Robust Knowledge Transfer in Tiered RL

Jiawei Huang
Department of Computer Science
ETH Zurich
jiawei.huang@inf.ethz.ch

Niao He
Department of Computer Science
ETH Zurich
jiawei.huang@inf.ethz.ch

Abstract

In this paper, we study the Tiered Reinforcement Learning setting, a parallel transfer learning framework, where the goal is to transfer knowledge from the low-tier (source) task to the high-tier (target) task to reduce the exploration risk of the latter while solving the two tasks in parallel. Unlike previous work, we do not assume the low-tier and high-tier tasks share the same dynamics or reward functions, and focus on robust knowledge transfer without prior knowledge on the task similarity. We identify a natural and necessary condition called the "Optimal Value Dominance" for our objective. Under this condition, we propose novel online learning algorithms such that, for the high-tier task, it can achieve constant regret on partial states depending on the task similarity and retain near-optimal regret when the two tasks are dissimilar, while for the low-tier task, it can keep near-optimal without making sacrifice. Moreover, we further study the setting with multiple low-tier tasks, and propose a novel transfer source selection mechanism, which can ensemble the information from all low-tier tasks and allow provable benefits on a much larger state-action space.

1 Introduction

Comparing with individual learning from scratch, transferring knowledge from other similar tasks or side information has been proven to be an effective way to reduce the exploration risk and improve sample efficiency in Reinforcement Learning (RL). Multi-Task RL (MT-RL) [29] and Transfer RL [26, 18, 37] are two mainstream knowledge transfer frameworks; however, both are subject to limitations when dealing with real-world scenarios. MT-RL studies the setting where a set of similar tasks are solved concurrently, and the main objective is to accelerate the learning by sharing information of all tasks together. However, in practice, in many MT-RL scenarios, the tasks are not equally important and we are more interested in the performance of certain tasks. For example, in robot learning, a few robots are more valuable and hard to fix, while the others are cheaper or just simulators. Most existing works on MT-RL treat all tasks equally and focus primarily on the reduction of the total regret of all tasks as a whole [3, 8, 36, 11], with no guarantee of improving a particular task. In contrast, transfer RL distinguishes the priority of different tasks by categorizing them into source and target tasks and aims at transferring the knowledge from source tasks (or some side information like value predictors) to facilitate the learning of target tasks [21, 27, 9, 10]. However, a key assumption in transfer RL is that the source task is completely solved before the learning of the target task, and this is not always practical. For example, in some sim-to-real domain, the source task simulator may require a long time to solve [5], and in some user-interaction scenarios [13], the source and target tasks refer to different user groups and they have to be served simultaneously. In these cases, it's more reasonable to solve the source and target tasks in parallel and transfer the information immediately once available.

Recently, [13] proposed a new "parallel knowledge transfer" framework, called Tiered RL, which is promising to fill the gap. Tiered RL considers the case when a source task M_{Lo} and a target task

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

https://doi.org/10.52202/075280-2268

M_{Hi} are learned in parallel, by two separate algorithms Alg^{Lo} and Alg^{Hi}, and its goal is to reduce the exploration risk and regret in learning M_{Hi} by leveraging knowledge transfer from M_{Lo} to M_{Hi}. [13] showed that under the strong assumption that $M_{\text{Lo}} = M_{\text{Hi}}$, it's possible to achieve constant regret in learning M_{Hi} while keeping regret in M_{Lo} optimal in gap-dependent setting. Yet, their algorithm based on Pessimistic Value Iteration (PVI) [16] can be hardly applied when the assumption $M_{\text{Lo}} = M_{\text{Hi}}$ breaks, given its pure exploitation nature, making their result very restrictive.

In this paper, we study the general Tiered RL setting *without prior knowledge* about how similar the tasks are [1]. The key question we would like to address is: **Can we design algorithms s.t.: (1) Regret in M_{Lo} keeps near-optimal; (2) Regret in M_{Hi} achieves provable benefits when M_{Lo} and M_{Hi} are similar while retaining near-optimal otherwise?** Note for Alg^{Lo}, we expect it to achieve near-optimal regret bounds, which is reasonable since the source task is often important and our results still hold if relaxing it. As for Alg^{Hi}, we expect it to be *robust*, i.e., it can adaptively exploit from M_{Lo} if it is close to M_{Hi}, while avoiding negative transfer in other cases. Notably, our setting strictly generalizes [13] and is much more challenging, for balancing the exploitation from M_{Lo} and exploration from M_{Hi} without prior knowledge of task similarity. We give positive answers to the above question in this paper and demonstrate provable benefits with robust knowledge transfer for Tiered RL framework. Below, we summarize our main contributions in three aspects.

Our first contribution is to identify essential conditions and notions about when and how our objective is achievable. We first provide a mild condition called *Optimal Value Dominance* (OVD), and in Sec. 3, we justify its necessity to our objective by a lower bound result. Our lower bound holds even if M_{Lo} is fully known to the learner, and therefore, it also justifies the necessity of similar assumptions in previous transfer RL literatures [9, 10]. Besides, we introduce the notion of *transferable states* to characterize states on which M_{Hi} is expected to achieve benefits by transferring knowledge from M_{Lo}. We believe those findings also provide useful insights for further works.

As **our second contribution**, in Sec. 4, we propose novel algorithms for Tiered Multi-Armed Bandit (MAB) and Tiered RL, which can achieve robust parallel transfer by balancing between pessimism-based exploitation from M_{Lo} and optimism-based online learning in M_{Hi}. Depending on the similarity between M_{Lo} and M_{Hi}, our algorithms can enjoy constant regret on a proportion of state-action pairs or even on the entire M_{Hi} by leveraging information from M_{Lo}, while timely interrupting negative transfer and retaining near-optimal regret on states dissimilar between two tasks. Moreover, in the bandit setting, our result implies a strictly improved regret bound when $M_{\text{Hi}} = M_{\text{Lo}}$, compared with previous results under the same setting [22, 13].

Beyond the single low-tier task setting, in many real-world scenarios, it's reasonable to assume there are multiple different low-tier tasks $\mathcal{M}_{\text{Lo}} = \{M_{\text{Lo},w}\}_{w=1}^{W}$ available. As **our third contribution**, in Sec. 5, we extend our algorithm to this setting with a new source task selection mechanism. By novel techniques, we show that, even if each $M_{\text{Lo},w}$ may only share similarity with M_{Hi} on a small portion of states, we are able to ensemble the information from each source task together to achieve constant regret on a much larger state-action space "stitched" from the transferable state-action set in each individual $M_{\text{Lo},w}$, at the expense of an additional $\log W$ factor in regret. Besides, our algorithm is still robust to model difference and retains near-optimal regret in general. Although we only study the Tiered RL setting in this paper, we believe our task selection strategy can be applied to standard transfer RL setting [9, 10] when multiple (partially correct) side information or value predictors are provided, which is an interesting direction for future work. Finally, we conduct experiments in toy examples to verify our theoretical results.

