mathbb{E}}_{s^e \sim P(t|s,a)}\left[V_t^{(k)}(s^e) - V_t^{(k+1)}(s^e)\right] \le c_1\eta_{\mathrm{eval}} + c_2\eta_{\mathrm{opt}},$$

where we have made use of Lemma B.1. Taking the maximum over $(s,a)$ on both sides of (B.13) yields

$$\begin{aligned}
&-\min_{s,a}\left(Q_t^{(k+1)}(s,a) - \mathrm{unit}^{(k+1)}(s,a)\right) \\
&\le -\alpha\min_{s,a}\left(Q_t^{(k)}(s,a) - \mathrm{unit}^{(k)}(s,a)\right) + (1-\alpha+c_1)\eta_{\mathrm{eval}} + c_2\eta_{\mathrm{opt}}. \qquad \text{(B.15)}
\end{aligned}$$

Bounding $|Q_u^r - Q_t^{(k+1)}|_\infty$. To begin with, let us decompose $Q_u^r(s,a) - Q_t^{(k+1)}(s,a)$ into several parts. Invoking the relation (4.5) in Lemma 4.2 as well as the property (2.8), we reach

(B.16)

$$
\begin{aligned}
&Q_u^r(s,a) - Q_\iota^{(k+1)}(s,a) \\
&= T_{u,h}(Q_u^r)(s,a) - \Big[ r(s,a) + a \mathop{E}_{s^e \sim P(t|s,a)} \big[ e_{u\dot{t}^{(k+1)}(s^e), \dot{e}^{(k+1)}(s^e)} - u h_{s^e}(\dot{e}^{(k+1)}(s)) \big] \Big] \\
&= T_{u,h}(Q_u^r)(s,a) - \Big[ r(s,a) + a \mathop{E}_{s^e \sim P(t|s,a)} \big[ e_{u\dot{t}^{(k+1)}(s^e), i^{(k+1)}(s^e)} - u h_{s^e}(i^{(k+1)}(s)) \big] \Big] \\
&\quad - a \mathop{E}_{s^e \sim P(t|s,a), a^e \sim m_{i^{(k+1)}(s^e)}} \big[ Q_\iota^{(k+1)}(s^e,a^e) - u\dot{t}^{(k+1)}(s^e,a^e) \big] \\
&= T_{u,h}(Q_u^r)(s,a) \\
&\quad - \Big[ r(s,a) + a \mathop{E}_{s^e \sim P(t|s,a)} \big[ e_{u\dot{t}^{(k+1)}(s^e), \dot{e}^{(k+1)}(s^e)} - u h_{s^e}(\dot{e}^{(k+1)}(s)) \big] \Big] \\
&\quad - u a \mathop{E}_{s^e \sim P(t|s,a)} \big[ e_{\dot{t}^{(k+1)}(s^e), i^{(k+1)}(s^e)} - \dot{e}^{(k+1)}(s^e) \big] \\
&\quad - h_{s^e}(i^{(k+1)}(s)) + h_{s^e}(\dot{e}^{(k+1)}(s)) \\
&\quad - a \mathop{E}_{s^e \sim P(t|s,a), a^e \sim m_{i^{(k+1)}(s^e)}} \big[ Q_\iota^{(k+1)}(s^e,a^e) - u\dot{t}^{(k+1)}(s^e,a^e) \big].
\end{aligned}
$$

In wat follows, we control the three terms in (B.16) separately.

