Provably Efficient Multi-Task Reinforcement Learning with Model Transfer

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study multi-task reinforcement learning (RL) in tabular episodic Markov decision processes (MDPs). We formulate a heterogeneous multi-player RL problem, in which a group of players concurrently face similar but not necessarily identical MDPs, with a goal of improving their collective performance through inter-player information sharing. We design and analyze a model-based algorithm, and provide gap-dependent and gap-independent upper and lower bounds that characterize the intrinsic complexity of the problem.

1 Introduction

In many real-world applications, reinforcement learning (RL) agents can be deployed as a group to complete similar tasks at the same time. For example, in healthcare robotics, robots are paired with people with dementia to perform personalized cognitive training activities by learning their preferences [36, 17]; in autonomous driving, a set of autonomous vehicles learn how to navigate and avoid obstacles in various environments [22]. In these settings, each learning agent alone may only be able to acquire a limited amount of data, while the agents as a group have the potential to collectively learn faster through sharing knowledge among themselves. Multi-task learning [5] is a practical framework that can be used to model such settings, where a set of learning agents share/transfer knowledge to improve their collective performance.

Despite many empirical successes of multi-task RL [e.g., 45, 23, 22] and transfer learning for RL [e.g., 21, 33], a theoretical understanding of when and how information sharing or knowledge transfer can provide benefits remains limited. Exceptions include [e.g., 12, 4, 9, 13, 26, 20], which study multi-task learning from parameter or representation transfer perspectives. However, these works still do not provide a completely satisfying answer: for example, in many application scenarios, the reward structures and the environment dynamics are only slightly different for each task—this is, however, not captured by representation transfer [9, 13] or existing works on clustering-based parameter transfer [12, 4]. In such settings, is it possible to design provably efficient multi-task RL algorithms that have guarantees never worse than agents learning individually, while outperforming the individual agents in favorable situations?

In this work, we formulate a multi-task RL problem that is applicable to the aforementioned settings. Specifically, inspired by a recent study on multi-task multi-armed bandits [37], we formulate the ϵ-Multi-Player Episodic Reinforcement Learning (abbreviated as ϵ-MPERL) problem, in which all tasks share the same state and action spaces, and the tasks are assumed to be similar—i.e., the dissimilarities between the environments of different tasks (specifically, the reward distributions and transition dynamics associated with the players/tasks) are bounded in terms of a dissimilarity parameter $\epsilon \geq 0$. This problem not only models concurrent RL [28, 12] as a special case by taking $\epsilon = 0$, but also captures richer multi-task RL settings when ϵ is nonzero. In this work, we study regret minimization for the ϵ-MPERL problem, specifically:

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

1. We identify a problem complexity notion named *subpar* state-action pairs, which captures the amenability of information sharing among tasks in ϵ-MPERL problem instances. As shown in the multi-task bandits literature [e.g., 37], information sharing is *not* always helpful. Subpar state-action pairs, intuitively speaking, are clearly suboptimal for all tasks, for which we can robustly take advantage of (possibly biased) data collected for other tasks.
2. In the setting where the dissimilarity parameter ϵ is known, we design a model-based algorithm MULTI-TASK-EULER (Algorithm 1), which is built upon state-of-the-art algorithms for learning single-task Markov decision processes (MDPs) [40, 30]. MULTI-TASK-EULER crucially utilizes the dissimilarity assumption to robustly take advantage of information sharing among tasks, and achieves regret upper bounds in terms of subpar state-action pairs, in both (suboptimality) gap-dependent and gap-independent fashions. Specifically, compared with a baseline algorithm that does not utilize information sharing, MULTI-TASK-EULER has a regret guarantee that: (1) is never worse, i.e., it avoids negative transfer [27]; (2) can be much superior when there are a large number of subpar state-action pairs.
3. We also present gap-dependent and gap-independent regret lower bounds for the ϵ-MPERL problem in terms of subpar state-action pairs. Together, the upper and lower bounds characterize the intrinsic complexity of the ϵ-MPERL problem.

2 Preliminaries

Throughout this paper, we denote by $[n] =: \{1, \ldots, n\}$. For a set A, we use A^C to denote its complement. Denote by $\Delta(\mathcal{X})$ the set of probability distributions over $\mathcal{X}$. For functions f, g, we use $f \lesssim g$ (resp. $f \gtrsim g$) to denote that there exists some constant $c > 0$, such that $f \leq cg$ (resp. $f \geq cg$), and use $f \eqsim g$ to denote $f \lesssim g$ and $f \gtrsim g$ simultaneously. Define $a \vee b := \min(a, b)$, and $a \wedge b := \max(a, b)$. We use $\mathbb{E}$ to denote the expectation operator, and use var to denote the variance operator.

Multi-task RL in episodic MDPs. We have a set of M MDPs $\left\{\mathcal{M}_p = (H, \mathcal{S}, \mathcal{A}, p_0, \mathbb{P}_p, r_p)\right\}_{p=1}^M$, each associated with a player $p \in [M]$. Each MDP $\mathcal{M}_p$ is regarded as a task. The MDPs share the same episode length $H \in \mathbb{N}_+$, finite state space $\mathcal{S}$, finite action space $\mathcal{A}$, and initial state distribution $p_0 \in \Delta(\mathcal{S})$. The transition probabilities $\mathbb{P}_p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ and reward distributions $r_p : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$ of the players are not necessarily identical. We assume that the MDPs are layered[1], in that the state space $\mathcal{S}$ can be partitioned into disjoint subsets $(\mathcal{S}_h)_{h=1}^H$, where p_0 is supported on $\mathcal{S}_1$, and for every $p \in [M]$, $h \in [H]$, and every $s \in \mathcal{S}_h, a \in \mathcal{A}$, $\mathbb{P}_p(\cdot \mid s, a)$ is supported on $\mathcal{S}_{h+1}$; here, we define $\mathcal{S}_{H+1} = \{\perp\}$ so that it contains a default terminal state $\perp$ (note that $\perp \notin \mathcal{S}$). We denote by $S := |\mathcal{S}|$ the size of the state space, and $A := |\mathcal{A}|$ the size of the action space.

