Proceedings of Machine Learning Research vol 242:1046–1057, 2024 6th Annual Conference on Learning for Dynamics and Control

Finite-Time Complexity of Incremental Policy Gradient Methods for Solving Multi-Task Reinforcement Learning

Yitao Bai YITAOB@VT.EDU
Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA

Thinh T. Doan THINHDOAN@VT.EDU
Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA

Editors: A. Abate, M. Cannon, K. Margellos, A. Papachristodoulou

Abstract

We consider a multi-task learning problem, where an agent is presented a number of N reinforcement learning tasks. To solve this problem, we are interested in studying the gradient approach, which iteratively updates an estimate of the optimal policy using the gradients of the value functions. The classic policy gradient method, however, may be expensive to implement in the multi-task settings as it requires access to the gradients of all the tasks at every iteration. To circumvent this issue, in this paper we propose to study an incremental policy gradient method, where the agent only uses the gradient of only one task at each iteration. Our main contribution is to provide theoretical results to characterize the performance of the proposed method. In particular, we show that incremental policy gradient methods converge to the optimal value of the multi-task reinforcement learning objectives at a sublinear rate $\mathcal{O}(1/\sqrt{k})$, where k is the number of iterations. To illustrate its performance, we apply the proposed method to solve a simple multi-task variant of GridWorld problems, where an agent seeks to find an policy to navigate effectively in different environments.

Keywords: Multi-Task Reinforcement Learning, Markov Decision Processes, Incremental Policy Gradient Methods

1 Introduction

Reinforcement learning (RL), a data-driven control approach for optimal decision making, has achieved remarkable accomplishments in tackling challenging problems in various applications, such as playing games Mnih et al. (2015); Silver et al. (2017), robotics Gu et al. (2017), and autonomous driving Kiran et al. (2021). While RL provides a powerful learning framework, it suffers a fundamental challenge in its data efficiency. The existing RL methods are known to require a significant amount of data and computational resources in their training. In addition, policies learned in one task might not be applicable to solve other tasks; performing well in a new task requires to restart the expensive training process. This challenge has limited the wide applicability of RL in solving real-world problems.

The limitation of RL has motivated the study of multi-task RL (MTRL) framework, where an agent aims to learn multiple tasks simultaneously. If the tasks are related in some ways, then learning them jointly should be more efficient than learning individually. Indeed, MTRL aims to improve generalization and efficiency by exploiting the inherent relationships between multiple tasks Caruana (1997); Khetarpal et al. (2020). Motivated by recent studies on policy gradient methods in single-task RL settings, in this paper we propose to study an incremental policy gradient method to solve MTRL problems. Unlike the classic policy gradient approach where the agent is required to access the gradients of all the task functions at every iteration, the proposed method uses the gradient of only one task per iteration in its update. Thus, the proposed incremental policy gradient method can be implemented efficiently when agent cannot access which task it is perform or agent have difficulty to accessing all the tasks at every step is challenging. For example, consider a modern smart power grid, the system must balance electricity supply and demand in real-time, managing a variety of tasks such as distributing power from renewable sources, responding to fluctuating consumer demand, and maintaining grid stability. The specific tasks and challenges faced by the grid management system change dynamically throughout the day due to factors like weather conditions, unexpected outages, and varying usage patterns. The system cannot always access detailed information about every task or situation it will encounter at each moment, and the complexity of the grid makes it challenging to have a comprehensive overview of all tasks at all times.

Contribution: This paper proposes to study an incremental policy mirror descent (IPMD) method for solving MTRL problems. Our main contribution is to provide theoretical results to characterize the performance of the proposed method. In particular, we show that incremental policy gradient methods converge to the optimal value of the multi-task reinforcement learning objectives at a sublinear rate $\mathcal{O}(1/\sqrt{k})$, where k is the number of iterations. To illustrate its performance, we apply the proposed method to solve a simple multi-task variant of GridWorld problems, where an agent seeks to find a policy to navigate effectively in different environments.

1.1 Related Works

The focus of this paper is to study an incremental variant of policy gradient methods for solving MTRL problems. Theoretical results of policy gradient methods are well understood, most policy gradient methods converge sublinearly Agarwal et al. (2021), Cen et al. (2022), Even-Dar et al. (2009), Mei et al. (2020), and Wang et al. (2019). These results were improved in Bhandari and Russo (2020), and Cen et al. (2022) where the authors use the contraction of the Bellman equation show policy gradient methods converge linearly. While Bhandari and Russo (2020) requires an exact line search in their algorithm, the result in Cen et al. (2022) is derived for entropy regularized problems. In Lan (2023), the author provides a new analysis to achieve linear convergence rate for general MDP problems. One can apply policy gradient methods to solve MTRL problems, however, they require access to the gradients of all the tasks at each iteration. This can be expensive to implement as the number of tasks can be very large. Our focus in this paper is to study an incremental policy gradient approach, where each iteration requires the gradient of only one task to update the underlying policy variable.

Another approach to solve MTRL problems is to use distributed policy gradient methods Zeng et al. (2021), Lan et al. (2023). This approach often has a network of agents, each performs one task. The agents are then collaborate either directly or indirectly through a centralized coordinator to aggregate their local solutions. Prior study in distributed policy gradient methods have investigated convergence rates under various scenarios. For instance, research by Lan et al. (2023) demonstrated $\mathcal{O}(1/k)$ convergence rates for their specific algorithm FedNPG-ADMM. Similarly, Zeng et al. (2021) explored the convergence behavior of their decentralized entropy regularized policy gradient methods, revealing $\mathcal{O}(1/\sqrt{k})$ convergence rates for certain architectures and loss functions. On the other hand, we study a different setting where there is an agent that aims to learn a number of tasks. Therefore, the theoretical results in the distributed methods are not applicable to our setting in this paper.