1.1 Closely Related Work

For the lack of space, we only discuss closely related work here and defer the rest to Appx. A.2. The most related to us is the Tiered RL framework [13], which was originally motivated by Tiered structure in user-oriented applications. However, they only studied the case when $M_{\text{Hi}} = M_{\text{Lo}}$, which limits the practicability of their algorithms. Although there is a sequence of work studying parallel transfer learning in multi-agent system [25, 19], but they mainly focused on heuristic empirical algorithms and did not have theoretical guarantees.

Transfer RL [37], compared to learning from scratch, can reduces exploration risk of target task by leveraging information in similar source tasks or side information [21, 27, 9, 10]. Comparing with

[1] We defer the detailed framework to Appx. A.1 for completeness, which is the same as Fr. 1 in [13].

https://doi.org/10.52202/075280-2268

transfer RL setting, our parallel transfer setting has some additional challenges. Firstly, M_{Hi} can only leverage estimated value/model/optimal policy from M_{Lo} with uncertainty, which implies we need additional efforts to control failure events with non-zero probability comparing with normal transfer RL setting. Secondly, the constraints on the optimality of $\text{Regret}_K(M_{\text{Lo}})$, although reasonable, restrict the transferable information because in M_{Lo} estimation uncertainty can only be controlled on those states frequently visited. Moreover, none of these previous work studies how to leverage multiple partially correct side information like what we did in Sec. 5. In MT-RL setting, the benefits of leveraging information gathered from other tasks has been observed from both empirical and theoretical works [31, 3, 8, 24, 36, 11]. But MTRL treats each task equally, and the reduction of total regret over all tasks does not directly imply benefits achieved in a particular task.

2 Preliminary and Problem Formulation

Tiered Stochastic MAB and Tiered Episodic Tabular RL In Tiered MAB setting, we consider a low-tier task M_{Lo} and a high-tier task M_{Hi} sharing the arm/action space $\mathcal{A} = \{1, 2, ..., A\}$. By pulling the arm $i \in [A]$ in M_{Lo} or M_{Hi}, the agent can observe a random variable $r_{\text{Lo}}(i)$ or $r_{\text{Hi}}(i) \in [0, 1]$. We will use $\mu_{\text{Lo}}(i) = \mathbb{E}[r_{\text{Lo}}(i)]$ and $\mu_{\text{Hi}}(i) = \mathbb{E}[r_{\text{Hi}}(i)]$ to denote the expected return of the i-th arm in M_{Lo} and M_{Hi}, respectively, and note that it's possible that $\mu_{\text{Lo}}(i) \neq \mu_{\text{Hi}}(i)$.

For Tiered RL, we assume that two tasks $M_{\text{Lo}} = \{\mathcal{S}, \mathcal{A}, H, \mathbb{P}_{\text{Lo}}, r_{\text{Lo}}\}$ and $M_{\text{Hi}} = \{\mathcal{S}, \mathcal{A}, H, \mathbb{P}_{\text{Hi}}, r_{\text{Hi}}\}$ share the finite state $\mathcal{S}$ and action space $\mathcal{A}$ across episode length H (i.e. $\mathcal{S}_h = \mathcal{S}, \mathcal{A}_h = \mathcal{A}$ for any $h \in [H]$), but may have different time-dependent transition and reward functions $\mathbb{P}_{\text{Lo}} = \{\mathbb{P}_{\text{Lo},h}\}_{h=1}^H, r_{\text{Lo}} = \{r_{\text{Lo},h}\}_{h=1}^H$ and $\mathbb{P}_{\text{Hi}} = \{\mathbb{P}_{\text{Hi},h}\}_{h=1}^H, r_{\text{Hi}} = \{r_{\text{Hi},h}\}_{h=1}^H$. W.l.o.g., we assume the initial state s_1 is fixed, and the reward functions r_{Lo} and r_{Hi} are deterministic and bounded by $[0, 1]$. In episodic MDPs, we study the time-dependent policy specified as $\pi := \{\pi_1, ..., \pi_H\}$ with $\pi_h : \mathcal{S}_h \to \Delta(\mathcal{A}_h)$ for all $h \in [H]$, where $\Delta(\mathcal{A}_h)$ denotes the probability simplex over the action space. With a slight abuse of notation, when π_h is a deterministic policy, we use $\pi_h : \mathcal{S}_h \to \mathcal{A}_h$ to denote the deterministic mapping. Besides, we use $Q_h^\pi(s, a) = \mathbb{E}[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'})|s_h = s, a_h = a, \pi]$, $V_h^\pi(s) = \mathbb{E}_{a\sim\pi}[Q_h^\pi(s, a)]$ to denote the value function for π at step $h \in [H]$, and denote $d_h^\pi(\cdot) := \Pr(s_h = \cdot|\pi)$ and $d_h^\pi(\cdot, \cdot) := \Pr(s_h = \cdot, a_h = \cdot|\pi)$ as the state and state-action occupancy w.r.t. policy π. We use π^* to denote the optimal policy, and V_h^*, Q_h^*, d_h^* as a short note when $\pi = \pi^*$. To avoid confusion, we will specify the policy and value functions in M_{Lo} and M_{Hi} by Lo and Hi in subscription, respectively. For example, in M_{Lo} we have $\pi_{\text{Lo}} := \{\pi_{\text{Lo},1}, ..., \pi_{\text{Lo},H}\}$, $Q_{\text{Lo},h}^{\pi_{\text{Lo}}}/V_{\text{Lo},h}^{\pi_{\text{Lo}}}$ and $Q_{\text{Lo},h}^*/V_{\text{Lo},h}^*$, $d_{\text{Lo},h}^{\pi_{\text{Lo}}}, d_{\text{Lo},h}^*$, and similarly for M_{Hi}.

Gap-Dependent Setting Throughout, we focus on gap-dependent setting [17, 23, 34, 7]. Below we introduce the notion of gap for M_{Lo} as an example, and those for M_{Hi} follows similarly. In MAB case, the gap in M_{Lo} w.r.t. arm i is defined as $\Delta_{\text{Lo}}(i) := \max_{j\in[A]} \mu_{\text{Lo}}(j) - \mu_{\text{Lo}}(i), \forall i \in [A]$, and for tabular RL setting, we have $\Delta_{\text{Lo}}(s_h, a_h) := V_{\text{Lo},h}^*(s_h) - Q_{\text{Lo},h}^*(s_h, a_h), \forall h \in [H], s_h \in \mathcal{S}_h, a_h \in \mathcal{A}_h$. We use $\Delta_{\text{Lo,min}}$ to refer to the minimal gap such that $\Delta(s_h, a_h) \geq \Delta_{\text{Lo,min}}$ for all non-optimal actions a_h, and use $\Delta_{\min} := \min\{\Delta_{\text{Lo,min}}, \Delta_{\text{Hi,min}}\}$ to denote the minimal gap over two tasks. In the gap-dependent setting, we assume $\Delta_{\min} > 0$.