Interaction process. The interaction process between the players and the environment is as follows: at the beginning, both $(r_p)_{p=1}^M$ and $(\mathbb{P}_p)_{p=1}^M$ are unknown to the players. For each episode $k \in [K]$, each player $p \in [M]$ interacts with its respective MDP $\mathcal{M}_p$; specifically, player p starts with state $s_{1,p}^k \sim p_0$, and at every step $h \in [H]$, it chooses action $a_{h,p}^k$, transitions to next state $s_{h+1,p}^k \sim \mathbb{P}_p(\cdot \mid s_{h,p}^k, a_{h,p}^k)$ and receives a stochastic immediate reward $r_{h,p}^k \sim r_p(\cdot \mid s_{h,p}^k, a_{h,p}^k)$; after all players have finished their k-th episode, they can communicate and share information. The goal of the players is to maximize their expected collective reward $\mathbb{E}\left[\sum_{k=1}^K \sum_{p=1}^M \sum_{h=1}^H r_{h,p}^k\right]$.

Policy and value functions. A deterministic policy π is a mapping from $\mathcal{S}$ to $\mathcal{A}$, which can be used by a player to make decisions in its respective MDP. For player p and step h, we define the value function $V_{h,p}^\pi : \mathcal{S}_h \to [0, H]$ and the action value function $Q_{h,p}^\pi : \mathcal{S}_h \times \mathcal{A} \to [0, H]$ as the expected return of player p conditioned on its being at a state at step h, and its being at a state and taking an action at step h, respectively. They satisfy the following recursive formula known as the Bellman equation:

$$\forall h \in [H] : \quad V_{h,p}^\pi(s) = Q_{h,p}^\pi(s, \pi(s)), \quad Q_{h,p}^\pi(s, a) = R_p(s, a) + (\mathbb{P}_p V_{h+1,p}^\pi)(s, a),$$

[1]This is a standard assumption (see [e.g., 38]). It is noteworthy that any episodic MDP (with possibly nonstationary transition and reward) can be converted to a layered MDP with stationary transition and reward, with the state space size being H times the size of the original state space.

where we use the convention that $V^{\pi}_{H+1,p}(\perp) = 0$, and for $f : \mathcal{S}_{h+1} \to \mathbb{R}$, $(\mathbb{P}_p f)(s,a) := \sum_{s' \in \mathcal{S}_{h+1}} \mathbb{P}_p(s' \mid s,a) f(s')$, and $R_p(s,a) = \mathbb{E}_{\hat{r} \sim r_p(\cdot|s,a)}[\hat{r}]$ is the expected immediate reward of player p. For player p and policy π, denote by $V^{\pi}_{0,p} = \mathbb{E}_{s_1 \sim p_0}\left[V^{\pi}_{1,p}(s_1)\right]$ its expected reward.

For player p, we also define its optimal value function $V^{\star}_{h,p} : \mathcal{S}_h \to [0,H]$ and the optimal action value function $Q^{\star}_{h,p} : \mathcal{S}_h \times \mathcal{A} \to [0,H]$ using the Bellman optimality equation:

$$\forall h \in [H] : \quad V^{\star}_{h,p}(s) = \max_{a \in \mathcal{A}} Q^{\star}_{h,p}(s,a), \;\; Q^{\star}_{h,p}(s,a) = R_p(s,a) + (\mathbb{P}_p V^{\star}_{h+1,p})(s,a), \qquad (1)$$

where we again use the convention that $V^{\star}_{H+1,p}(\perp) = 0$. For player p, denote by $V^{\star}_{0,p} = \mathbb{E}_{s_1 \sim p_0}\left[V^{\star}_{1,p}(s_1)\right]$ its optimal expected reward.

Given a policy π, as $V^{\pi}_{h,p}$ for different h's are only defined in the respective layer $\mathcal{S}_h$, we "collate" the value functions $(V^{\pi}_{h,p})_{h=1}^{H}$ and obtain a single value function $V^{\pi}_p : \mathcal{S} \to \mathbb{R}$. Formally, for every $h \in [H]$ and $s \in \mathcal{S}_h$,

$$V^{\pi}_p(s) := V^{\pi}_{h,p}(s).$$

We define $Q^{\pi}_p, V^{\star}_p, Q^{\star}_p$ similarly. For player p, given its optimal action value functions $Q^{\star}_p$, its optimal policy $\pi^{\star}_p : \mathcal{S} \to \mathcal{A}$ is greedy with respect to $Q^{\star}_p$, that is, $\pi^{\star}_p(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^{\star}_p(s,a)$.

Suboptimality gap. For player p, we define the suboptimality gap of state-action pair (s,a) as $\mathrm{gap}_p(s,a) = V^{\star}_p(s) - Q^{\star}_p(s,a)$. We define the mininum suboptimality gap of player p as $\mathrm{gap}_{p,\min} = \min_{(s,a):\mathrm{gap}_p(s,a)>0} \mathrm{gap}_p(s,a)$, and the minimum suboptimailty gap over all players as $\mathrm{gap}_{\min} = \min_{p \in [M]} \mathrm{gap}_{p,\min}$. For player $p \in [M]$, define $Z_{p,\mathrm{opt}} := \left\{(s,a) : \mathrm{gap}_p(s,a) = 0\right\}$ as the set of optimal state-action pairs with respect to p.

Performance metric. The performance metric of the players studied in this paper is their collective regret, i.e., over a total of K episodes, how much extra reward they would have collected in expectation if they were executing their respective optimal policies from the beginning. Formally, suppose for each episode k, player p executes policy $\pi^k(p)$, then the collective regret of the players is defined as:

$$\mathrm{Reg}(K) = \mathbb{E}\left[\sum_{p=1}^{M}\sum_{k=1}^{K}\left(V^{\star}_{0,p} - V^{\pi^k(p)}_{0,p}\right)\right] = \mathbb{E}\left[\sum_{p=1}^{M}\sum_{k=1}^{K}\left(V^{\star}_{1,p}(s^k_{1,p}) - V^{\pi^k(p)}_{1,p}(s^k_{1,p})\right)\right]$$

Baseline: individual EULER. A naive baseline for multi-player RL is to let each player run a separate RL algorithm without communication. For concreteness, we choose to let each player run the state of the art EULER algorithm [40, 30], which enjoys minimax gap independent [2, 6] and gap-dependent regret guarantees, and refer to this strategy as individual EULER. Specifically, as it is known that EULER has a regret of $O(\sqrt{H^2SAK})$, individual EULER has a collective regret of $O(M\sqrt{H^2SAK})$. In addition, by summing up the gap-dependent regret guarantee of EULER for the M MDPs altogether, it can be easily checked that individual EULER has a collective regret of

$$\mathrm{Reg}(K) \lesssim \ln(MSAHK)\left(\sum_{p \in [M]}\left(\sum_{(s,a) \in Z_{p,\mathrm{opt}}} \frac{H^3}{\mathrm{gap}_{p,\min}} + \sum_{(s,a) \in Z^C_{p,\mathrm{opt}}} \frac{H^3}{\mathrm{gap}_p(s,a)}\right) + MS^2AH^3 \ln \frac{MH}{\mathrm{gap}_{\min}}\right),$$

for any large enough K. Our goal is to design multi-task RL algorithms that can achieve collective regret strictly lower than this baseline in both gap-dependent and gap-independent fashions.