Incremental gradient methods have been well studied in the optimization literature, in both convex and nonconvex settings Bertsekas et al. (2011); Reddi et al. (2016). These methods are known to perform well in large-scale problems, e.g., when the number of tasks is large. In this paper, we are interested in applying this approach to solve multi-task RL problems. It is known that RL objectives are nonconvex Agarwal et al. (2020). Thus, one can apply the existing results of incremental methods for nonconvex problems to the RL setting. However, this approach will mostly result in a convergence to stationary points. Although RL objectives are nonconvex, they satisfy the so-called gradient domination, implying every stationary point is a global optimal. Indeed, recent theoretical guarantees for the convergence of policy gradient methods to the global optimal solution in the single-task RL setting are based on this property. Our focus in this paper is to exploit the gradient domination condition to study the convergence of the incremental policy gradient methods in the multi-task RL setting.

2 Multi-Task Reinforcement Learning

We consider a multi-task reinforcement learning problem, where an agent is presented N tasks, each is modeled by a discounted Markov decision process (MDP). In particular, the MDP $\mathcal{M}^i$ is a collection of 5-tuples, $\mathcal{M}^i = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}^i, \gamma)$ where $\mathcal{S}$ and $\mathcal{A}$ are the finite sets of states and actions. The transition probability kernel $\mathcal{P}$ specifies the distribution of the next state s' given the current state and action, i.e., $s' \sim \mathcal{P}(\cdot \mid s, a)$. In addition, $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function for task i and $\gamma \in (0, 1)$ is a discount factor. Here, without loss of generality, we consider the reward between $(0, 1)$, but it can be extended to any bounded interval. A policy $\pi(\cdot, s)$ is a probability distribution over the action space $\mathcal{A}$ for each different state $s \in \mathcal{S}$. Each choice of policy in task i induces a long-term expected discounted reward

$$V_\pi^i(s) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}^i(s_t, a_t) \mid s_0 = s, a_t \sim \pi(\cdot \mid s_t), s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)\right],$$

and the state-action value function Q

$$Q_\pi^i(s, a) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}^i(s_t, a_t) \mid s_0 = s, a_0 = a, s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t), a_{t+1} \sim \pi(\cdot \mid s_t)\right].$$

Thus, $V_\pi^i(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q_\pi^i(s, a)]$. Since $\mathcal{R}^i \in (0, 1)$ we have $|Q_\pi^i(s, a)| \leq 1/(1-\gamma)$ for all s, a. In addition, it is known that $Q_\pi^i(s, a)$ satisfies the Bellman equation

$$Q_\pi^i(s, a) = \mathcal{R}^i(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s' \mid s, a) V_\pi^i(s').$$

The goal of MTRL is to find a policy that simultaneously optimizes the aggregate of the value functions of the tasks at every state, i.e., the agent aims to find $\pi^\star$ such that $\sum_i V_{\pi^\star}^i(s)$ is maximized for every state $s \in \mathcal{S}$. It is known that for the finite MDP setting there exists such an optimal policy $\pi^\star$ Bellman and Dreyfus (1959). Finding $\pi^\star$ is essentially equivalent to solve

$$\max_\pi f(\pi) := \sum_{i=1}^{N} f^i(\pi). \tag{1}$$

where $f^i(\pi) = \mathbf{E}_{s \sim \rho^\star(\cdot)}[V_\pi^i(s)]$ and $\rho^\star$ is the stationary distribution corresponding to the optimal policy $\pi^\star$. Note that the knowledge of $\rho^\star$ is not required to the implementation of the proposed algorithm studied in the next section.

We conclude this section by introducing a few notation that will facilitate our algorithmic development in the next section. Given a policy π, we denote by $d_\pi^s(s')$ the discounted state visitation

$$d_\pi^s(s') = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathcal{P}_\pi(s_t = s' \mid s_0 = s),$$

which represents the amount of time that the agent visits state s' when it starts from s. Given two policies π, μ, let D_{π}^{μ} be the Bregman's distance defined as

$$D_{\pi}^{\mu}(s) = \omega(\mu(\cdot \mid s)) - \omega(\pi(\cdot \mid s)) - \langle \nabla_{\pi}\omega(\pi(\cdot \mid s)), \mu(\cdot \mid s) - \pi(\cdot \mid s)\rangle, \quad \forall s \in \mathcal{S},$$

where ω is a strongly convex function. In this paper, we will consider ω as an entropy function

$$\omega(\pi(\cdot \mid s)) = \sum_{a\in\mathcal{A}} \pi(a \mid s)\log(\pi(a \mid s)).$$

Under this choice, D_{π}^{μ} can be simplified to

$$D_{\pi}^{\mu}(s) = \sum_{a\in\mathcal{A}} \mu(a \mid s)\log\frac{\mu(a \mid s)}{\pi(a \mid s)},$$

which by using the Pinsker's inequality we obtain

$$\|\mu(\cdot \mid s) - \pi(\cdot \mid s)\|_2^2 \leq \|\mu(\cdot \mid s) - \pi(\cdot \mid s)\|_1^2 \leq 2D_{\pi}^{\mu}(s), \tag{2}$$

where the first inequality is due to the fact that $\|\cdot\|_2 \leq \|\cdot\|_1$

3 Incremental Policy Mirror Descent

To solve problem (1), one can apply the policy gradient approach, for example, the policy mirror descent method studied in Lan (2023). This method iteratively updates π_k, an estimate of $\pi^\star$, starting from an arbitrary policy π_0 as

$$\pi_{k+1}(\cdot \mid s) = \text{argmax}_{\mu(\cdot|s)\in\Delta_{|\mathcal{A}|}}\{\alpha_k\Big\langle \sum_i Q^i_{\pi_k}(s,\cdot), \mu(\cdot, s)\Big\rangle - D^{\mu}_{\pi_k}(s)\}.$$

To implement this update, one needs to estimate the state-value function Q^i for every task i at any iteration k, which is equivalent to solving N policy evaluation problems. This can be expensive in practice as the number of tasks can be large.