- To begin with, we repeat an argument similar to that for (A.9) to show that

$$
\begin{aligned}
&T_{u,h}(Q_u^r)(s,a) \\
&\quad - \Big[ r(s,a) + a \mathop{E}_{s^e \sim P(t|s,a)} \big[ e_{u\dot{t}^{(k+1)}(s^e), \dot{e}^{(k+1)}(s^e)} - u h_{s^e}(\dot{e}^{(k+1)}(s)) \big] \Big] \\
&= T_{u,h}(Q_u^r)(s,a) - T_{u,h}(\dot{t}^{(k+1)})(s,a) \\
&\quad q\, a \big| Q_\iota^r - u\dot{t}^{(k+1)} \big|_y .
\end{aligned}
$$

- The second term of (B.16) can be bounded by applying (B.8) with k replaced by k + 1:

$$
\begin{aligned}
&e_{\dot{t}^{(k+1)}(s^e), i^{(k+1)}(s^e)} - \dot{e}^{(k+1)}(s^e) \\
&\quad - h_{s^e}(i^{(k+1)}(s)) + h_{s^e}(\dot{e}^{(k+1)}(s)) \quad q\, \frac{a r_{opt}}{1 + au}.
\end{aligned}
$$

- As for the third term of (B.16), taking the maximum over all $(s,a) \in S \times A$ gives

$$
Q_u^{(k+1)}(s^e,a^e) - u\dot{t}^{(k+1)}(s^e,a^e) \quad q \quad \min_{s,a} \big( Q_u^{(k+1)}(s,a) - u\dot{t}^{(k+1)}(s,a) \big).
$$

Taken together, the above bounds and the decomposition (B.16) lead to

$$
\begin{aligned}
&\big| Q_u^t - Q_\iota^{(k+1)} \big|_y \\
&\quad q\, a \big| Q_u^r - u\dot{t}^{(k+1)} \big|_y - a \min_{s,a} \big( Q_u^{(k+1)}(s,a) - u\dot{t}^{(k+1)}(s,a) \big) + a(1-a) r_{opt}.
\end{aligned} \tag{B.17}
$$

A linear system of interest. Combining (B.12), (B.15), and (B.17), we reach the following linear system:

$$z_{k+1} \le B z_k + b, \tag{B.18}$$

where

$$B := \begin{bmatrix} 1-\alpha & \alpha & \alpha \\ \alpha(1-\alpha) & \alpha & \alpha \\ 0 & 0 & \alpha \end{bmatrix}, \qquad z_k := \begin{bmatrix} \|Q_t^r - Q_t^{(k)}\|_\infty \\ \|Q^t - u_t^{(k)}\|_\infty \\ -\min_{s,a} Q_t^{(k)}(s,a) - u_t^{(k)}(s,a) \end{bmatrix},$$

$$b := \begin{bmatrix} \alpha(2-2\alpha+c_1)n_{\mathrm{eval}} + \alpha(1-\alpha+c_2)n_{\mathrm{opt}} \\ (1-\alpha)n_{\mathrm{eval}} \\ (1-\alpha+c_1)n_{\mathrm{eval}} + c_2 n_{\mathrm{opt}} \end{bmatrix}. \tag{B.19}$$

This linear system of three variables captures how the estimation error progresses as the iteration count increases.

B.3. Step 3: Linear system analysis. In this step, we analyze the behavior of the linear system (B.18) derived above. Observe that the eigenvalues and respective eigenvectors of the matrix $B$ are given by

$$\alpha_1 = \alpha + (1-\alpha)\alpha, \quad \alpha_2 = \alpha, \quad \alpha_3 = 0, \tag{B.20}$$

$$v_1 = \begin{bmatrix} 1 \\ \alpha \\ 0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \quad v_3 = \begin{bmatrix} \alpha - 1 \\ \alpha \\ 0 \end{bmatrix}. \tag{B.21}$$