Notion of similarity. Throughout this paper, we will consider the following notion of similarity between MDPs in the multi-task episodic RL setting.

Definition 1. *A collection of MDPs $(\mathcal{M}_p)_{p=1}^{M}$ is said to be ϵ-dissimilar, if for all $p, q \in [M]$, and $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\left|R_p(s,a) - R_q(s,a)\right| \le \epsilon, \;\; \|\mathbb{P}_p(\cdot \mid s,a) - \mathbb{P}_q(\cdot \mid s,a)\|_1 \le \frac{\epsilon}{H}.$$

Algorithm 1: MULTI-TASK-EULER

Initialize: Set $V_p(\perp) = 0$ for all p in $[M]$, where $\perp$ is the only state in $\mathcal{S}_{H+1}$;

for $k = 1, 2, \ldots, K$ **do**
  for $p = 1, 2, \ldots, M$ **do**
    // Construct optimal value estimates for player p
    for $h = H, H-1, \ldots, 1$ **do**
      for $(s, a) \in \mathcal{S}_h \times \mathcal{A}$ **do**
        Compute:
        $\overline{\text{ind-}Q}_p(s,a) = \hat{R}_p(s,a) + (\hat{\mathbb{P}}_p \overline{V}_p)(s,a) + \text{ind-}b_p(s,a)$;
        $\underline{\text{ind-}Q}_p(s,a) = \hat{R}_p(s,a) + (\hat{\mathbb{P}}_p \underline{V}_p)(s,a) - \text{ind-}b_p(s,a)$;
        $\overline{\text{agg-}Q}_p(s,a) = \hat{R}(s,a) + (\hat{\mathbb{P}} \overline{V}_p)(s,a) + \text{agg-}b_p(s,a)$;
        $\underline{\text{agg-}Q}_p(s,a) = \hat{R}(s,a) + (\hat{\mathbb{P}} \underline{V}_p)(s,a) - \text{agg-}b_p(s,a)$;
        Update optimal action value function upper and lower bound estimates:
        $\overline{Q}_p(s,a) = \min\left\{H, \overline{\text{ind-}Q}_p(s,a), \underline{\text{ind-}Q}_p(s,a)\right\}$;
        $\underline{Q}_p(s,a) = \max\left\{0, \underline{\text{ind-}Q}_p(s,a), \underline{\text{agg-}Q}_p(s,a)\right\}$;
      for $s \in \mathcal{S}_h$ **do**
        Define $\pi^k(p)(s) = \text{argmax}_{a \in \mathcal{A}} \overline{Q}_p(s,a)$;
        Update $\overline{V}_p(s) = \overline{Q}_p(s, \pi^k(p)(s))$, $\underline{V}_p(s) = \underline{Q}_p(s, \pi^k(p)(s))$.
    // All players p interact with their respective environments, and update reward and transition estimates
    for $p = 1, 2, \ldots, M$ **do**
      Player p executes policy $\pi^k(p)$ on $\mathcal{M}_p$ and obtains trajectory $(s^k_{h,p}, a^k_{h,p}, r^k_{h,p})_{h=1}^H$.
      Update individual estimates of transition probability $\hat{\mathbb{P}}_p$, reward $\hat{R}_p$ and count $n_p(\cdot,\cdot)$.
    Update aggregate estimates of transition probability $\hat{\mathbb{P}}$, reward $\hat{R}$ and count $n(\cdot,\cdot)$.

If this happens, we call $(\mathcal{M}_p)_{p=1}^M$ an ϵ-Multi-Player Episodic Reinforcement Learning (abbrev. ϵ-MPERL) problem instance.

If the MDPs in $(\mathcal{M}_p)_{p=1}^M$ are 0-dissimilar, then they are identical by definition, and our interaction protocol degenerates to the concurrent RL protocol [28]. Our dissimilarity notion is complementary to those of [4, 12], in that they require the MDPs to be either identical, or have well-separated parameters for at least one state-action pair; in contrast, our dissimilarity notion allows the MDPs to be nonidentical and arbitrarily close.

We have the following intuitive lemma that shows the closeness of optimal value functions of different MDPs, in terms of the dissimilarity parameter ϵ:

Lemma 2. *If $(\mathcal{M}_p)_{p=1}^M$ are ϵ-dissimilar, then for every $p, q \in [M]$, and $(s,a) \in \mathcal{S} \times \mathcal{A}$, $\left|Q^\star_p(s,a) - Q^\star_q(s,a)\right| \le 2H\epsilon$; consequently, $\left|\text{gap}_p(s,a) - \text{gap}_q(s,a)\right| \le 4H\epsilon$.*

3 Algorithm

We now describe our main algorithm, MULTI-TASK-EULER (Algorithm 1). Our model-based algorithm is built upon recent works on episodic RL that provide algorithms with sharp instance-dependent guarantees in the single task setting [40, 30]. In a nutshell, for each episode k and each player p, the algorithm performs optimistic value iteration to construct high-probability upper and lower bounds for the optimal value and action value functions $V^\star_p$ and $Q^\star_p$, and use them to guide its exploration and decision making process.