In this paper, we are interested in studying an incremental variant of the policy mirror descent methods, where the agent can only have access to the state-value function of one task at a time. Our algorithm is formally stated in Algorithm 1. At any iteration k, the agent chooses a task i to compute the state-value function $Q^i_{\pi_k}$ using the current policy π_k. The agent then implements a policy mirror descent step to update π_k as in (3). The main challenge of this setting is due to the appearance of the task is random, this result in a noise between the true gradient and the average gradient computed.

In Algorithm 1, we allow the agent to choose the task index i at any iteration arbitrarily. For example, the agent can apply the cyclic rule (i.e., choosing the task in increasing order) or random rule (i.e., randomly picking the task index). However, we do enforce that each task is chosen infinitely often to guarantee that the agent will perform all the tasks. One way to have this condition is to consider the following assumption.

Assumption 1 *Given a positive integer $\tau \geq N$, each $i \in \{1,\ldots,N\}$ in Step 1 of Algorithm 1 will be drawn at least one time within every interval $[k-\tau, k)$ for every $k \geq \tau$.*

One can view that τ represents an upper bound on the delay in computing the value of the state-action function Q^i at time k. We denote τ_k^i as the last time that task i is drawn at time k. If i is not selected at time k then $Q^i_{\tau_k^i}$ is the most recent Q value of task i using policy $\pi_{\tau_k^i}$. If $\tau = N$, then Algorithm 1 reduces to the classic incremental gradient method with cyclic rules.

Finally, we present below some preliminary results that will be useful in deriving our main result studied in the next section. First, we consider the so-called three-point lemma, which is used to characterize the updates of mirror descent. Its proof can be found in Lan (2023).

Algorithm 1 Incremental Policy Mirror Descent (IPMD)

```
Input: $\pi_0$, step sizes $\{\alpha_k\}_{k\geq 0}$
for $k = 0, 1, ..., K-1$ do
    [1] Draw $i \in \{1, \cdots, N\}$
    [2] Compute $Q_k^i \triangleq Q_{\pi_k}^i$ using the policy $\pi_k$
    [3] Update $\pi_k$ for all $s \in \mathcal{S}$ as
        $$\pi_{k+1}(\cdot \mid s) = \text{argmax}_{\mu(\cdot|s)\in\triangle_{|\mathcal{A}|}} \{\alpha_k \langle Q_k^i(s,\cdot), \mu(\cdot, s)\rangle - D_{\pi_k}^{\mu}(s)\}. \quad (3)$$
end
```

Lemma 1 *For any policy μ, the sequence $\{\pi_k\}$ generated by Algorithm 1 satisfies for all $s \in \mathcal{S}$*

$$\alpha_k \left\langle Q_k^i, \mu(\cdot \mid s) - \pi_{k+1}(\cdot \mid s)\right\rangle \leq D_{\pi_k}^{\mu}(s) - D_{\pi_{k+1}}^{\mu}(s) - D_{\pi_k}^{\pi_{k+1}}(s).$$

Next, we present the popular performance difference lemma in reinforcement learning Kakade and Langford (2002).

Lemma 2 *For any two policies π and π', we have*

$$V_{\pi'}^i(s) - V_\pi^i(s) = \frac{1}{1-\gamma}\sum_{s'\in\mathcal{S}} d_{\pi'}^s(s')\langle Q_\pi^i\left(s',\cdot\right), \pi'(\cdot \mid s') - \pi(\cdot \mid s')\rangle.$$

Finally, the following two lemmas are to characterize the properties of the sequence $\{\pi_k\}$ generated by Algorithm 1. Their analysis is presented in Appendix 6.1 and 6.2, respectively.

Lemma 3 *For any $s \in \mathcal{S}$, the sequence $\left\{\pi_{\tau_k^i}\right\}$ generated by Algorithm 1 satisfies for all $s \in \mathcal{S}$*

$$V_{\tau_k^i+1}^i(s) - V_{\tau_k^i}^i(s) \leq \left\langle Q_{\tau_k^i}^i(s,\cdot), \pi_{\tau_k^i+1}(\cdot \mid s) - \pi_{\tau_k^i}(\cdot \mid s)\right\rangle \leq \frac{-1}{\alpha_k}\left[D_{\pi_{\tau_k^i+1}}^{\pi_{\tau_k^i}}(s) + D_{\pi_{\tau_k^i}}^{\pi_{\tau_k^i+1}}(s)\right],$$

Lemma 4 *The sequence $\{\pi_k\}$ generated by Algorithm I satisfies for all $s \in \mathcal{S}$*

$$\|\pi_{k+1}(\cdot \mid s) - \pi_k(\cdot \mid s)\|_2 \leq \|\pi_{k+1}(\cdot \mid s) - \pi_k(\cdot \mid s)\|_1 \leq \frac{\alpha_k}{1-\gamma}.$$

4 Main Results

In this section, we will present the main result of this paper, where we will study the convergence rate of Algorithm 1. In particular, we show that under Assumption 1, Algorithm 1 can return an optimal policy of problem (1) at a sublinear rate. The following theorem is to present this result.