Armed with these, we can decompose $z_0$ in terms of the eigenvectors of $B$ as follows:

$$\begin{aligned} z_0 &\le \begin{bmatrix} \|Q_t^r - Q_t^{(0)}\|_\infty \\ \|Q_t^r - u_t^{(0)}\|_\infty \\ \|Q_t^{(0)} - u_t^{(0)}\|_\infty \end{bmatrix} \\ &= \frac{1}{\alpha + (1-\alpha)\alpha}\Big[(1-\alpha)\|Q_t^r - Q_u^{(0)}\|_\infty + \alpha\|Q_u^r - u_t^{(0)}\|_\infty + \alpha\|Q_t^{(0)} - u_t^{(0)}\|_\infty\Big] v_1 \\ &\quad + \|Q_t^r - u_t^{(0)}\|_\infty v_2 + e_z v_3 \\ &\le \frac{1}{\alpha + (1-\alpha)\alpha}\Big[\|Q_t^r - Q_t^{(0)}\|_\infty + 2\alpha\|Q_t^r - u_t^{(0)}\|_\infty\Big] v_1 \\ &\quad + \|Q^t - u_t^{(0)}\|_\infty v_2 + e_z v_3, \end{aligned} \tag{B.22}$$

where $e_z \in \mathbb{R}$ is some constant that does not affect our final result. Also, the vector $b$ defined in (B.19) satisfies

$$\begin{aligned} b &\le \begin{bmatrix} \alpha(2-2\alpha+c_1)n_{\mathrm{eval}} + \alpha(1-\alpha+c_2)n_{\mathrm{opt}} \\ (1-\alpha)n_{\mathrm{eval}} + (1-\alpha)n_{\mathrm{opt}} \\ (1-\alpha+c_1)n_{\mathrm{eval}} + c_2 n_{\mathrm{opt}} \end{bmatrix} \\ &= \begin{bmatrix} (2-2\alpha+c_1)n_{\mathrm{eval}} + (1-\alpha+c_2)n_{\mathrm{opt}} \end{bmatrix} v_1 + \begin{bmatrix} (1-\alpha+c_1)n_{\mathrm{eval}} + c_2 n_{\mathrm{opt}} \end{bmatrix} v_2. \end{aligned} \tag{B.23}$$

Using the decomposition in (B.22) and (B.23) and applying the system relation (B.18) recursively, we can derive

$$
\begin{aligned}
z_{k+1} &\le \beta^{k+1} z_0 + \sum_{t=0}^{k} \beta^{k-t} b \\
&\le \beta^{k+1} \Big[ \frac{1}{\alpha + (1-\alpha)\alpha} \Big[ \big| Q_u^r - Q_u^{(0)} \big|_y + 2\alpha \big| Q_u^r - u i^{(0)} \big|_y \Big] v_1 + \big| Q_u^r - u i^{(0)} \big|_y v_2 + e_z v_3 \Big] \\
&\quad + \sum_{t=0}^{k} \beta^{k-t} \Big[ \big[ (2 - 2\alpha + c_1) n_{eval} + (1 - \alpha + c_2) n_{opt} \big] v_1 + \big[ (1 - \alpha + c_1) n_{eval} + c_2 n_{opt} \big] v_2 \Big] \\
&= \alpha_1^{k+1} \Big( \big| Q_u^r - Q_u^{(0)} \big|_y + 2\alpha \big| Q_u^r - u i^{(0)} \big|_y \Big) \\
&\quad + \frac{1 - \alpha_1^{k+1}}{1 - \alpha_1} \big[ (2 - 2\alpha + c_1) n_{eval} + (1 - \alpha + c_2) n_{opt} \big] v_1 \\
&\quad + \alpha^{k+1} \big| Q_u^r - u i^{(0)} \big|_y + \frac{1 - \alpha^{k+1}}{1 - \alpha} \big[ (1 - \alpha + c_1) n_{eval} + c_2 n_{opt} \big] v_2 .
\end{aligned}
$$

Recognizing that the first two entries of $v_2$ are nonpositive, we can discard the term involving $v_2$ and obtain

$$
\begin{bmatrix} |Q_u^r - Q_u^{(k+1)}|_y \\ |Q_u^r - u i^{(k+1)}|_y \end{bmatrix} \le \alpha_1^{k+1} \Big( \big| Q_u^r - Q_u^{(0)} \big|_y + 2\alpha \big| Q_u^r - u i^{(0)} \big|_y \Big) + C \begin{bmatrix} 1 \\ 1 \end{bmatrix},
$$

where

$$
C := \frac{\alpha}{1 - \alpha_1} \big[ (2 - 2\alpha + c_1) n_{eval} + (1 - \alpha + c_2) n_{opt} \big].
$$