Knowledge Transfer from Multiple Low-Tier Tasks In this case, we assume there are $W > 1$ different source tasks $\mathcal{M}_{\text{Lo}} = \{M_{\text{Lo},w}\}_{w=1}^W$ and all the tasks share the same state and action spaces but may have different transition and reward function. We defer the extended framework for this setting to Appx. A. We specify the task index $w \in [W]$ in sub-scription to distinguish the notation for different source tasks (e.g. $\mathbb{P}_{\text{Lo},w,h}, Q_{\text{Lo},w,h}^{(\cdot)}$). Moreover, we define $\Delta_{\min} := \min\{\Delta_{\text{Lo},1,\text{min}}, ..., \Delta_{\text{Lo},W,\text{min}}, \Delta_{\text{Hi,min}}\}$. For convenience, in the rest of the paper, we use TRL-MST (Tiered RL with Multiple Source Task) as a short abbreviation for this setting.

Performance Measure We use Pseudo-Regret as performance measure: $\text{Regret}_K(M_{\text{Lo}}) := \mathbb{E}\left[\sum_{k=1}^K V_1^*(s_1) - V_1^{\pi_{\text{Lo}}^k}(s_1)\right]$; $\text{Regret}_K(M_{\text{Hi}}) := \mathbb{E}\left[\sum_{k=1}^K V_1^*(s_1) - V_1^{\pi_{\text{Hi}}^k}(s_1)\right]$, where K is the number of iterations, $\{\pi_{\text{Lo}}^k\}_{k=1}^K$ and $\{\pi_{\text{Hi}}^k\}_{k=1}^K$ are generated by the algorithms.

Frequently Used Notations We denote $[n] = \{1, 2, ..., n\}$. Given a transition matrix $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and a function $V : \mathcal{S} \to \mathbb{R}$, we use $\mathbb{P}V(s_h, a_h)$ as a short note of $\mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[V(s')]$. We will use $i_{\text{Lo}}^*/i_{\text{Hi}}^*$ to denote the optimal arm in bandit setting, and $\pi_{\text{Lo}}^*/\pi_{\text{Hi}}^*$ denotes the optimal policy in RL setting. In TRL-MST setting, we use $i_{\text{Lo},w}^*/\pi_{\text{Lo},w}^*$ to distinguish different source tasks.

52075 https://doi.org/10.52202/075280-2268

2.1 Assumptions and Characterization of Transferable States

Throughout the paper, we make several assumptions. The first one is the uniqueness of the optimal policy, which is common in the literature [22, 4].

Assumption A. Both M_{Lo} (or $\{M_{\text{Lo},w}\}_{w=1}^{W}$) and M_{Hi} have unique optimal arms/policies.

Next, we introduce a new concept called "Optimal Value Dominance" (OVD for short), which says that for each state (or at least those states reachable by optimal policy in M_{Lo}), the optimal value of M_{Lo} is an approximate overestimation for the optimal value of M_{Hi}. In Sec. 3, we will use a lower bound to show such a condition is necessary to attain the robust transfer objective.

Assumption B. In single source task setting, we assume M_{Lo} has Optimal Value Dominance (OVD) over M_{Hi}, s.t., $\forall h \in [H]$, for all $s_h \in \mathcal{S}_h$ (or only for those s_h with $d^*_{\text{Lo},h}(s_h) > 0$), we have: $V^*_{\text{Lo},h}(s_h) \geq V^*_{\text{Hi},h}(s_h) - \frac{\Delta_{\min}}{2(H+1)}$. In TRL-MST setting, we assume each $M_{\text{Lo},w}$ has OVD over M_{Hi}.[2]

We remark that Assump. B is a rather mild condition that naturally holds with reward shaping. Note that since $V^*_{\text{Hi},h}(\cdot) \leq H - h$, by shifting the reward function of M_{Lo} to $r'_{\text{Lo},h}(\cdot,\cdot) = r_{\text{Lo},h}(\cdot,\cdot) + 1$, we immediately obtain the OVD property. Even though, such a reward shift may impair the set of transferable states as we will introduce in Def. 2.2. We provide several reasonable settings in Appx. A.3 including identical model [13], small model difference, and known model difference, where Assump. B is satisfied and there exists a non-empty set of transferable states. We also point out that several existing work on transfer RL assumed something similar or even stronger [9, 10], which we defer a thorough discussion to Appx. A.2.

Assumption C. The learner has access to a quantity $\widetilde{\Delta}_{\min}$ satisfying $0 \leq \widetilde{\Delta}_{\min} \leq \Delta_{\min}$.

The final one is about the knowledge of a lower bound of $\Delta_{\min}$, which can always be satisfied by choosing $\widetilde{\Delta}_{\min} = 0$. Nevertheless, it would be more beneficial if the learner has access to some quantity $\widetilde{\Delta}_{\min}$ closer to $\Delta_{\min}$ than 0. As we introduce below, the magnitude of $\widetilde{\Delta}_{\min}$ is related to how we quantify the similarity between M_{Lo} and M_{Hi} and which states we expect to benefit from knowledge transfer. Below we focus on the single source task setting and defer the discussion for TRL-MST setting to Sec. 5.

Definition 2.1 (ε-Close)**.** Task M_{Hi} is ε-close to task M_{Lo} on s_h at step h for some $\varepsilon > 0$, if $V^*_{\text{Lo},h}(s_h) - V^*_{\text{Hi},h}(s_h) \leq \varepsilon$ and $\pi^*_{\text{Hi}}(s_h) = \pi^*_{\text{Lo}}(s_h)$.

Definition 2.2 (λ-Transferable States)**.** State s_h is λ-transferable for some $\lambda > 0$, if $d^*_{\text{Lo}}(s_h) > \lambda$ and M_{Hi} is $\frac{\widetilde{\Delta}_{\min}}{4(H+1)}$-close to M_{Lo} on s_h. The set of λ-transferable states at $h \in [H]$ is denoted as $\mathcal{Z}^{\lambda}_h$.

We regard s_h in M_{Lo} has transferable knowledge to M_{Hi}, if it can be reached by optimal policy in M_{Lo} and the optimal value and action at s_h for two tasks are similar. Here the condition $d^*_{\text{Lo}}(s_h) > 0$ is necessary since in M_{Lo}, only the states reachable by π^*_{Lo} can be explored sufficiently by Alg^{Lo} due to its near optimal regret. Combining with Assump. B, one can observe that the value difference on transferable states are controlled by $|V^*_{\text{Lo},h}(s_h) - V^*_{\text{Hi},h}(s_h)| = O(\frac{\widetilde{\Delta}_{\min}}{H}) \leq O(\frac{\Delta_{\min}}{H})$. As we will show in Thm. 3.2 in Sec. 3, the term $O(\Delta_{\min})$ is indeed unimprovable if we expect robustness.