Theorem 5 *Suppose that Assumption 1 holds. Let $\{\pi_k\}_{k\geq 0}$ be generated by Algorithm 1 and step sizes be chosen as $\alpha_k = \frac{2}{\sqrt{k+\tau+1}}$. Then we have*

$$\max_{t=1,\cdots,k}\left[f(\pi^\star) - f(\pi_t)\right] \leq \frac{2[f(\pi^\star) - f(\pi_0)] + N\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D_{\pi_0}^{\pi^\star}(s)\right]}{4(1-\gamma)(\sqrt{k+\tau+2} - \sqrt{\tau+3})} + \frac{N\gamma\sqrt{\tau} + N(1+\gamma)\tau(1+\ln(k+\tau))}{(1-\gamma)^4(\sqrt{k+\tau+2} - \sqrt{\tau+3})}. \quad (4)$$

Remark 6 *It can be seen from Eq. (4) that the IPMD finds the optimal value of problem (1) at a rate $\mathcal{O}(1/\sqrt{k})$. This rate is much weaker than the result of centralized policy mirror descent studied in Lan (2023); Xiao (2022), where the rate is exponential using constant step sizes. However, we note that the existing analysis requires access to the full gradient of the objective in (1), i.e., $\sum_i Q^i_{\pi_k}$, at every iteration. On the other hand, the implementation of IPMD requires the Q function of only one task per iteration. Indeed, IPMD can be viewed as a stochastic variant of the existing policy mirror descent. In this setting, one needs to use decaying step sizes to guarantee an exact convergence to the optimal solution. On the other hand, to achieve an exponential convergence rate of the stochastic counterpart, the work in Lan (2023) assumes that the variance of the gradient samples decay exponentially fast. This assumption obviously does not hold in the context of IPMD. One can potentially apply the variance reduction techniques in Defazio et al. (2014); Johnson and Zhang (2013) to achieve this condition. This approach, however, is nontrivial since the analysis in Lan (2023) is not applicable due to the heterogeneity of the value functions of the tasks. Finally, our theoretical result also shows that the rate scales linearly with the number of tasks, quadratically on the delay interval τ, and cubically on the problem horizon $1/(1-\gamma)$. The convergence of policy mirror descent, however, only scales linearly with $1/(1-\gamma)$ Lan (2023); Xiao (2022). Addressing this theoretical gap will be an interesting topic, which we leave for our future studies.*

Proof Recall that τ^i_k is the last time that task i is selected in the time interval $[k-\tau, k]$. Using Lemma 1 with $\mu = \pi^\star$ we obtain for all $s \in \mathcal{S}$

$$\begin{aligned}
D^{\pi^\star}_{\pi_{\tau^i_k}}(s) - D^{\pi^\star}_{\pi_{\tau^i_k+1}}(s) - D^{\pi_{\tau^i_k}}_{\pi_{\tau^i_k+1}}(s) &\geq \alpha_{\tau^i_k}\left\langle Q^i_{\tau^i_k}(s,\cdot), \pi^\star(\cdot\mid s) - \pi_{\tau^i_k+1}(\cdot\mid s)\right\rangle \\
&= \alpha_{\tau^i_k}\left\langle Q^i_{\tau^i_k}(s,\cdot), \pi^\star(\cdot\mid s) - \pi_{\tau^i_k}(\cdot\mid s)\right\rangle + \alpha_{\tau^i_k}\left\langle Q^i_{\tau^i_k}(s,\cdot), \pi_{\tau^i_k}(\cdot\mid s) - \pi_{\tau^i_k+1}(\cdot\mid s)\right\rangle \\
&\geq \alpha_{\tau^i_k k}\left\langle Q^i_{\tau^i_k}(s,\cdot), \pi^\star(\cdot\mid s) - \pi_{\tau^i_k}(\cdot\mid s)\right\rangle + \alpha_{\tau^i_k}\left(V^i_{\tau^i_k}(s) - V^i_{\tau^i_k+1}(s)\right), \qquad (5)
\end{aligned}$$

where the last inequality used Lemma 3. We next consider the first term on the right-hand side of the preceding relation. By Lemma 2 and choose $\pi' = \pi^\star$, $\pi = \tau^i_k$ we have

$$(1-\gamma)\left(V^i_{\pi^\star}(s) - V^i_{\tau^i_k}(s)\right) = \mathbb{E}_{s'\sim d^s_{\pi^\star}(\cdot)}\left[\left\langle Q^i_{\tau^i_k}(s',\cdot), \pi^\star(\cdot\mid s') - \pi_{\tau^i_k}(\cdot\mid s')\right\rangle\right],$$

which by using the fact that $\mathbb{E}_{s\sim\rho^\star(\cdot)}\mathbb{E}_{s'\sim d^s_{\pi^\star}(\cdot)} = \mathbb{E}_{s\sim\rho^\star(\cdot)}$ gives

$$(1-\gamma)\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\tau^i_k}(s)\right] = \mathbb{E}_{s\sim\rho^\star(\cdot)}\left[\left\langle Q^i_{\tau^i_k}(s,\cdot), \pi^\star(\cdot\mid s) - \pi_{\tau^i_k}(\cdot\mid s)\right\rangle\right]. \qquad (6)$$