Making use of the fact that $1 - \alpha_1 = (1 - \alpha)(1 - \alpha)$, we can conclude that

$$
C = \frac{1}{1 - \alpha} \Big[ \Big( 2 + \frac{c_1}{1 - \alpha} \Big) n_{eval} + \Big( 1 + \frac{c_2}{1 - \alpha} \Big) n_{opt} \Big].
$$

The above bound essentially says that

$$
\big| Q_u^r - Q_u^{(k+1)} \big|_y \le \alpha \Big[ \alpha_1^k \Big( \big| Q_u^r - Q_u^{(0)} \big|_y + 2\alpha \big| Q_u^r - u i^{(0)} \big|_y \Big) + C \Big]
$$

and

$$
\big| Q_u^r - u i^{(k+1)} \big|_y \le \alpha_1^{k+1} \Big( \big| Q_u^r - Q_u^{(0)} \big|_y + 2\alpha \big| Q_u^r - u i^{(0)} \big|_y \Big) + C.
$$

Turning to $V_u^r(s) - V_u^{(k+1)}(s)$, by an argument similar to that of (4.13), we have

$$
\begin{aligned}
&V_u^r(s) - V_u^{(k+1)}(s) \\
&= \Big[ Q_u^r(s) - Q_u^{(k+1)}(s), i^{(k+1)}(s) \Big] \\
&\quad + \big[ u(h_s(i^{(k+1)}(s))) - h_s(i_u^r(s))) - \big\langle Q_u^r(s), i^{(k+1)}(s) - i_u^r(s) \big\rangle \big] \\
&= \Big\langle Q_u^r(s) - Q_u^{(k+1)}(s), i^{(k+1)}(s) \Big\rangle + u D_{h_s}(i^{(k+1)}, i_u^r; g_u^r) \\
&\le \big| Q_u^r - Q_u^{(k+1)} \big|_y + u D_{h_s}(i^{(k+1)}, \lambda^{\sim(k+1)}; i^{(k+1)}) \\
&\quad + u D_{h_s}(\lambda^{\sim(k+1)}, i_u^r; g_u^r) + u \big\langle i^{(k+1)}(s) - \lambda^{\sim(k+1)}(s), i^{(k+1)}(s) - g_u^r(s) \big\rangle,
\end{aligned}
$$

where the third step results from the standard three-point lemma. To control the second term, we rearrange terms in (B.4) and reach at

$$\begin{aligned}
n_{\text{opt}}\, q\, & -\Big\langle \tilde{Q}_t^{(k)}(s), \theta^{(k+1)}(s)\Big\rangle + u h_s\big(\theta^{(k+1)}(s)\big) + \frac{1}{a} D_{h_s}\big(\theta^{(k+1)}, \theta^{(k)}; \theta_t^{(k)}\big) \\
& + \Big\langle \tilde{Q}_t^{(k)}(s), \hat{\theta}^{(k+1)}(s)\Big\rangle - u h_s\big(\hat{\theta}^{(k+1)}(s)\big) - \frac{1}{a} D_{h_s}\big(\hat{\theta}^{(k+1)}, \theta^{(k)}; \theta_t^{(k)}\big) \\
= & \Big\langle \tilde{Q}_t^{(k)}(s), \hat{\theta}^{(k+1)}(s) - \theta^{(k+1)}(s)\Big\rangle + \frac{1+au}{a} h_s\big(\theta^{(k+1)}(s)\big) - h_s\big(\tilde{\lambda}^{(k+1)}(s)\big) \\
& + \frac{1}{a}\Big\langle \theta_t^{(k)}(s), \hat{\theta}^{(k+1)}(s) - \theta^{(k+1)}(s)\Big\rangle \\
= & \frac{1+au}{a} D_{h_s}\big(\theta^{(k+1)}, \tilde{\lambda}^{(k+1)}; \theta_t^{(k+1)}\big).
\end{aligned}$$