3 Lower Bound Results: Necessary Condition for Robust Transfer

Now we establish lower bounds that show Assump. B is necessary and how the magnitude of $\Delta_{\min}$ restricts the robust transfer objective. The results in this section are based on two-armed Bernoulli bandits for simplicity, and the proofs are deferred to Appx. B. By extending these hard instances to RL case, there is a gap caused by the additional $\frac{1}{H}$ in Assump. B and Def. 2.2, which comes from the requirement of our algorithm design, and we leave it to the future work.

Justification for Assump. B We show that if Assump. B is violated, it is impossible to have algorithms $(\text{Alg}^{\text{Lo}}, \text{Alg}^{\text{Hi}})$ to simultaneously achieve constant regret when $M_{\text{Lo}} = M_{\text{Hi}}$, while retaining sub-linear regret for all the cases regardless of the similarity between M_{Lo} and M_{Hi}. Here we require constant regret on $M_{\text{Lo}} = M_{\text{Hi}}$ since we believe it is a minimal expectation to achieve benefits in transfer when

[2] For convenience, we include the bandit setting as a special case with $H = 0$; see also Def. 2.1, 2.2 and F.1.

https://doi.org/10.52202/075280-2268

the two tasks are identical. Intuitively, without Assump. B, even if we know $\mu_{\text{Lo}}(i^*_{\text{Lo}}) = \mu_{\text{Hi}}(i^*_{\text{Lo}})$, we cannot ensure i^*_{Lo} is the optimal arm in M_{Hi}. Then, if $(\text{Alg}^{\text{Lo}}, \text{Alg}^{\text{Hi}})$ can achieve constant regret on $M_{\text{Lo}} = M_{\text{Hi}}$, the algorithm must stop exploration on the arm $i \neq i^*_{\text{Lo}}$ after finite steps, and thus, it suffers from linear regret in another instance of M_{Hi} where $i^*_{\text{Lo}} \neq i^*_{\text{Hi}}$.

Moreover, Thm. 3.1 holds even if the learner has full information of M_{Lo}, where the setting degenerates to normal transfer RL since there is no need to explore M_{Lo}. This explains why similar assumptions to Assump. B are considered in previous transfer RL works [9, 10].

Theorem 3.1. *Under the violation of Assump. B, even regardless of the optimality of Alg^{Lo}, for each algorithm pair (Alg^{Lo}, Alg^{Hi}), it cannot simultaneously (1) achieve constant regret for the case when $M_{Lo} = M_{Hi}$ and (2) ensure sub-linear regret in all the other cases.*

$\Delta_{\min}$ in Tolerance Error is Inevitable Next, we show that, if M_{Hi} and M_{Lo} are Δ-close for some $\Delta \geq \frac{\Delta_{\min}}{2}$, in general we cannot expect to achieve constant regret on M_{Hi} by leveraging M_{Lo} without other loss. Similar to Thm. 3.1, the main idea is to construct different instances for M_{Hi} with different optimal arms and cannot be distinguished within finite number of trials.

Theorem 3.2. *[Transferable States are Restricted by $\Delta_{\min}$] Under Assump. B, regardless of the optimality of Alg^{Lo}, given arbitrary $\Delta_{\min}$ and arbitrary $\Delta \in [\frac{\Delta_{\min}}{2}, \Delta_{\min}]$, for each algorithm pair (Alg^{Lo}, Alg^{Hi}), it cannot simultaneously (1) achieve constant regret for the case when M_{Lo} and M_{Hi} with minimal gap $\Delta_{\min}$ are Δ-close, and (2) ensure sub-linear regret in all other cases.*

4 Robust Tiered MAB/RL with Single Source Task

In this section, we study Tiered MAB and Tiered RL when a single low-tier task M_{Lo} is available. The key challenge compared with [13] is that we do not have knowledge about whether M_{Lo} and M_{Hi} are similar or not so the pure exploitation will not work. Instead, the algorithm should be able to identify whether M_{Lo} and M_{Hi} are close enough to transfer by data collected so far, and balance between the exploration by itself and the exploitation from M_{Lo} at the same time.

To overcome the challenge, we identify a state-wise checking event, such that, under Assump. B, if s_h is transferable, the event is true almost all the time, and otherwise, every mistake will reduce the uncertainty so the chance the event holds is limited. By utilizing it, our algorithm can wisely switch between optimistic exploration and pessimistic exploitation and achieve robust transfer. In Sec. 4.1, we start with the MAB setting and illustrate the main idea, and in Sec. 4.2, we generalize our techniques to RL setting, and discuss how to overcome the challenges brought by state transition.

4.1 Robust Transfer in Tiered Multi-Armed Bandits

The algorithm is provided in Alg. 1. We choose Alg^{Lo} as UCB, and Alg^{Hi} as an exploitation-or-UCB style algorithm branched by a checking event in line 7, which is the key step to avoid negative transfer.

Algorithm 1: Robust Tiered MAB

1 **Initilize**: $\alpha > 2$; $N^1_{\text{Lo}}(i)$, $N^1_{\text{Hi}}(i)$, $\widehat{\mu}^1_{\text{Lo}}(i)$, $\widehat{\mu}^1_{\text{Hi}}(i) \leftarrow 0$, $\forall i \in \mathcal{A}$; $f(k) := 1 + 16A^2(k+1)^2$
2 Pull each arm at the beginning A iterations
3 **for** $k = A+1, A+2, ..., K$ **do**
4 $\quad \overline{\mu}^k_{\text{Lo}}(i) \leftarrow \widehat{\mu}^k_{\text{Lo}}(i) + \sqrt{\frac{2\alpha \log f(k)}{N^k_{\text{Lo}}(i)}}, \quad \overline{\pi}^k_{\text{Lo}} \leftarrow \arg\max_i \overline{\mu}^k_{\text{Lo}}(i), \quad \pi^k_{\text{Lo}} \leftarrow \overline{\pi}^k_{\text{Lo}}.$
5 $\quad \underline{\mu}^k_{\text{Lo}}(i) \leftarrow \widehat{\mu}^k_{\text{Lo}}(i) - \sqrt{\frac{2\alpha \log f(k)}{N^k_{\text{Lo}}(i)}}, \quad \underline{\pi}^k_{\text{Lo}} \leftarrow \arg\max_i \underline{\mu}^k_{\text{Lo}}(i).$
6 $\quad \overline{\mu}^k_{\text{Hi}}(i) \leftarrow \widehat{\mu}^k_{\text{Hi}}(i) + \sqrt{\frac{2\alpha \log f(k)}{N^k_{\text{Hi}}(i)}}, \quad \overline{\pi}^k_{\text{Hi}} \leftarrow \arg\max_i \overline{\mu}^k_{\text{Hi}}(i).$
7 $\quad$ **if** $\underline{\mu}^k_{\text{Lo}}(\underline{\pi}^k_{\text{Lo}}) \leq \overline{\mu}^k_{\text{Hi}}(\underline{\pi}^k_{\text{Lo}}) + \varepsilon$ and $N^k_{\text{Lo}}(\underline{\pi}^k_{\text{Lo}}) > k/2$ **then** $\pi^k_{\text{Hi}} \leftarrow \underline{\pi}^k_{\text{Lo}}$ **else** $\pi^k_{\text{Hi}} \leftarrow \overline{\pi}^k_{\text{Hi}}$.
8 $\quad$ Interact $M_{\text{Hi}}/M_{\text{Lo}}$ by $\pi^k_{\text{Hi}}/\pi^k_{\text{Lo}}$; Update $N^{k+1}_{\text{Lo}}/N^{k+1}_{\text{Hi}}$ and empirical mean $\widehat{\mu}^{k+1}_{\text{Lo}}/\widehat{\mu}^{k+1}_{\text{Hi}}$.
9 **end**