Take expectation w.r.t. $s \sim \rho^\star(\cdot)$ on both sides of Eq. (5) and using Eq. (6) we obtain

$$\begin{aligned}
&\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_{\tau^i_k}}(s) - D^{\pi^\star}_{\pi_{\tau^i_k+1}}(s) - D^{\pi_{\tau^i_k}}_{\pi_{\tau^i_k+1}}(s)\right] \\
&\quad\geq (1-\gamma)\alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\tau^i_k}(s)\right] + \alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\tau^i_k}(s) - V^i_{\tau^i_k+1}(s)\right] \\
&\quad= \alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\tau^i_k+1}(s)\right] - \gamma\alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\tau^i_k}(s)\right] \\
&\quad= \alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\pi_{k+1}}(s)\right] - \gamma\alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\pi_k}(s)\right] \\
&\qquad+ \alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi_{k+1}}(s) - V^i_{\tau^i_k+1}(s)\right] - \gamma\alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi_k}(s) - V^i_{\tau^i_k}(s)\right].
\end{aligned}$$

Reorganizing both sides of the preceding equation gives

$$\alpha_{\tau^i_k}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V^i_{\pi^\star}(s) - V^i_{\pi_{k+1}}(s)\right]$$

$$\begin{aligned}
&\leq \gamma\alpha_{\tau_k^i}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_k}^i(s)\right]+\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D_{\pi_{\tau_k^i}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi_{\tau_k^i}}(s)\right]\\
&\quad+\gamma\alpha_{\tau_k^i}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi_k}^i(s)-V_{\tau_k^i}^i(s)\right]+\alpha_{\tau_k^i}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\tau_k^i+1}^i(s)-V_{\pi_{k+1}}^i(s)\right].
\end{aligned}\tag{7}$$

We next analyze each term on the right-hand side of the preceding equation. First, by Lemma 2

$$\begin{aligned}
\left|V_{\pi_{k+1}}^i(s)-V_{\tau_k^i+1}^i(s)\right| &= \left|\frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\left[\left\langle Q_{\tau_k^i+1}^i\left(s',\cdot\right),\pi_{k+1}\left(\cdot\mid s'\right)-\pi_{\tau_k^i+1}\left(\cdot\mid s'\right)\right\rangle\right]\right|\\
&\leq \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\left\|Q_{\tau_k^i+1}^i\left(s',\cdot\right)\right\|_\infty\left\|\pi_{k+1}\left(\cdot\mid s'\right)-\pi_{\tau_k^i+1}\left(\cdot\mid s'\right)\right\|_1\\
&\leq \frac{1}{(1-\gamma)^2}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\left\|\pi_{k+1}\left(\cdot\mid s'\right)-\pi_{\tau_k^i+1}\left(\cdot\mid s'\right)\right\|_1\\
&= \frac{1}{(1-\gamma)^2}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\left\|\sum_{t=\tau_k^i+1}^{k}\pi_{\ell+1}\left(\cdot\mid s'\right)-\pi_\ell\left(\cdot\mid s'\right)\right\|_1\\
&\leq \frac{1}{(1-\gamma)^2}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\sum_{t=\tau_k^i+1}^{k}\left\|\pi_{\ell+1}\left(\cdot\mid s'\right)-\pi_\ell\left(\cdot\mid s'\right)\right\|_1,
\end{aligned}$$

where the first inequality is due to the Holder's inequality while we use the fact that $\left|Q_{\pi_t}^i(s,a)\right|\leq\frac{1}{1-\gamma}$ for all (s,a) in the second inequality. Using Lemma 4 and $\alpha_k\leq\alpha_{\tau_k^i}\leq\alpha_{k-\tau}$ for all $\tau_k^i\in(k-\tau,k]$, we have from the relation above

$$\begin{aligned}
\left|V_{\pi_{k+1}}^i(s)-V_{\tau_k^i+1}^i(s)\right| &\leq \frac{1}{(1-\gamma)^2}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\sum_{t=k-\tau+1}^{k}\left\|\pi_{\ell+1}\left(\cdot\mid s'\right)-\pi_\ell\left(\cdot\mid s'\right)\right\|_1\\
&\leq \frac{1}{(1-\gamma)^3}\mathbb{E}_{s'\sim d_{\pi_{k+1}}^s(\cdot)}\sum_{t=k-\tau+1}^{k}\alpha_t\leq\frac{\tau\alpha_{k-\tau}}{(1-\gamma)^3},
\end{aligned}\tag{8}$$

where the last inequality is due to the property of discounted state distribution. Using the same argument as above we can have

$$\left|V_{\tau_k^i}^i(s)-V_{\pi_k}^i(s)\right|\leq\frac{\tau\alpha_{k-\tau}}{(1-\gamma)^3}.\tag{9}$$

Substituting Eqs. (8) and (9) into Eq. (7) and use the fact that $\alpha_{\tau_k^i}\leq\alpha_{k-\tau}$ we obtain

$$\begin{aligned}
\alpha_{\tau_k^i}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_{k+1}}^i(s)\right] &\leq \gamma\alpha_{\tau_k^i}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_k}^i(s)\right]+\frac{(1+\gamma)\tau(\alpha_{k-\tau})^2}{(1-\gamma)^3}\\
&\quad+\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D_{\pi_{\tau_k^i}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi_{\tau_k^i}}(s)\right],
\end{aligned}$$

which yields

$$\begin{aligned}
\alpha_{k+1}\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_{k+1}}^i(s)\right] &\leq \gamma\alpha_k\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_k}^i(s)\right]+\frac{(1+\gamma)\tau(\alpha_{k-\tau})^2}{(1-\gamma)^3}\\
&\quad+\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D_{\pi_{\tau_k^i}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi_{\tau_k^i}}(s)\right]+\gamma(\alpha_{\tau_k^i}-\alpha_k)\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_k}^i(s)\right]\\
&\leq \gamma\alpha_k\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[V_{\pi^\star}^i(s)-V_{\pi_k}^i(s)\right]+\frac{(1+\gamma)\tau(\alpha_{k-\tau})^2}{(1-\gamma)^3}\\
&\quad+\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D_{\pi_{\tau_k^i}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi^\star}(s)-D_{\pi_{\tau_k^i+1}}^{\pi_{\tau_k^i}}(s)\right]+\frac{2\gamma}{1-\gamma}(\alpha_{\tau_k^i}-\alpha_k).
\end{aligned}$$