Empirical estimates of model parameters. For each player p, the construction of its value function bound estimates relies on empirical estimates on its transition probability and expected reward function. For both estimands, we use two estimators with complementary roles, which are at two different points of the bias-variance tradeoff spectrum: one estimator uses only the player's own data (termed *individual estimate*), which is unbiased but has large variance, the other estimator uses the data collected by all players (termed *aggregate estimate*), which is biased but has lower variance. Specifically, at the end of episode k, for every $h \in [H]$ and $(s, a) \in \mathcal{S}_h \times \mathcal{A}$, the algorithm maintains its empirical count of encountering (s, a) for each player p, along with its total empirical count across all players, respectively:

$$n_p(s,a) := \sum_{l=1}^{k} \mathbf{1}\left((s_{h,p}^l, a_{h,p}^l) = (s,a)\right), \; n(s,a) := \sum_{l=1}^{k}\sum_{p=1}^{M} \mathbf{1}\left((s_{h,p}^l, a_{h,p}^l) = (s,a)\right).$$

The individual and aggregate estimates of immediate reward $R(s, a)$ are defined as:

$$\hat{R}_p(s,a) := \frac{\sum_{l=1}^{k} \mathbf{1}\left((s_{h,p}^l, a_{h,p}^l) = (s,a)\right) r_{h,p}^l}{n_p(s,a)}, \; \hat{R}(s,a) := \frac{\sum_{l=1}^{k}\sum_{p=1}^{M} \mathbf{1}\left((s_{h,p}^l, a_{h,p}^l) = (s,a)\right) r_{h,p}^l}{n(s,a)}.$$

Similarly, for every $h \in [H]$ and $(s, a, s') \in \mathcal{S}_h \times \mathcal{A} \times \mathcal{S}_{h+1}$, we also define the individual and aggregate estimates of transition probability as:

$$\hat{\mathbb{P}}_p(s' \mid s,a) := \frac{\sum_{l=1}^{k} \mathbf{1}\left((s_{h,p}^l, a_{h,p}^l, s_{h+1,p}^l) = (s,a,s')\right)}{n_p(s,a)}.$$

$$\hat{\mathbb{P}}(s' \mid s,a) := \frac{\sum_{l=1}^{k}\sum_{p=1}^{M} \mathbf{1}\left((s_{h,p}^l, a_{h,p}^l, s_{h+1,p}^l) = (s,a,s')\right)}{n(s,a)},$$

If $n(s, a) = 0$, we define $\hat{R}(s, a) := 0$ and $\hat{\mathbb{P}}(s' \mid s, a) := \frac{1}{|\mathcal{S}_{h+1}|}$; and if $n_p(s, a) = 0$, we define $\hat{R}_p(s, a) := 0$ and $\hat{\mathbb{P}}(s' \mid s, a) := \frac{1}{|\mathcal{S}_{h+1}|}$. The counts and reward estimates can be maintained by MULTI-TASK-EULER efficiently in an incremental manner.

Constructing value function estimates via optimistic value iteration. For each player p, based on these model parameter estimates, MULTI-TASK-EULER performs optimistic value iteration to compute the value functions estimates for states at all layers (lines 3 to 15). For the terminal layer $H + 1$, $V_{H+1}^\star(\perp) = 0$ trivially, so nothing needs to be done. For earlier layers $h \in [H]$, MULTI-TASK-EULER iteratively builds its value function estimates in a backward fashion. At the time of estimating values for layer h, the algorithm have already obtained optimal value estimates for layer $h + 1$. Based on the Bellman optimality equation (1), MULTI-TASK-EULER estimates $(Q_p^\star(s, a))_{s\in\mathcal{S}_h, a\in\mathcal{A}}$ using model parameter estimates and its estimates of $(V_p^\star(s))_{s\in\mathcal{S}_{h+1}}$, i.e., $(\overline{V}_p(s))_{s\in\mathcal{S}_{h+1}}$ and $(\underline{V}_p(s))_{s\in\mathcal{S}_{h+1}}$. (lines 5 to 12).

Specifically, MULTI-TASK-EULER constructs estimates of $(Q_p^\star(s, a))$ for all $s \in \mathcal{S}_h, a \in \mathcal{A}$ in two different ways. First, it uses the individual estimates of model of player p to construct $\underline{\text{ind-}Q}_p$ and $\overline{\text{ind-}Q}_p$, upper and lower bound estimates of $Q_p^\star$ (lines 8 and 9); this construction is reminiscent of EULER and STRONG-EULER [40, 30], in that if we were to use $\underline{\text{ind-}Q}_p$ and $\overline{\text{ind-}Q}_p$ as our optimal action value function estimate $\overline{Q}$ and $\underline{Q}$, our algorithm becomes individual EULER. Second, it uses the aggregate estimate of model to construct $\underline{\text{agg-}Q}_p$ and $\overline{\text{agg-}Q}_p$, also upper and lower bound estimates of $Q_p^\star$ (lines 6 and 7); this construction is unique to the multitask learning setting, and is our new algorithmic contribution.

To ensure that $\overline{\text{agg-}Q}_p$ and $\overline{\text{ind-}Q}_p$ (resp. $\underline{\text{agg-}Q}_p$ and $\underline{\text{ind-}Q}_p$) are valid upper bounds (resp. lower bounds) of $Q_p^\star$, MULTI-TASK-EULER adds bonus terms ind-$b_p(s, a)$ and agg-$b_p(s, a)$, respectively, in the optimistic value iteration process, to account for estimation error of the model estimates against the true models. Specifically, both bonus terms are composed of three parts:

$$\text{ind-}b_p(s,a) = b_{\text{rw}}(n_p(s,a), 0) + b_{\text{prob}}(\hat{\mathbb{P}}_p(\cdot \mid s,a), n_p(s,a), \overline{V}_p, \underline{V}_p, 0) + b_{\text{str}}(\hat{\mathbb{P}}_p(\cdot \mid s,a), n_p(s,a), \overline{V}_p, \underline{V}_p, 0),$$

$$\text{agg-}b_p(s,a) = b_{\text{rw}}(n(s,a), \epsilon) + b_{\text{prob}}(\hat{\mathbb{P}}(\cdot \mid s,a), n(s,a), \overline{V}_p, \underline{V}_p, \epsilon) + b_{\text{str}}(\hat{\mathbb{P}}(\cdot \mid s,a), n(s,a), \overline{V}_p, \underline{V}_p, \epsilon),$$

where

$$b_{\mathrm{rw}}(n,\kappa) \approx 1 \wedge \sqrt{\frac{L(n)}{n}} + \kappa,$$

$$b_{\mathrm{prob}}(q,n,\overline{V},\underline{V},\kappa) \approx H \wedge \sqrt{\frac{\mathrm{var}_{s'\sim q}\left[\overline{V}(s')\right] L(n)}{n}} + \sqrt{\frac{\mathbb{E}_{s'\sim q}\left[(\overline{V}(s') - \underline{V}(s'))^2\right] L(n)}{n}} + \frac{HL(n)}{n} + \kappa,$$

$$b_{\mathrm{str}}(q,n,\overline{V},\underline{V},\kappa) \approx \sqrt{\frac{S\mathbb{E}_{s'\sim q}\left[(\overline{V}(s') - \underline{V}(s'))^2\right] L(n)}{n}} + \frac{SHL(n)}{n} + \kappa,$$

and $L(n) \approx \ln(MHSAn)$.