Key Insights: Separation between Transferable and Non-Transferable Cases To understand our checking event, we consider the following two cases: (1) M_{Lo} and M_{Hi} are ε-close, and (2) $i^*_{\text{Hi}} \neq i^*_{\text{Lo}}$

https://doi.org/10.52202/075280-2268

(in the rest cases we have $i^*_{\text{Hi}} = i^*_{\text{Lo}}$ but $\mu_{\text{Lo}}(i^*_{\text{Lo}}) > \mu_{\text{Hi}}(i^*_{\text{Lo}}) + \varepsilon$, so exploiting from M_{Lo} is harmless). Recall Assump. B, in Case 1, we have $\mu_{\text{Lo}}(i^*_{\text{Lo}}) \le \mu_{\text{Hi}}(i^*_{\text{Lo}}) + \varepsilon$, while in Case 2, with an appropriate choice of ε (e.g. $\varepsilon = \frac{\widetilde{\Delta}_{\min}}{4} < \frac{\Delta_{\min}}{4}$), we have $\mu_{\text{Lo}}(i^*_{\text{Lo}}) \ge \mu_{\text{Hi}}(i^*_{\text{Hi}}) - \frac{\Delta_{\min}}{2} \ge \mu_{\text{Hi}}(i^*_{\text{Lo}}) + \varepsilon + \frac{\Delta_{\text{Hi}}(i^*_{\text{Lo}})}{2}$, which reveals the separation between two cases. As a result, if we can construct an uncertainty-based upper bound $\overline{\mu}^k_{\text{Hi}}(i^*_{\text{Lo}})$ for $\mu_{\text{Hi}}(i^*_{\text{Lo}})$, we should expect the event $\mathcal{E} := \mu_{\text{Lo}}(i^*_{\text{Lo}}) < \overline{\mu}^k_{\text{Hi}}(i^*_{\text{Lo}}) + \varepsilon$ almost always be true in Case 1, while in Case 2, everytime $\mathcal{E}$ occurs and Alg^{Hi} takes i^*_{Lo}, the "self-correction" is triggered: the uncertainty is reduced so the estimation $\overline{\mu}^k_{\text{Hi}}(i^*_{\text{Lo}})$ gets closer to $\mu_{\text{Lo}}(i^*_{\text{Lo}})$, and because of the separation between $\mu_{\text{Lo}}(i^*_{\text{Lo}})$ and $\mu_{\text{Hi}}(i^*_{\text{Lo}})$, the number of times that $\mathcal{E}$ is true is limited. The remaining issue is that we do not know $\mu_{\text{Lo}}(i^*_{\text{Lo}})$ and i^*_{Lo}, and we approximate them with LCB value $\underline{\mu}^k_{\text{Lo}}(\cdot)$ and its greedy policy. The additional checking event $N^k_{\text{Lo}}(\underline{\pi}^k_{\text{Lo}}) > k/2$ is used to increase the confidence that $\underline{\pi}^k_{\text{Lo}} = i^*_{\text{Lo}}$ once transfer, which also contributes to reducing the regret. Finally, to achieve constant regret, we use α to control the total failure rate to be $\sum_{k=1}^{+\infty} k^{-\Theta(\alpha)} = C$ for some constant C. We summarize the main result in Thm. 4.1 and defer the proof to Appx. C.

Theorem 4.1. *[Tiered MAB with Single Source Tasks] Under Assump. A, B and C, by running Alg. 1 with $\varepsilon = \frac{\widetilde{\Delta}_{\min}}{4}$ and $\alpha > 2$, we always have $\textit{Regret}_K(M_{Hi}) = O(\sum_{\Delta_{Hi}(i)>0} \frac{1}{\Delta_{Hi}(i)} \log K)$. Moreover, if M_{Hi} and M_{Lo} are $\frac{\widetilde{\Delta}_{\min}}{4}$-close, we have: $\textit{Regret}_K(M_{Hi}) = O(\sum_{\Delta_{Hi}(i)>0} \frac{1}{\Delta_{Hi}(i)} \log \frac{A}{\Delta_{\min}})$.*

Comparison with [22, 13] As we can see, our algorithm can automatically achieve constant regret if tasks are similar while retaining near-optimal otherwise. Notably, even when $M_{\text{Hi}} = M_{\text{Lo}}$, our regret bound $\widetilde{O}(\sum_{\Delta_{\text{Hi}}(i)>0} \frac{1}{\Delta_{\text{Hi}}(i)})$ is strictly better than $\widetilde{O}(\sqrt{\frac{A}{\Delta_{\min}}}\sqrt{\sum_{\Delta_{\text{Hi}}(i)>0} \frac{1}{\Delta_{\text{Hi}}(i)}})$ in [22] and $\widetilde{O}(\sum_{\Delta_{\text{Hi}}(i)>0}(A-i)(\frac{1}{\Delta_{\text{Hi}}(i)} - \frac{\Delta_{\text{Hi}}(i)}{\Delta_{\text{Hi}}(i-1)^2}))$ in [13] under the same setting.

4.2 Robust Transfer in Tiered Tabular RL

In this section, we focus on RL setting. We provide the algorithm in Alg. 2, where we defer the details of **ModelLearning** function and the requirements for **Bonus** function to Appx. D.1. In the following, we first highlight our main result.

Theorem 4.2. *[Tiered RL with Single Source Tasks] Under Assump. A, B and C, Cond. D.1 for Alg^{Lo} and Cond. D.3 for **Bonus** function, by running Alg. 2 with $\varepsilon = \frac{\widetilde{\Delta}_{\min}}{4(H+1)}$, $\alpha > 2$, an arbitrary $\lambda > 0$, we have*

$$\textit{Regret}_K(M_{Hi}) = O\Big(SH \sum_{h=1}^{H} \sum_{(s_h,a_h)\in\mathcal{S}_h\times\mathcal{A}_h\setminus\mathcal{C}_h^\lambda} (\frac{H}{\Delta_{\min}} \wedge \frac{1}{\Delta_{Hi}(s_h,a_h)}) \log(SAHK)\Big).$$

Here the set $\mathcal{C}_h^\lambda \subset \mathcal{S}_h \times \mathcal{A}_h$ captures the benefitable state-action pairs to be introduced later. For simplicity, in Thm. 4.2 above, we omit all constant terms independent with K that may include λ^{-1}, $\Delta_{\min}^{-1}$ or $\log 1/d^*_h(\cdot)$. The complete version of Thm. 4.2 can be found in Thm. D.16. As we can see, comparing with pure online learning algorithms [23, 34, 7], in our setting, M_{Hi} only suffers non-constant regret on a subset of the state-action space. The SH factor may be further improved by choosing better **Bonus** functions than our Example D.4 given in Appx. D.1.