Taking the average of the preceding relation over $i \in [1, N]$ we obtain

$$\alpha_{k+1}\left[f(\pi^\star) - f(\pi_{k+1})\right] \leq \gamma\alpha_k\left[f(\pi^\star) - f(\pi_k)\right] + \sum_{i=1}^{N} \frac{(1+\gamma)\tau(\alpha_{k-\tau})^2}{(1-\gamma)^3}$$

$$+ \sum_{i=1}^{N} \mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_{\tau_k^i}}(s) - D^{\pi^\star}_{\pi_{\tau_k^i+1}}(s) - D^{\pi_{\tau_k^i}}_{\pi_{\tau_k^i+1}}(s)\right] + \sum_{i=1}^{N} \frac{2\gamma}{1-\gamma}(\alpha_{\tau_k^i} - \alpha_k).$$

Divide both side by $\gamma\alpha_{k+1}\left[f(\pi^\star) - f(\pi_{k+1})\right]$ gives

$$(1-\gamma)\alpha_{k+1}\left[f(\pi^\star) - f(\pi_{k+1})\right] \leq \gamma\left(\alpha_k[f(\pi^\star) - f(\pi_k)] - \alpha_{k+1}[f(\pi^\star) - f(\pi_{k+1})]\right)$$

$$+ \frac{N(1+\gamma)\tau(\alpha_{k-\tau})^2}{(1-\gamma)^3} + \sum_{i=1}^{N} \mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_{\tau_k^i}}(s) - D^{\pi^\star}_{\pi_{\tau_k^i+1}}(s)\right] + \frac{2N\gamma}{1-\gamma}(\alpha_{k-\tau} - \alpha_k).$$

Summing up both sides of the preceding inequality over $\tau_k^i = k-\tau+1, \ldots, k$ gives

$$(1-\gamma)\tau\alpha_{k+1}\left[f(\pi^\star) - f(\pi_{k+1})\right] \leq \gamma\tau\left(\alpha_k[f(\pi^\star) - f(\pi_k)] - \alpha_{k+1}[f(\pi^\star) - f(\pi_{k+1})]\right)$$

$$+ \frac{2\tau N\gamma}{1-\gamma}(\alpha_{k-\tau} - \alpha_k) + \frac{N(1+\gamma)\tau^2(\alpha_{k-\tau})^2}{(1-\gamma)^3} + \sum_{i=1}^{N}\sum_{\tau_k^i=k-\tau+1}^{k} \mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_{\tau_k^i}}(s) - D^{\pi^\star}_{\pi_{\tau_k^i+1}}(s)\right],$$

which when summing up both sides over k yields

$$(1-\gamma)\tau\sum_{t=0}^{k-1}\alpha_{t+1}\left[f(\pi^\star) - f(\pi_{t+1})\right] \leq \gamma\tau\left(\alpha_0[f(\pi^\star) - f(\pi_0)] - \alpha_{k+1}[f(\pi^\star) - f(\pi_{k+1})]\right)$$

$$+ \sum_{t=0}^{k-1}\frac{2\tau N\gamma}{1-\gamma}(\alpha_{t-\tau} - \alpha_t) + \sum_{t=0}^{k-1}\frac{N(1+\gamma)\tau^2(\alpha_{t-\tau})^2}{(1-\gamma)^3}$$

$$+ \sum_{i=1}^{N}\sum_{t=0}^{k-1}\sum_{\tau_k^i=t-\tau+1}^{t} \mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_{\tau_k^i}}(s) - D^{\pi^\star}_{\pi_{\tau_k^i+1}}(s)\right]$$

$$\leq \gamma\tau\alpha_0[f(\pi^\star) - f(\pi_0)] + \sum_{t=0}^{k-1}\frac{2\tau N\gamma}{1-\gamma}(\alpha_{t-\tau} - \alpha_t) + N\tau\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_0}(s)\right]$$

$$+ \sum_{t=0}^{k-1}\frac{N(1+\gamma)\tau^2(\alpha_{t-\tau})^2}{(1-\gamma)^3}. \tag{10}$$

Next, using $\alpha_k = \frac{2}{\sqrt{k+\tau+1}}$ for $k \geq 0$ and let $\alpha_k = \alpha_0$ for $k < 0$ we have

$$\sum_{t=0}^{k-1}\alpha_{t+1} \geq \frac{2}{\sqrt{\tau+2}} + \int_1^k \frac{2}{\sqrt{t+\tau+2}}dt \geq 4\sqrt{k+\tau+2} - 4\sqrt{\tau+3}.$$

$$\sum_{t=0}^{k-1}(\alpha_{t-\tau})^2 \leq \sum_{t=0}^{\tau-1}(\alpha_0)^2 + (\alpha_0)^2 + \int_\tau^{k-1}\frac{4}{t+\tau+1}dt \leq 4 + 4\ln(k+\tau).$$

$$\sum_{t=0}^{k-1}(\alpha_{t-\tau} - \alpha_t) = \sum_{t=0}^{\tau-1}\alpha_0 + \sum_{t=0}^{k-\tau-1}\alpha_t - \sum_{t=0}^{k-1}\alpha_t \leq 2\sqrt{\tau}.$$