The three components in the bonus terms serve for different purposes:

1. The first component accounts for the uncertainty in the reward estimation: with high probability, $\left|\hat{R}_p(s,a) - R_p(s,a)\right| \le b_{\mathrm{rw}}(n_p(s,a),0)$, and $\left|\hat{R}(s,a) - R_p(s,a)\right| \le b_{\mathrm{rw}}(n(s,a),\epsilon)$.
2. The second component accounts for the uncertainty in estimating $(\mathbb{P}_p V_p^\star)(s,a)$: with high probability, $\left|(\hat{\mathbb{P}}_p V_p^\star)(s,a) - (\mathbb{P}_p V_p^\star)(s,a)\right| \le b_{\mathrm{prob}}(\hat{\mathbb{P}}_p(\cdot \mid s,a), n(s,a), \overline{V}_p, \underline{V}_p, 0)$ and $\left|(\hat{\mathbb{P}} V_p^\star)(s,a) - (\mathbb{P}_p V_p^\star)(s,a)\right| \le b_{\mathrm{prob}}(\hat{\mathbb{P}}(\cdot \mid s,a), n(s,a), \overline{V}_p, \underline{V}_p, \epsilon)$.
3. The third component accounts for the lower order terms to ensure strong optimism [30]: with high probability, $\left|(\hat{\mathbb{P}}_p - \mathbb{P}_p)(\overline{V}_p - V_p^\star)(s,a)\right| \le b_{\mathrm{str}}(\hat{\mathbb{P}}_p(\cdot \mid s,a), n(s,a), \overline{V}_p, \underline{V}_p, 0)$, and $\left|(\hat{\mathbb{P}} - \mathbb{P}_p)(\overline{V}_p - V_p^\star)(s,a)\right| \le b_{\mathrm{prob}}(\hat{\mathbb{P}}(\cdot \mid s,a), n(s,a), \overline{V}_p, \underline{V}_p, \epsilon)$.

Based on the above concentration inequalities and the definitions of bonus terms, it can be shown inductively that, with high probability, both $\overline{\text{agg-}Q}_p$ and $\overline{\text{ind-}Q}_p$ (resp. $\underline{\text{agg-}Q}_p$ and $\underline{\text{ind-}Q}_p$) are valid upper bounds (resp. lower bounds) of $Q_p^\star$.

Finally, observe that $Q_p^\star$ trivially lies in $[0,H]$. By taking intersections of all confidence bounds of $Q_p^\star$ it has obtained, MULTI-TASK-EULER constructs its final upper and lower bound estimates for $Q_p^\star$, $\overline{Q}_p$ and $\underline{Q}_p$, respectively for $s \in \mathcal{S}_h$ (line 11 to 12). A similar idea on using data from multiple sources to construct confidence intervals and guide explorations has been proposed by [31] for multi-task linear contextual bandits. Using the relationship between the optimal value $V_p^\star$ and and optimal action value $Q_p^\star$, MULTI-TASK-EULER also constructs upper and lower bound estimates for $V_p^\star$, $\overline{V}_p$ and $\underline{V}_p$, respectively for $s \in \mathcal{S}_h$ (line 15).

Executing optimistic policies. For each episode k and each player p, its optimal action-value function upper bound estimate $\overline{Q}_p$ induces a greedy policy $\pi^k(p) : s \mapsto \mathrm{argmax}_{a\in\mathcal{A}} \overline{Q}_p(s,a)$ (line 14); the player then executes this policy at this episode to collect a new trajectory and use this to update its individual model parameter estimates. After all players finish their episode k, the algorithm also updates its aggregate model parameter estimates (line 16 to 19) and continues to the next episode.

4 Performance guarantees

Before stating the guarantees of Algorithm 1, we define an instance-dependent complexity measure that characterizes the amenability of information sharing.

Definition 3. *The set of subpar state-action pairs is defined as:*

$$\mathcal{I}_\epsilon := \left\{(s,a) \in \mathcal{S}\times\mathcal{A} : \exists p \in [M], \mathrm{gap}_p(s,a) \ge 96H\epsilon\right\},$$

where we recall that $\mathrm{gap}_p(s,a) = V_p^\star(s) - Q_p^\star(s,a)$.

Definition 3 generalizes the notion of subpar arms defined for multi-task multi-armed bandit learning [37] in two ways: first, it is with regards to state-action pairs as opposed to actions only; second, in RL, suboptimality gaps depend on both immediate reward and subsequent long-term return.

To ease our later presentation, we also present the following lemma.

Lemma 4. *For any $(s,a) \in \mathcal{I}_\epsilon$, we have that: (1) for all $p \in [M]$, $(s,a) \notin Z_{p,\mathrm{opt}}$, where we recall that $Z_{p,\mathrm{opt}} = \left\{(s,a) : \mathrm{gap}_p(s,a) = 0\right\}$ is the set of optimal state-action pairs with respect to p; (2) for all $p, q \in [M]$, $\mathrm{gap}_p(s,a) \geq \frac{1}{2}\mathrm{gap}_q(s,a)$.*

The lemma follows directly from Lemma 2; its proof can be found in the Appendix along with proofs of the following theorems. Item 1 implies that any subpar state action pair is suboptimal for all players. In other words, for every player p, the state-action space $\mathcal{S} \times \mathcal{A}$ can be partitioned to three disjoint sets: $\mathcal{I}_\epsilon, Z_{p,\mathrm{opt}}, (\mathcal{I}_\epsilon \cup Z_{p,\mathrm{opt}})^C$. Item 2 implies that for subpar (s,a), its suboptimal gaps with respect to all players are within a constant of each other.

4.1 Upper bounds

Equipped with the above definitions, we are now ready to present the performance guarantees of Algorithm 1. We first present a gap-independent collective regret bound of MULTI-TASK-EULER.