Different from the bandit setting, the state transition causes more challenges. In the following, we first explain the algorithm design to highlight how we overcome the difficulties, and then provide the analysis and proof sketch. Detailed proofs can be found in Appx. D.

Technical Challenges and Algorithm Design Similar to MAB setting, for M_{Lo} we choose an arbitrary near-optimal algorithm, and for M_{Hi} we set up a state-wise checking event $\underline{V}^k_{\text{Lo},h}(\cdot) \le \widetilde{Q}^k_{\text{Hi},h}(\cdot, \pi^k_{\text{Lo},h}) + \varepsilon$ in line 15 to determine whether to exploit from M_{Lo} or not. Here $\underline{V}^k_{\text{Lo},h}$ and $\widetilde{Q}^k_{\text{Hi},h}$ serve as lower and upper bounds for V^*_{Lo} and Q^*_{Hi}, and $\underline{V}^k_{\text{Lo},h}$ and $\underline{\pi}^k_{\text{Lo},h}$ are constructed by Pessimistic Value Iteration [16], which can be shown to converge to $V^*_{\text{Lo},h}$ and $\pi^*_{\text{Lo},h}$, respectively. Similar to [13], for the choice of Alg^{Lo} and the bonus term used to construct lower/upper confidence estimation, we consider general algorithm framework under Cond. D.1 and Cond. D.3 in Appx. D.1.

To overcome challenges resulting from state transition, we make two major modifications when moving from MAB to RL setting. First of all, because of the constraint on the optimality of Alg^{Lo},

https://doi.org/10.52202/075280-2268

Algorithm 2: Robust Tiered RL

Input: Ratio $\lambda \in (0,1)$; $\alpha > 2$; Auxiliary functions **Bonus** and **ModelLearning**; Sequence of confidence level $(\delta_k)_{k\geq 1}$ with $\delta_k = 1/SAHk^\alpha$; $\varepsilon := \widetilde{\Delta}_{\min}/4(H+1)$ for some $\widetilde{\Delta}_{\min} \leq \Delta_{\min}$

Initialize: $D^0_{\text{Lo}}/D^0_{\text{Hi}} \leftarrow \{\}$; $\forall k$, $\underline{V}^k_{(\cdot),H+1}, \underline{Q}^k_{(\cdot),H+1}, \widetilde{V}^k_{\text{Hi},H+1}, \widetilde{Q}^k_{\text{Hi},H+1} \leftarrow 0$.

for $k = 1, 2, \ldots$ **do**

  $\pi_{\text{Lo}} \leftarrow \text{Alg}^{\text{Lo}}(D^{k-1}_{\text{Lo}})$;

  $\{\widehat{\mathbb{P}}^k_{\text{Lo},h}\}_{h=1}^H \leftarrow$ **ModelLearning**(D^{k-1}_{Lo}), $\{b^k_{\text{Lo},h}\}_{h=1}^H \leftarrow$ **Bonus**$(D^{k-1}_{\text{Lo}}, \delta_k)$.

  for $h = H, H-1 \ldots, 1$ **do**

    $\underline{Q}^k_{\text{Lo},h}(\cdot,\cdot) \leftarrow \max\{0, r_{\text{Lo},h}(\cdot,\cdot) + \widehat{\mathbb{P}}^k_{\text{Lo},h}\underline{V}^k_{\text{Lo},h+1}(\cdot,\cdot) - b^k_{\text{Lo},h}(\cdot,\cdot)\}$.

    $\underline{V}^k_{\text{Lo},h}(\cdot) = \max_a \underline{Q}^k_{\text{Lo},h}(\cdot,a)$, $\quad \underline{\pi}^k_{\text{Lo},h}(\cdot) \leftarrow \arg\max_a \underline{Q}^k_{\text{Lo},h}(\cdot,a)$.

  end

  $\{\widehat{\mathbb{P}}^k_{\text{Hi},h}\}_{h=1}^H \leftarrow$ **ModelLearning**(D^{k-1}_{Hi}); $\quad \{b^k_{\text{Hi},h}\}_{h=1}^H \leftarrow$ **Bonus**$(D^{k-1}_{\text{Hi}}, \delta_k)$.

  for $h = H, H-1 \ldots, 1$ **do**

    $\underline{Q}^{\pi^k_{\text{Hi}}}_{\text{Hi},h}(\cdot,\cdot) \leftarrow \max\{0, r_{\text{Hi},h}(\cdot,\cdot) + \widehat{\mathbb{P}}^k_{\text{Hi},h}\underline{V}^k_{\text{Hi},h+1}(\cdot,\cdot) - b^k_{\text{Hi},h}(\cdot,\cdot)\}$

    $\widetilde{Q}^k_{\text{Hi},h}(\cdot,\cdot) \leftarrow \min\{H, r_{\text{Hi},h}(\cdot,\cdot) + \widehat{\mathbb{P}}^k_{\text{Hi},h}\widetilde{V}^k_{\text{Hi},h+1}(\cdot,\cdot) + b^k_{\text{Hi},h}(\cdot,\cdot)\}$.

    for $s_h \in \mathcal{S}_h$ **do**

      if $\underline{V}^k_{Lo,h}(s_h) \leq \widetilde{Q}^k_{Hi,h}(s_h, \underline{\pi}^k_{Lo,h}) + \varepsilon$ *and* $\max_a N^k_{Lo,h}(s_h,a) > \frac{\lambda}{3}k$ **then**

        $\pi^k_{\text{Hi}}(s_h) \leftarrow \arg\max_a N^k_{\text{Lo},h}(s_h,a)$. // "Trust and Exploit" Branch

      end

      else $\pi^k_{\text{Hi}}(\cdot) \leftarrow \arg\max_a \widetilde{Q}^k_{\text{Hi},h}(\cdot,a)$. // "Explore by itself" Branch ;

      $\widetilde{V}^k_{\text{Hi},h}(s_h) \leftarrow \min\{H, \widetilde{Q}^k_{\text{Hi},h}(s_h,\pi^k_{\text{Hi}}) + \frac{1}{H}(\widetilde{Q}^k_{\text{Hi},h}(s_h,\pi^k_{\text{Hi}}) - \underline{Q}^{\pi^k_{\text{Hi}}}_{\text{Hi},h}(s_h,\pi^k_{\text{Hi}}))\}$

      $\underline{V}^{\pi^k_{\text{Hi}}}_{\text{Hi},h}(s_h) = \underline{Q}^{\pi^k_{\text{Hi}}}_{\text{Hi},h}(s_h,\pi^k_{\text{Hi}})$.