Using these relations into Eq. (10) we have

$$4(1-\gamma)\tau(\sqrt{k+\tau+2} - \sqrt{\tau+3})\max_{t=1,\cdot,k}\left[f(\pi^\star) - f(\pi_t)\right]$$

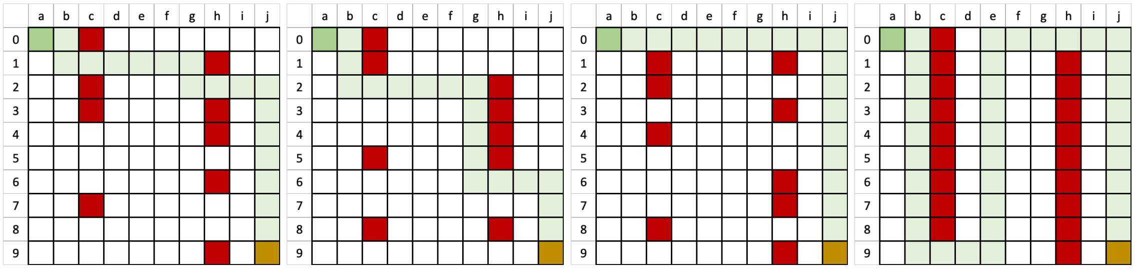
$$\leq \gamma\tau\alpha_0[f(\pi^\star) - f(\pi_0)] + N\tau\mathbb{E}_{s\sim\rho^\star(\cdot)}\left[D^{\pi^\star}_{\pi_0}(s)\right] + \frac{4N\gamma\sqrt{\tau^3}}{1-\gamma} + \frac{4N(1+\gamma)\tau^2(1+\ln(k+\tau))}{(1-\gamma)^3},$$

which by dividing both side by $4(1-\gamma)\tau(\sqrt{k+\tau+2} - \sqrt{\tau+3})$ and using $\gamma < 1$ we immediately obtain (4). ■

5 Simulation

In this section, we will illustrate the convergence of IPMD in Algorithm 1 in solving a multi-task GridWorld problem. The goal of this simulation is to investigate the performance of IPMD.

GridWorld Environments. We create six GridWorld environments, each has size 10×10. Each environment has the same initial starting point (the blue square in the top left corner) and the goal (the yellow square in the bottom right corner). The goal of the agent is to find a policy that can help it navigate from the starting point to the target while avoiding all the obstacles (the red square). The obstacles in each environment are located in different positions in the grid. The first three figures in Figure 1 presents some example of GridWorld environments, while the last figure in Figure 1 shows the obstacles at all the environments put together. There is an optimal path that can solve the task simultaneously (e.g., light green path in the last figure), while there are multiple different solutions for each task (e.g., light green paths in the first three figures). To solve this multi-task problem, the agent seeks to find the light green path in the last figure. For our MTRL setting, we assign the reward being $+1$ at the target while the agent gets a -1 if it runs into the obstacles. For our simulation, we implement Algorithm 1 with a cyclic rule and decreasing step-size as in Theorem 5, where the environments are chosen in increasing order. In addition, since the transition probability matrices of these environments are known, we can easily compute the state-action value function Q at each iteration, e.g., by solving the Bellman equation. The results of our simulation are shown in Figure 2, where we show the error (the difference between these rewards and the optimal value) on the left and the log plot of it compare with $\frac{f(\pi^\star)-f(\pi_0)}{\sqrt{k}}$. We can see that the proposed algorithm returns an optimal policy that can solve the multi-task GridWorld problem, which agrees with our theoretical results in Theorem 5. In this simulation, the rate of convergence seems to be faster than $1/\sqrt{k}$, which implies that the current analysis might not be tight. We note that the convergence rate of PMD for solving problem (1) is linear. Thus, one might need a new approach to study the best theoretical results on the performance of IPMD.



GridWorld 0 GridWorld 1 GridWorld 2 Total GridWorld

Figure 1: GridWorlds

Acknowledgement

This work was supported by the National Science Foundation under awards ECCS-CAREER-2339509 and CCF-2343599.