Theorem 5 (Gap-independent bound)**.** MULTI-TASK-EULER *satisfies that*

$$\mathrm{Reg}(K) \lesssim M\sqrt{H^2|\mathcal{I}_\epsilon^C|K} + \sqrt{MH^2|\mathcal{I}_\epsilon|K} + MS^2AH^3(\ln(MSAHK))^2.$$

We again compare this regret upper bound with individual STRONG-EULER's gap independent regret bound. Recall that STRONG-EULER guarantees that

$$\mathrm{Reg}(K) \lesssim M\sqrt{H^2SAK} + MS^2AH^3(\ln(MSAHK))^2.$$

We again focus on the comparison on the leading terms, i.e., the $\sqrt{K}$ terms. As $M\sqrt{H^2SAK} \eqsim M\sqrt{H^2|\mathcal{I}_\epsilon|\,K} + M\sqrt{H^2\left|\mathcal{I}_\epsilon^C\right|\,K}$, we see that an improvement in the collective regret bound comes from the contributions from the subpar state-action pairs: the $M\sqrt{H^2|\mathcal{I}_\epsilon|K}$ term is reduced to $\sqrt{MH^2|\mathcal{I}_\epsilon|K}$, which is an improvement by a factor of $\sqrt{M}$. Moreover, if $\left|\mathcal{I}_\epsilon^C\right| \ll SA$ and $M \gg 1$, MULTI-TASK-EULER provides a regret bound of lower order than individual EULER.

We next present a gap-dependent upper bound on its collective regret.

Theorem 6 (Gap-dependent upper bound)**.** MULTI-TASK-EULER *satisfies*

$$\mathrm{Reg}(K) \lesssim \ln(MSAHK)\left(\sum_{p\in[M]}\left(\sum_{(s,a)\in Z_{p,\mathrm{opt}}}\frac{H^3}{\mathrm{gap}_{p,\min}} + \sum_{(s,a)\in(\mathcal{I}_\epsilon\cup Z_{p,\mathrm{opt}})^C}\frac{H^3}{\mathrm{gap}_p(s,a)}\right) + \sum_{(s,a)\in\mathcal{I}_\epsilon}\frac{H^3}{\min_p \mathrm{gap}_p(s,a)}\right) + \ln(MSAHK)\cdot MS^2AH^3\ln\frac{MH}{\mathrm{gap}_{\min}},$$

where we recall that $\mathrm{gap}_{p,\min} = \min_{(s,a):\mathrm{gap}_p(s,a)>0}\mathrm{gap}_p(s,a)$*, and* $\mathrm{gap}_{\min} = \min_p \mathrm{gap}_{p,\min}$.

Comparing this regret bound with the regret bound obtained by the individual STRONG-EULER baseline, recall that by summing over the regret guarantees of STRONG-EULER for all players, we deduce that individual STRONG-EULER guarantees a collective regret bound of

$$\mathrm{Reg}(K) \lesssim \ln(MSAHK)\left(\sum_{p\in[M]}\left(\sum_{(s,a)\in Z_{p,\mathrm{opt}}}\frac{H^3}{\mathrm{gap}_{p,\min}} + \sum_{(s,a)\in(\mathcal{I}_\epsilon\cup Z_{p,\mathrm{opt}})^C}\frac{H^3}{\mathrm{gap}_p(s,a)}\right) + \sum_{(s,a)\in\mathcal{I}_\epsilon}\sum_{p\in[M]}\frac{H^3}{\mathrm{gap}_p(s,a)}\right) + \ln(MSAHK)\cdot MS^2AH^3\ln\frac{MH}{\mathrm{gap}_{\min}}.$$

We focus on comparing the leading terms, i.e., the terms that have polynomial dependences on the suboptimality gaps in the above two bounds. It can be seen that an improvement in the regret bound by MULTI-TASK-EULER comes from the contributions from the subpar state-action pairs: for each $(s,a) \in \mathcal{I}_\epsilon$, the regret bound is reduced from $\sum_{p\in[M]} \frac{H^3}{\mathrm{gap}_p(s,a)}$ to $\frac{H^3}{\min_p \mathrm{gap}_p(s,a)}$, which is an improvement by a factor of $O(\frac{1}{M})$. Recent work of [38] has shown that in the single-task setting, it is possible to replace $\sum_{(s,a)\in Z_{p,\mathrm{opt}}} \frac{H^3}{\mathrm{gap}_{p,\min}}$ with a sharper problem-dependent complexity term that depends on the multiplicity of optimal state-action pairs. We leave improving the guarantee of Theorem 6 in a similar manner as an interesting open problem.

4.2 Lower bounds

To complement the above upper bounds, we now present gap-dependent and gap-independent regret lower bounds that also depends on our subpar state-action pair notion. Our lower bounds are inspired by regret bounds for episodic RL [30, 6] and multi-task bandits [37].

Theorem 7 (Gap-independent lower bound)**.** *For any $A \geq 2$, $H \geq 2$, $S \geq \Omega(H)$, $K \geq A$, $M \in \mathbb{N}$, and $l, l^C \in \mathbb{N}$ such that $l + l^C = SA$ and $l \leq SA - \Theta(S + HA)$, there exists some ϵ such that for any algorithm* Alg*, there exists an ϵ-MPERL problem instance with S states, A actions, M players and an episode length of H such that $\left|\mathcal{I}_{\Theta(\frac{\epsilon}{H})}\right| \geq l$, and*

$$\mathrm{Reg}_{\mathrm{Alg}}(K) \geq \Omega\left(M\sqrt{H^2 l^C K} + \sqrt{MH^2 l K}\right).$$

We also present a gap-dependent lower bound. Before that, we first formally define the notion of sublinear regret algorithms. We say that an algorithm Alg is a sublinear regret algorithm for the ϵ-MPERL problem if there exists some $C > 0$ and $\alpha < 1$ such that $\mathrm{Reg}_{\mathrm{Alg}}(K) \leq CT^\alpha$.