    end

  end

  Deploy $\pi_{\text{Lo}}/\pi_{\text{Hi}}$ to $M_{\text{Lo}}/M_{\text{Hi}}$ and get $\tau^k_{\text{Lo}}/\tau^k_{\text{Hi}}$; and update $D^k_{\text{Lo}}, D^k_{\text{Hi}}$.

end

we cannot expect M_{Lo} to provide useful information on those (s_h, a_h) with $d^*_{\text{Lo}}(s_h,a_h) = 0$ since they will not be explored sufficiently. Therefore, in the checking event in line 15, we include $\max_a N^k_{\text{Lo},h}(s_h,a) > \Theta(\lambda k)$ as a criterion, where λ is a hyper-parameter chosen and input to the algorithm. Intuitively, for all s_h, a_h, we should expect $N^k_{\text{Lo},h}(s_h,a_h) \approx \widetilde{O}(d^*_{\text{Lo}}(s_h,a_h)\cdot k)$ when k is large enough. Therefore, by comparing $N^k_{\text{Lo},h}$ with λk, we can filter out those s_h with $d^*_{\text{Lo}}(s_h) < \lambda$ to avoid harm from inaccurate estimation.

Secondly and more importantly, different from MAB setting, besides the error occurred at a particular step, we also need to handle the error accumulated during the back-propagation process of value iteration. In our case, this is reflected by the loss of overestimation when we incorporate selective exploitation into the optimism-based exploration framework. To see this, suppose at some s_h, we have an overestimation on optimal value $Q^*_{\text{Hi},h}$ denoted as $\widetilde{Q}^k_{\text{Hi},h}$. When the checking criterion is satisfied, if we mimic the MAB setting, i.e., assign $\pi^k_{\text{Lo},h}$ to $\pi^k_{\text{Hi},h}$ and update value by $\widetilde{V}^k_{\text{Hi},h}(s_h) \leftarrow \widetilde{Q}^k_{\text{Hi},h}(s_h,\pi^k_{\text{Hi},h})$, when $\pi^*_{\text{Lo}}(s_h) \neq \pi^*_{\text{Hi},h}(s_h)$, $\widetilde{V}^k_{\text{Hi},h}(s_h)$ is no longer guaranteed to be an overestimation for $V^*_{\text{Hi},h}(s_h)$. As $\widetilde{V}^k_{\text{Hi},h}(s_h)$ involves in back-propagation, it will pull down the estimation value for its ancestor states, thus reducing the chance to visit s_h and slowing down the "self-correction process" which works well in MAB setting.

The key insight to overcome such difficulty is that, if the checking event holds yet $\pi^*_{\text{Lo}}(s_h) \neq \pi^*_{\text{Hi},h}(s_h)$, the gap between $\widetilde{Q}^k_{\text{Hi},h}(s_h,\pi^*_{\text{Lo}})$ and $Q^*_{\text{Hi},h}(s_h,\pi^*_{\text{Lo}})$ should not be small, and we can show that $\widetilde{Q}^k_{\text{Hi},h}(s_h,\underline{\pi}^k_{\text{Lo}}) \approx \widetilde{Q}^k_{\text{Hi},h}(s_h,\pi^*_{\text{Lo}}) \geq Q^*_{\text{Hi},h}(s_h,\pi^*_{\text{Lo}}) + \Theta(\frac{H}{H+1}\Delta_{\text{Hi}}(s_h,\pi^*_{\text{Lo}}))$ with the choice of

https://doi.org/10.52202/075280-2268

$\varepsilon = O(\widetilde{\Delta}_{\min}/H)$. Therefore, revising $\widetilde{Q}^k_{\text{Hi},h}(s_h, \underline{\pi}^k_{\text{Lo}})$ by adding $1/H$ of the gap $\widetilde{Q}^k_{\text{Hi},h}(s_h, \underline{\pi}^k_{\text{Lo}}) - Q^*_{\text{Hi},h}(s_h, \pi^*_{\text{Lo}})$ (line 19) is enough to guarantee the overestimation. Lastly, since $Q^*_{\text{Hi},h}(s_h, \pi^*_{\text{Lo}})$ in unknown, we construct an underestimation $\underline{Q}^{\pi^k_{\text{Hi}}}_{\text{Hi},h}(s_h, \pi^*_{\text{Lo}})$ and use it instead. As a result, we have the following theorem, where the clip function is defined by $\text{Clip}[x|w] := x \cdot \mathbb{I}[x \geq w]$.

Theorem 4.3. *There exists $k_{ost} = Poly(S, A, H, \lambda^{-1}, \Delta^{-1}_{\min})$, such that, for all $k \geq k_{ost}$, on some event $\mathcal{E}_k$ with $\mathbb{P}(\mathcal{E}_k) \leq 3\delta_k$, we have $Q^*_{Hi,h}(s_h, a_h) \leq \widetilde{Q}^k_{Hi,h}(s_h, a_h),\ V^*_{Hi,h}(s_h) \leq \widetilde{V}^k_{Hi,h}(s_h), \forall h \in [H],\ s_h \in \mathcal{S}_h,\ a_h \in \mathcal{A}_h$ and*

$$V^*_{Hi,1}(s_1) - V^{\pi^k_{Hi}}_{Hi,1}(s_1) \leq 2e\mathbb{E}_{\pi^k_{Hi}}\left[\sum_{h=1}^{H} Clip\left[\min\{H, 4b^k_{Hi,h}(s_h, a_h)\}|\frac{\Delta_{\min}}{4eH} \vee \frac{\Delta_{Hi}(s_h, a_h)}{4e}\right]\right]. \quad (1)$$