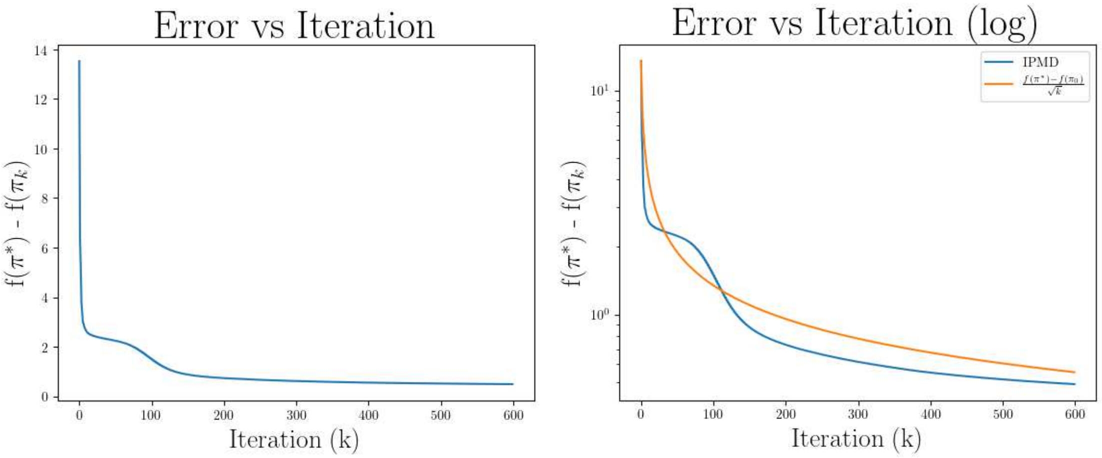


Figure 2: Performance of IPMD Methods

6 Appendix

6.1 Proof of Lemma 3

First, using Lemma 1 with $\mu = \pi_{\tau_k^i}$ we have

$$\langle Q^i_{\tau_k^i}\left(s',\cdot\right), \pi_{\tau_k^i}(\cdot \mid s') - \pi_{\tau_k^i+1}(\cdot \mid s')\rangle \leq -\frac{1}{\alpha_k}\left[D^{\pi_{\tau_k^i}}_{\pi_{\tau_k^i+1}}(s) + D^{\pi_{\tau_k^i+1}}_{\pi_{\tau_k^i}}(s)\right] \leq 0.$$

Next, using Lemma 2 with $\pi' = \pi_{\tau_k^i+1}$ and $\pi = \pi_{\tau_k^i}$, and the relation above gives

$$\begin{aligned} V^i_{\tau_k^i}(s) - V^i_{\tau_k^i+1}(s) &= \frac{1}{1-\gamma}\sum_{s'\in\mathcal{S}} d^s_{\tau_k^i+1}(s')\langle Q^i_{\tau_k^i}\left(s',\cdot\right), \pi_{\tau_k^i}(\cdot \mid s') - \pi_{\tau_k^i+1}(\cdot \mid s')\rangle \\ &\leq \frac{1}{1-\gamma} d^s_{\tau_k^i+1}(s)\langle Q^i_{\tau_k^i}\left(s,\cdot\right), \pi_{\tau_k^i}(\cdot \mid s) - \pi_{\tau_k^i+1}(\cdot \mid s)\rangle \\ &\leq \langle Q^i_{\tau_k^i}\left(s,\cdot\right), \pi_{\tau_k^i}(\cdot \mid s) - \pi_{\tau_k^i+1}(\cdot \mid s)\rangle \\ &\leq -\frac{1}{\alpha_k}\left[D^{\pi_{\tau_k^i}}_{\pi_{\tau_k^i+1}}(s) + D^{\pi_{\tau_k^i+1}}_{\pi_{\tau_k^i}}(s)\right], \end{aligned}$$

where in the second inequality we use the fact that $d^s_{\tau_k^i+1}(s) \geq 1-\gamma$.

6.2 Proof of Lemma 4

Without loss of generality, let task i be chosen at time $t \in [k-\tau, k]$. Thus, using Lemma 1 with $\mu = \pi_t$ we have

$$\alpha_t \left\langle Q^i_{\pi_t}(s,\cdot), \pi_t(\cdot \mid s) - \pi_{t+1}(\cdot \mid s)\right\rangle \leq -D^{\pi_t}_{\pi_{t+1}}(s) - D^{\pi_{t+1}}_{\pi_t}(s) \leq -\left\|\pi_{t+1}(\cdot \mid s) - \pi_t(\cdot \mid s)\right\|_1^2,$$

where the last inequality is due to (2). Since $\left|Q^i_{\pi_t}(s,a)\right| \leq \frac{1}{1-\gamma}$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, multiply both side by -1 and use the Holder's inequality the preceding relation gives

$$\begin{aligned} \left\|\pi_{t+1}(\cdot \mid s) - \pi_t(\cdot \mid s)\right\|_1^2 &\leq \alpha_t \left\|\pi_{t+1}(\cdot \mid s) - \pi_t(\cdot \mid s)\right\|_1 \left\|Q^i_{\pi_t}(s,\cdot)\right\|_\infty \\ &\leq \frac{\alpha_t}{1-\gamma}\left\|\pi_{t+1}(\cdot \mid s) - \pi_t(\cdot \mid s)\right\|_1. \end{aligned}$$

References

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in Markov decision processes. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 64–66. PMLR, 09–12 Jul 2020.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Richard Bellman and Stuart Dreyfus. Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, 13(68):247–251, 1959.

Dimitri P Bertsekas et al. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

Jalaj Bhandari and Daniel Russo. A note on the linear convergence of policy gradient methods. *arXiv preprint arXiv:2007.11120*, page 79, 2020.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578, 2022.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligent Research*, 75:1401–1476, 2020.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

Guangchen Lan, Han Wang, James Anderson, Christopher Brinton, and Vaneet Aggarwal. Improved communication efficiency in federated natural policy gradient via admm-based gradient updates. *arXiv preprint arXiv:2310.19807*, 2023.

Guanghui Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical programming*, 198(1):1059–1106, 2023.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pages 6820–6829. PMLR, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for nonconvex optimization. *arXiv preprint arXiv:1603.06159*, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*, 2019.

Lin Xiao. On the convergence rates of policy gradient methods. *The Journal of Machine Learning Research*, 23(1):12887–12922, 2022.

Sihan Zeng, Malik Aqeel Anwar, Thinh T. Doan, Arijit Raychowdhury, and Justin Romberg. A decentralized policy gradient approach to multi-task reinforcement learning. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1002–1012. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/zeng21a.html.