Theorem 8 (Gap-dependent lower bound)**.** *Fix $\epsilon \geq 0$. For any $S \in \mathbb{N}$, $A \geq 2$, $H \geq 2$, $M \in \mathbb{N}$, such that $S \geq 2(H-1)$, let $S_1 = S - 2(H-1)$; and let $\left\{\Delta_{s,a,p}\right\}_{(s,a,p)\in[S_1]\times[A]\times[M]}$ be any set of values such that (1) each $\Delta_{s,a,p} \in [0, H/8]$, (2) for every $(s,p) \in [S_1] \times [M]$, there exists at least one action $a \in [A]$ such that $\Delta_{s,a,p} = 0$, and (3) for every $(s,a) \in [S_1] \times [A]$ and $p, q \in [M]$, $\left|\Delta_{s,a,p} - \Delta_{s,a,q}\right| \leq \epsilon/4$. There exists an ϵ-MPERL problem instance with S states, A actions, M players and an episode length of H, such that $\mathcal{S}_1 = [S_1]$, $|\mathcal{S}_h| = 2$ for all $h \geq 2$, and*

$$\mathrm{gap}_p(s,a) = \Delta_{s,a,p}, \quad \forall (s,a,p) \in [S_1] \times [A] \times [M];$$

for this problem instance, any sublinear regret algorithm Alg *for the ϵ-MPERL problem must have regret at least*

$$\mathrm{Reg}_{\mathrm{Alg}}(K) \geq \Omega\left(\ln K \left(\sum_{p\in[M]} \sum_{\substack{(s,a)\in\mathcal{I}^C_{\Theta(\epsilon/H)}:\\ \mathrm{gap}_p(s,a)>0}} \frac{H^2}{\mathrm{gap}_p(s,a)} + \sum_{(s,a)\in\mathcal{I}_{\Theta(\epsilon/H)}} \frac{H^2}{\min_p \mathrm{gap}_p(s,a)}\right)\right).$$

Comparing the lower bounds with MULTI-TASK-EULER's regret upper bounds in Theorems 5 and 6, we can see that the upper and lower bounds nearly match for any constant H. When H is large, the key difference between the upper and lower bounds is that the former are in terms of $\mathcal{I}_\epsilon$, while the latter are in terms of $\mathcal{I}_{\Theta(\frac{\epsilon}{H})}$. We conjecture that our upper bounds can be improved by replacing $\mathcal{I}_\epsilon$ with $\mathcal{I}_{\Theta(\frac{\epsilon}{H})}$—our analysis uses a clipping trick similar to [30], which may be the reason for a suboptimal dependence on H. We leave closing this gap as an open question.

5 Related Work

Regret minimization for MDPs. Our work belongs to the literature of regret minimization for MDPs [e.g., 3, 14, 6, 2, 7, 15, 8, 40, 30, 43, 39, 38]. In the episodic setting, [2, 8, 40, 30, 43] achieve minimax $\sqrt{H^2SAK}$ regret bounds for general stationary MDPs. Furthermore, the EULER

algorithm [40] achieves adaptive problem-dependent regret guarantees when the total reward within an episode is small or when the environmental norm of the MDP is small. [30] refines EULER, proposing STRONG-EULER that provides more fine-grained gap-dependent $O(\log K)$ regret guarantees. [39, 38] show that the optimistic Q-learning algorithm [15] and its variants can also achieve gap-dependent logarithmic regret guarantees. Remarkably, [38] achieves a regret bound that improves over that of [30], in that it replaces the dependence on the number of optimal state-action pairs with the number of non-unique state-action pairs.

Transfer and lifelong learning for RL. A considerable portion of related works concerns transfer learning for RL tasks (see [34, 19, 44] for surveys from different angles), and many studies investigate a batch setting: given some source tasks and target tasks, transfer learning agents have access to batch data collected for the source tasks (and sometimes for the target tasks as well). In this setting, model-based approaches have been explored in [e.g., 33]; theoretical guarantees for transfer of samples across tasks have been established in [e.g., 20, 35]. Similarly, sequential transfer has been studied under the framework of lifelong RL in [e.g., 32, 1, 11, 18]—in this setting, an agent faces a sequence of RL tasks and aims to take advantage of knowledge gained from previous tasks for better performance in future tasks; in particular, analyses on the sample complexity of transfer learning algorithms are presented in [4, 24] under the assumption that an upper bound on the total number of unique (and well-separated) RL tasks is known. We note that, in contrast, we study an online setting in which no prior data are available and multiple RL tasks are learned concurrently by RL agents.

Concurrent RL. Data sharing between multiple RL agents that learn concurrently has also been investigated. In [e.g., 16, 29, 12, 10], a group of agents interact in parallel with *identical* environments. Another setting is studied in [12], in which agents solve different RL tasks (MDPs); however, similar to [4, 24], it is assumed that there is a finite number of unique tasks, and different tasks are well-separated, i.e., there is a minimum gap. In this work, we assume that players face similar but not necessarily identical MDPs, and we do not assume a minimum gap. [13] studies multi-task RL with linear function approximation with representation transfer, where it is assumed that the optimal value functions of all tasks are from a low dimensional linear subspace. Our setting and results are the most similar to [26], which studies concurrent exploration in similar MDPs with continuous states in the PAC setting; however, their PAC guarantee does not hold for target error rate arbitrarily close to zero. In contrast, our algorithm has a fall-back guarantee, in that it always has a sublinear regret.

Other related topics and models. In many multi-agent RL models [41, 25], a set of learning agents interact an environment with shared global states; in particular, [42] studies the setting with heterogeneous reward distributions, and provides convergence guarantees for two policy gradient-based algorithms. In contrast, in our setting, our learning agents interact with separate environments. Multi-agent bandits with similar, heterogeneous reward distributions are investigated in [37]; herein, we generalize the multi-armed bandit setting to tabular, episodic MDPs.

6 Conclusion and Future Directions

In this paper, we generalize the multi-task bandit learning framework in [37] and formulate a multi-task concurrent RL problem, in which tasks are similar but not necessarily identical. We provide a provably efficient model-based algorithm that takes advantage of knowledge transfer between different tasks. Our instance-dependent regret upper and lower bounds formalize an intuition that subpar state-action pairs are amenable of information sharing among tasks.

There still remain gaps between our upper and lower bounds which can be closed by either a finer analysis or a better algorithm: first, the dependence on $\mathcal{I}_\epsilon$ in the upper bound does not match the dependence of $\mathcal{I}_{\Theta(\epsilon/H)}$ in the lower bound when H is large; second, the gap-dependent upper bound has $O(H^3)$ dependence, whereas the gap-dependent lower bound only has $\Omega(H^2)$ dependence; third, the additive dependence on the number of optimal state-action pairs can potentially be removed by new algorithmic ideas [38]. Another interesting future direction is to consider more general parameter transfer for online RL, for example, in the context of function approximation.