For the remaining terms, recall that $\theta_t^{(k+1)} = c_s^{(k+1)} \nabla h_s(\hat{\theta}^{(k+1)}(s))$ with some constant $c_s^{(k+1)}$. So we have

$$\begin{aligned}
& u D_{h_s}\big(\tilde{\lambda}^{(k+1)}, \theta_u^r; g_t^r\big) + u\Big\langle \theta^{(k+1)}(s) - \tilde{\lambda}^{(k+1)}(s), \theta_t^{(k+1)}(s) - g_t^r(s)\Big\rangle \\
& \quad = u h_s\big(\tilde{\lambda}^{(k+1)}(s)\big) - u h_s\big(\theta_t^r(s)\big) - \Big\langle \tilde{\lambda}^{(k+1)}(s) - \theta_t^r(s), Q_t^r(s)\Big\rangle \\
& \qquad + u\Big\langle \theta^{(k+1)}(s) - \tilde{\lambda}^{(k+1)}(s), \theta^{(k+1)}(s) - g_t^r(s)\Big\rangle \\
& \quad q \Big\langle \theta_u^r(s) - \tilde{\lambda}^{(k+1)}(s), Q_t^r(s) - u\theta^{(k+1)}\Big\rangle \\
& \qquad - \Big\langle \theta^{(k+1)}(s) - \tilde{\lambda}^{(k+1)}(s), Q_t^r(s) - u\theta_t^{(k+1)}(s)\Big\rangle \\
& \quad = \Big\langle \theta_u^r(s) - \theta^{(k+1)}(s), Q_u^r(s) - u\theta^{(k+1)}\Big\rangle \\
& \quad q \frac{2}{u}\big\| Q_u(s) - u\theta^{(k+1)}(s)\big\|_*^2.
\end{aligned}$$

Taken together, we conclude that

$$\begin{aligned}
& V_t^r(s) - V_t^{(k+1)}(s) \\
& \quad q \big\| Q^r - Q^{(k+1)}\big\|_* + 2\big\| Q_u^r(s) - u\theta^{(k+1)}(s)\big\|_* + \frac{au}{1+au} r_{\text{opt}} \\
& \quad q (a+2)\Big[a_1^k\Big(\big\| Q_u^r - Q_u^{(0)}\big\|_* + 2a\big\| Q_u^r - u\theta^{(0)}\big\|_* + C\Big)\Big] + \frac{au}{1+au} n_{\text{opt}}.
\end{aligned}$$

Finally, plugging in the choices of $c_1$ and $c_2$ (cf. (B.14)), we have $C \le C_2$ when $\{h_s\}$ is convex and $C \le C_3$ when $\{h_s\}$ is 1-strongly convex w.r.t. the $l_1$ norm. In addition, for the latter case, we can follow an argument similar to that of (4.14) to demonstrate that

$$\begin{aligned}
& \big\| \theta_u^r(s) - \hat{\theta}_t^{(k)}(s)\big\|_1 \\
& \quad q u^{-1}\Big((1-a)a + a^2\Big)^k\Big(\big\| Q_u^r - Q_u^{(0)}\big\|_* + 2a\big\| Q_u^r - u\theta^{(0)}\big\|_*\Big) + u^{-1} C_3,
\end{aligned}$$

which taken together with (B.11) gives

$$\begin{aligned}
\big\| \theta_u^r(s) - \theta_u^{(k)}(s)\big\|_1 q & \big\| \theta_u^r(s) - \hat{\theta}_u^{(k)}(s)\big\|_1 + \big\| \theta_u^{(k)}(s) - \hat{\theta}_u^{(k)}(s)\big\|_1 \\
& q u^{-1}\Big((1-a)a + a^2\Big)^k\Big(\big\| Q_u^r - Q_u^{(0)}\big\|_* + 2a\big\| Q_u^r - u\theta^{(0)}\big\|_*\Big) \\
& \quad + u^{-1} C_3 + \frac{\sqrt{2a r_0 p_t}}{1+au}.
\end{aligned}$$

This concludes the proof.