Societal Impacts. One major limitation of our algorithm is that it only provides guarantees in tabular MDPs. To avoid unintended consequences when deployed in the real world, another major open problem is to ensure safety of our algorithm under model misspecification.

References

[1] David Abel, Yuu Jinnai, Sophie Yue Guo, George Konidaris, and Michael Littman. Policy and value transfer in lifelong reinforcement learning. In *International Conference on Machine Learning*, pages 20–29. PMLR, 2018.

[2] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

[3] Peter L Bartlett and Ambuj Tewari. Regal: a regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 35–42, 2009.

[4] Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 122–131, 2013.

[5] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[6] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2818–2826, 2015.

[7] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: uniform pac bounds for episodic reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5717–5727, 2017.

[8] Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pages 1507–1516. PMLR, 2019.

[9] Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing knowledge in multi-task deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

[10] Maria Dimakopoulou and Benjamin Van Roy. Coordinated exploration in concurrent reinforcement learning. In *International Conference on Machine Learning*, pages 1271–1279. PMLR, 2018.

[11] Francisco M Garcia and Philip S Thomas. A meta-mdp approach to exploration for lifelong reinforcement learning. *arXiv preprint arXiv:1902.00843*, 2019.

[12] Zhaohan Guo and Emma Brunskill. Concurrent pac rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[13] Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. Near-optimal representation learning for linear bandits and linear rl. *arXiv preprint arXiv:2102.04132*, 2021.

[14] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

[15] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4868–4878, 2018.

[16] R Matthew Kretchmar. Parallel reinforcement learning. In *The 6th World Conference on Systemics, Cybernetics, and Informatics*. Citeseer, 2002.

[17] Alyssa Kubota, Emma IC Peterson, Vaishali Rajendren, Hadas Kress-Gazit, and Laurel D Riek. Jessie: Synthesizing social robot behaviors for personalized neurorehabilitation and beyond. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 121–130, 2020.

[18] Nicholas C Landolfi, Garrett Thomas, and Tengyu Ma. A model-based approach for sample-efficient multi-task reinforcement learning. *arXiv preprint arXiv:1907.04964*, 2019.

[19] Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pages 143–173. Springer, 2012.

[20] Alessandro Lazaric and Marcello Restelli. Transfer from multiple mdps. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[21] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 544–551, 2008.

[22] Xinle Liang, Yang Liu, Tianjian Chen, Ming Liu, and Qiang Yang. Federated transfer reinforcement learning for autonomous driving. *arXiv preprint arXiv:1910.06001*, 2019.

[23] Boyi Liu, Lujia Wang, and Ming Liu. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters*, 4(4):4555–4562, 2019.

[24] Yao Liu, Zhaohan Guo, and Emma Brunskill. Pac continuous state online multitask reinforcement learning with identification. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, page 438–446, 2016.

[25] Afshin OroojlooyJadid and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963*, 2019.

[26] Jason Pazis and Ronald Parr. Efficient pac-optimal exploration in concurrent, continuous state mdps with delayed updates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[27] Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, 2005.

[28] David Silver, Leonard Newnham, David Barker, Suzanne Weller, and Jason McFall. Concurrent reinforcement learning from customer interactions. In *International conference on machine learning*, pages 924–932. PMLR, 2013.

[29] David Silver, Leonard Newnham, David Barker, Suzanne Weller, and Jason McFall. Concurrent reinforcement learning from customer interactions. In *International conference on machine learning*, pages 924–932. PMLR, 2013.

[30] Max Simchowitz and Kevin Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. *arXiv preprint arXiv:1905.03814*, 2019.

[31] Marta Soare, Ouais Alsharif, Alessandro Lazaric, and Joelle Pineau. Multi-task linear bandits. *NIPS2014 Workshop on Transfer and Multi-task Learning : Theory meets Practice*, 2014.

[32] Fumihide Tanaka and Masayuki Yamamura. Multitask reinforcement learning on the distribution of mdps. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 3, pages 1108–1113. IEEE, 2003.

[33] Matthew E Taylor, Nicholas K Jong, and Peter Stone. Transferring instances for model-based reinforcement learning. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 488–505. Springer, 2008.

[34] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

[35] Andrea Tirinzoni, Mattia Salvini, and Marcello Restelli. Transfer of samples in policy search via multiple importance sampling. In *International Conference on Machine Learning*, pages 6264–6274. PMLR, 2019.

[36] Konstantinos Tsiakas, Cheryl Abellanoza, and Fillia Makedon. Interactive learning and adaptation for robot assisted therapy for people with dementia. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pages 1–4, 2016.

[37] Zhi Wang, Chicheng Zhang, Manish Kumar Singh, Laurel Riek, and Kamalika Chaudhuri. Multitask bandit learning through heterogeneous feedback aggregation. In *International Conference on Artificial Intelligence and Statistics*, pages 1531–1539. PMLR, 2021.

[38] Haike Xu, Tengyu Ma, and Simon S Du. Fine-grained gap-dependent bounds for tabular mdps via adaptive multi-step bootstrap. *arXiv preprint arXiv:2102.04692*, 2021.

[39] Kunhe Yang, Lin Yang, and Simon Du. Q-learning with logarithmic regret. In *International Conference on Artificial Intelligence and Statistics*, pages 1576–1584. PMLR, 2021.

[40] Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pages 7304–7312. PMLR, 2019.

[41] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

[42] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881. PMLR, 2018.

[43] Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learningvia reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33, 2020.

[44] Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020.

[45] Hankz Hankui Zhuo, Wenfeng Feng, Qian Xu, Qiang Yang, and Yufeng Lin. Federated reinforcement learning. *arXiv preprint arXiv:1901.08277*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We study multi-task learning for regret minimization on episodic tabular MDPs, formulating a new notion of task similarity and design and analyze provably efficient algorithms in this setting. See the second half of Section 1 for a summary of our contributions.
   (b) Did you describe the limitations of your work? [Yes] We have discussed the limitations of our work in Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We have discussed potential negative societal impacts of our work in Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sections 2 and 4 for our assumptions of the theoretical results.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix for the full proofs.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]