## REFERENCES


[1] Y. Abbasi-Yadkori, P. Bartlett, K. Bhatia, N. Lazic, C. Szepesvari, and G. Weisz, Politex: Regret bounds for policy iteration using expert prediction, in Proceedings of the International Conference on Machine Learning, PMLR, 2019, pp. 3692--3702.

[2] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, Reinforcement Learning: Theory and Algorithms, tech. report, 2019.

[3] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, Optimality and approximation with policy gradient methods in Markov decision processes, in Proceedings of the Conference on Learning Theory, PMLR, 2020, pp. 64--66.

[4] A. Agazzi and J. Lu, Global optimality of softmax policy gradient with single hidden layer neural networks in the mean-field regime, in Proceedings of the International Conference on Learning Representations, ICLR, 2021.

[5] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, Concrete Problems in AI Safety, manuscript, 2016.

[6] A. Beck, First-Order Methods in Optimization, SIAM, Philadelphia, 2017, https://doi.org/10.1137/1.9781611974997.

[7] A. Beck and M. Teboulle, Mirror descent and nonlinear projected subgradient methods for convex optimization, Oper. Res. Lett., 31 (2003), pp. 167--175.

[8] D. P. Bertsekas, Dynamic Programming and Optimal Control, 4th ed., Athena Scientific, Belmont, MA, 2017.

[9] J. Bhandari and D. Russo, Global Optimality Guarantees for Policy Gradient Methods, preprint, https://arxiv.org/abs/1906.01786, 2019.

[10] J. Bhandari and D. Russo, A Note on the Linear Convergence of Policy Gradient Methods, preprint, https://arxiv.org/abs/2007.11120v1, 2020.

[11] S. Cen, F. Chen, and Y. Chi, Independent Natural Policy Gradient Methods for Potential Games: Finite-Time Global Convergence with Entropy Regularization, preprint, https://arxiv.org/abs/2204.05466, 2022.

[12] S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi, Fast global convergence of natural policy gradient methods with entropy regularization, Oper. Res., 70 (2022), pp. 2563--2578.

[13] S. Cen, Y. Chi, S. S. Du, and L. Xiao, Faster Last-Iterate Convergence of Policy Optimization in Zero-Sum Markov Games, preprint, https://arxiv.org/abs/2210.01050, 2022.

[14] S. Cen, Y. Wei, and Y. Chi, Fast policy extragradient methods for competitive games with entropy regularization, in Proceedings of the 35th Conference on Neural Information Processing Systems, 2022, pp. 27952--27964.

[15] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, A Lyapunov-based approach to safe reinforcement learning, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2019, pp. 8103--8112.

[16] Y. Chow, O. Nachum, and M. Ghavamzadeh, Path consistency learning in Tsallis entropy regularized MDPs, in Proceedings of the International Conference on Machine Learning, PMLR, 2018, pp. 979--988.

[17] B. Dai, A. Shaw, L. Li, L. Xiao, N. He, Z. Liu, J. Chen, and L. Song, SBEED: Conver-gent reinforcement learning with nonlinear function approximation, in Proceedings of the International Conference on Machine Learning, PMLR, 2018, pp. 1125--1134.

[18] D. Ding, X. Wei, Z. Yang, Z. Wang, and M. Jovanovic, Provably eficient safe exploration via primal-dual policy optimization, in Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 3304--3312.

[19] Y. Efroni, S. Mannor, and M. Pirotta, Exploration-Exploitation in Constrained MDPs, preprint, https://arxiv.org/abs/2003.02189, 2020.

[20] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, Global convergence of policy gradient meth-ods for the linear quadratic regulator, in Proceedings of the International Conference on Machine Learning, 2018, pp. 1467--1476.

[21] M. Geist, B. Scherrer, and O. Pietquin, A theory of regularized Markov decision processes, in Proceedings of the International Conference on Machine Learning, 2019, pp. 2160--2169.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in Proceedings of the International Conference on Machine Learning, PMLR, 2018, pp. 1861--1870.

[23] B. Hao, N. Lazic, Y. Abbasi-Yadkori, P. Joulani, and C. Szepesvári, Adaptive approximate policy iteration, in Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 523--531.

[24] S. M. Kakade, A natural policy gradient, in Proceedings of the 14th International Conference on Neural Information Processing Systems, 2001, pp. 1531--1538.

[25] S. Khodadadian, P. R. Jhunjhunwala, S. M. Varma, and S. T. Maguluri, On the Linear Convergence of Natural Policy Gradient Algorithm, preprint, https://arxiv.org/abs/2105.01424, 2021.
[26] K. C. Kiwiel, Proximal minimization methods with generalized Bregman functions, SIAM J. Control Optim., 35 (1997), pp. 1142--1168, https://doi.org/10.1137/S0363012995281742.
[27] G. Lan, Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes, Math. Program., 198 (2023), pp. 1059--1106.
[28] G. Lan, Z. Lu, and R. D. Monteiro, Primal-dual first-order methods with O(1/n ) iteration-complexity for cone programming, Math. Program., 126 (2011), pp. 1--29.
[29] G. Lan and Y. Zhou, An optimal randomized incremental gradient method, Math. Program., 171 (2018), pp. 167--215.
[30] N. Lazic, D. Yin, Y. Abbasi-Yadkori, and C. Szepesvari, Improved regret bound and experience replay in regularized policy iteration, in Proceedings of the International Conference on Machine Learning, PMLR, 2021, pp. 6032--6042.
[31] K. Lee, S. Choi, and S. Oh, Sparse Markov decision processes with causal sparse Tsallis entropy regularization for reinforcement learning, IEEE Robot. Autom. Lett., 3 (2018), pp. 1466--1473.
[32] K. Lee, S. Kim, S. Lim, S. Choi, and S. Oh, Tsallis Reinforcement Learning: A Unified Framework for Maximum Entropy Reinforcement Learning, preprint, https://arxiv.org/abs/1902.00137, 2019.
[33] G. Li, Y. Wei, Y. Chi, and Y. Chen, Softmax policy gradient methods can take exponential time to converge, Math. Program., to appear.
[34] B. Liu, Q. Cai, Z. Yang, and Z. Wang, Neural trust region/proximal policy optimization attains globally optimal policy, in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2020, pp. 10565--10576.
[35] Y. Liu, K. Zhang, T. Basar, and W. Yin, An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2021.
[36] J. Mei, Y. Gao, B. Dai, C. Szepesvari, and D. Schuurmans, Leveraging non-uniformity in first-order non-convex optimization, in Proceedings of the International Conference on Machine Learning, PMLR, 2021, pp. 7555--7564.
[37] J. Mei, C. Xiao, B. Dai, L. Li, C. Szepesvári, and D. Schuurmans, Escaping the gravitational pull of softmax, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2021.
[38] J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans, On the global convergence rates of softmax policy gradient methods, in Proceedings of the International Conference on Machine Learning, PMLR, 2020, pp. 6820--6829.
[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, Human-level control through deep reinforcement learning, Nature, 518 (2015), pp. 529--533.
[40] T. M. Moldovan and P. Abbeel, Safe Exploration in Markov Decision Processes, manuscript, 2012.
[41] A. S. Nemirovsky and D. B. Yudin, Problem Complexity and Method Eficiency in Optimization, manuscript, 1983.
[42] G. Neu, A. Jonsson, and V. Gómez, A Unified View of Entropy-Regularized Markov Decision Processes, preprint, https://arxiv.org/abs/1705.07798, 2017.
[43] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley \& Sons, New York, 2014.
[44] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, Trust region policy optimization, in Proceedings of the International Conference on Machine Learning, PMLR, 2015, pp. 1889--1897.
[45] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal Policy Optimization Algorithms, preprint, https://arxiv.org/abs/1707.06347, 2017.
[46] L. Shani, Y. Efroni, and S. Mannor, Adaptive Trust Region Policy Optimization: Global Convergence and Faster Rates for Regularized MDPs, preprint, https://arxiv.org/abs/1909.02769, 2019.
[47] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in Proceedings of the 12th International Conference on Neural Information Processing Systems, 1999, pp. 1057--1063.

[48] M. Tomar, L. Shani, Y. Efroni, and M. Ghavamzadeh, Mirror Descent Policy Optimization, preprint, https://arxiv.org/abs/2005.09814, 2020.
[49] C. Tsallis, Possible generalization of Boltzmann-Gibbs statistics, J. Statist. Phys., 52 (1988), pp. 479--487.
[50] N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, and M. Geist, Leverage the average: An analysis of KL regularization in reinforcement learning, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2021.
[51] L. Wang, Q. Cai, Z. Yang, and Z. Wang, Neural policy gradient methods: Global optimal-ity and rates of convergence, in Proceedings of the International Conference on Learning Representations, 2019.
[52] W. Wang, J. Han, Z. Yang, and Z. Wang, Global convergence of policy gradient for linear-quadratic mean-field control/game in continuous time, in Proceedings of the International Conference on Machine Learning, PMLR, 2021, pp. 10772--10782.
[53] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforce-ment learning, Mach. Learn., 8 (1992), pp. 229--256.
[54] R. J. Williams and J. Peng, Function optimization using connectionist reinforcement learning algorithms, Connect. Sci., 3 (1991), pp. 241--268.
[55] L. Xiao, On the Convergence Rates of Policy Gradient Methods, preprint, https://arxiv.org/abs/2201.07443, 2022.
[56] P. Xu, F. Gao, and Q. Gu, Sample eficient policy gradient methods with recursive variance reduction, in Proceedings of the International Conference on Learning Representations, 2019.
[57] T. Xu, Y. Liang, and G. Lan, A Primal Approach to Constrained Policy Optimization: Global Optimality and Finite-Time Analysis, preprint, https://arxiv.org/abs/2011.05869v1, 2020.
[58] M. Yu, Z. Yang, M. Kolar, and Z. Wang, Convergent policy optimization for safe reinforce-ment learning, in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2020, pp. 3127--3139.
[59] 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.
[60] J. Zhang, J. Kim, B. O'Donoghue, and S. Boyd, Sample Eficient Reinforcement Learning with REINFORCE, preprint, https://arxiv.org/abs/2010.11364, 2020.
[61] J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang, Variational policy gradi-ent method for reinforcement learning with general utilities, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2021, pp. 4572--4583.
[62] J. Zhang, C. Ni, Z. Yu, C. Szepesvari, and M. Wang, On the convergence and sample efi-ciency of variance-reduced policy gradient method, in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2021, pp. 2228--2240.
[63] K. Zhang, B. Hu, and T. Basar, Policy optimization for $H_2$ linear control with $H_\infty$ robustness guarantee: Implicit regularization and global convergence, SIAM J. Control Optim., 59 (2021), pp. 4081--4109, https://doi.org/10.1137/20M1347942.
[64] Y. Zhao, Y. Tian, J. Lee, and S. Du, Provably eficient policy optimization for two-player zero-sum Markov games, in Proceedings of the International Conference on Artificial In-telligence and Statistics, PMLR, 2022, pp. 2736--2761.