Benefits of Knowledge Transfer We first take a look at $k \geq k_{ost}$. As implied from Eq. (1), we can upper bound the regret on each s_h, a_h by summing over the RHS of Eq. (1). Note that by Cond. D.3, $b^k_{\text{Hi},h}(s_h, a_h) = O(\frac{\text{Poly}(SAH)\log k}{\sqrt{N^k_{\text{Hi},h}(s_h, a_h)}})$ and $\mathbb{E}[N^k_{\text{Hi},h}(s_h, a_h)] = \sum_{k'=1}^{k-1} d^{\pi^k_{\text{Hi}}}(s_h, a_h)$, we can establish the near-optimal regret bound with similar techniques in [23] regardless of the similarity between M_{Hi} and M_{Lo}. Moreover, because of the knowledge transfer from M_{Lo}, we can achieve better regret bounds for M_{Hi}. In the following, we characterize three subclasses of state-action pairs, on which Alg^{Hi} only suffers constant regret. *First of all*, for those $s_h \in \mathcal{Z}^\lambda_h$, we can expect the checking event almost always hold for arbitrary k. Hence, when k is large enough, $\pi^k_{\text{Hi},h}(s_h) = \underline{\pi}^k_{\text{Lo},h}(s_h) \approx \pi^*_{\text{Hi},h}(s_h)$, implying Alg^{Hi} will almost never take sub-optimal actions at s_h since then. We denote this first subclass as $\mathcal{C}^{1,\lambda}_h := \{(s_h, a_h)|s_h \in \mathcal{Z}^\lambda_h, a_h \neq \pi^*_{\text{Hi},h}(s_h)\}$. *Secondly*, note that, given a state s_h, if all possible trajectories starting from s_1 to s_h have overlap with $\mathcal{C}^{1,\lambda}_{h'}$ for some $h' \in [h-1]$, when k is large enough, π^k_{Hi} will almost have no chance to reach s_h and will not suffer the regret at s_h. For convenience, we define function $\text{Block}(\{\mathcal{C}^{1,\lambda}_{h'}\}^{h-1}_{h'=1}, s_h)$ which takes `True` for those states described above, and takes `False` for the others. Then, we define the second subclass by $\mathcal{C}^{2,\lambda}_h := \{(s_h, a_h)|\text{Block}(\{\mathcal{C}^{1,\lambda}_{h'}\}^{h-1}_{h'=1}, s_h) = \texttt{True},\ s_h \notin \mathcal{Z}^\lambda_h,\ a_h \in \mathcal{A}_h\}$. *Finally*, for those s_h, a_h with $d^*_{\text{Hi}}(s_h, a_h) > 0$, we can show $N^k_{\text{Hi},h}(s_h, a_h) \approx d^*_{\text{Hi}}(s_h, a_h)k$. Therefore, $b^k_{\text{Hi},h}(s_h, a_h) \propto \log k/\sqrt{N^k_{\text{Hi},h}(s_h, a_h)}$ in Eq. (1) will decay and the clipping operator will take effect, which leads to constant regret. This third subclass is denoted by $\mathcal{C}^*_h := \{(s_h, a_h)|d^*_{\text{Hi}}(s_h, a_h) > 0\}$. Based on the above discussion, we define $\mathcal{C}^\lambda_h := \mathcal{C}^{\lambda,1}_h \cup \mathcal{C}^{\lambda,2}_h \cup \mathcal{C}^*_h$ to be the benefitable states set in Thm. 4.2.

For $k \leq k_{ost}$, for the lack of overestimation, we simply use H to upper bound the value gap $V^* - V^{\pi^k_{\text{Hi}}}$. This results in a $\text{Poly}(S, A, H, \lambda^{-1}, \Delta^{-1}_{\min})$ burn-in term, which was omitted in Thm. 4.2 since it is independent with K. Besides, by the definition of k_{ost} in Thm. 4.3, we can see the trade-off of choosing λ: a smaller λ can enlarge $\mathcal{C}^\lambda_h$ so we have constant regret on more state-action pairs, while it also results in the delay of overestimation by the larger k_{ost}.

Constant Regret in the Entire MDP We may expect constant regret in the entire M_{Hi} in some special cases. Note that, if $\forall h \in [H], \forall s_h$ with $d^*_{\text{Hi}}(s_h) > 0$, $s_h \in \mathcal{Z}^\lambda_h$, we have $\mathcal{C}^\lambda_h = \mathcal{S}_h \times \mathcal{A}_h$, $\text{Regret}_K(M_{\text{Hi}})$ will be independent w.r.t. K. From this perspective, if λ is chosen appropriately, e.g. $\lambda \leq \min_{s_h} d^*_{\text{Hi}}(s_h)$, we can recover the constant regret under the setting $M_{\text{Lo}} = M_{\text{Hi}}$ in [13].

Choice of λ In this paper, we do not treat λ as a parameter to optimize. In practice, without prior knowledge about $\max_{s_h} d^*_{\text{Lo}}(s_h)$, one may choose $\lambda = O(1/S)$ to ensure some chance that transferable states exist, since there exists at least some states satisfying $d^*_{\text{Lo}}(s_h) \geq 1/S$.

5 Robust Tiered MAB/RL with Multiple Low-Tier Tasks

Now, we focus on the case when a source task set $\mathcal{M}_{\text{Lo}} := \{M_{\text{Lo},w}\}^W_{w=1}$ is available (see Frw. 5 in Appx. A). Our objective is to achieve benefits on those states s_h as long as there exists some task $w \in [W]$ such that $M_{\text{Lo},w}$ and M_{Hi} are close on s_h, while retaining near-optimal regret in other cases under Assump. B. The key challenge comparing with single task case is that, Alg^{Hi} has to identify for each state which task in $\mathcal{M}_{\text{Lo}}$ is the appropriate one to leverage. The main novelty and contribution

https://doi.org/10.52202/075280-2268

in this section is a task selection mechanism we call *"Trust till Failure"*, which can automatically adapt to the similar task if it exists. We first highlight the main results for MAB and RL setting.

Theorem 5.1. *[Tiered MAB with Multiple Source Tasks] Under Assump. A, B, and C, by running Alg. 3 with $\mathcal{M}_{Lo} = \{M_{Lo,w}\}_{w=1}^{W}$ and M_{Lo}, with $\varepsilon = \frac{\widetilde{\Delta}_{\min}}{4}$ and $\alpha > 2$, we always have: $Regret_K(M_{Hi}) = O(\sum_{\Delta_{Hi}(i)>0} \frac{1}{\Delta_{Hi}(i)} \log(WK))$. Moreover, if at least one task in $\mathcal{M}_{Lo}$ is $\frac{\widetilde{\Delta}_{\min}}{4}$-close to M_{Hi}, we further have: $Regret_K(M_{Hi}) = O(\sum_{\Delta_{Hi}(i)>0} \frac{1}{\Delta_{Hi}(i)} \log \frac{AW}{\Delta_{\min}})$.*

Theorem 5.2. *[Tiered RL with Multiple Source Tasks] Under Assump. A, B, C, and Cond. D.3, F.4, by running Alg. 7 in Appx. F.1 with $\varepsilon = \frac{\widetilde{\Delta}_{\min}}{4(H+1)}$, $\alpha > 2$ and any $\lambda > 0$, we have*

$$Regret_K(M_{Hi}) = O\Big(SH \sum_{h=1}^{H} \sum_{(s_h,a_h)\in\mathcal{S}_h\times\mathcal{A}_h\setminus\mathcal{C}_h^{\lambda,[W]}} (\frac{H}{\Delta_{\min}} \wedge \frac{1}{\Delta_{Hi}(s_h,a_h)}) \log(SAHWK)\Big).$$

Algorithm 3: Robust Tiered MAB with Multiple Source Tasks

1 **Initilize**: $\alpha > 2$; $N_{\text{Lo}}^1(i)$, $N_{\text{Hi}}^1(i)$, $\widehat{\mu}_{\text{Lo}}^1(i)$, $\widehat{\mu}_{\text{Hi}}^1(i) \leftarrow 0$, $\forall i \in \mathcal{A}$; $f(k) := 1 + 16TA^2(k+1)^2$
2 Pull each arm at the beginning A iterations. Set $w^A \leftarrow$ `Null`.
3 **for** $k = A+1, 2, ..., K$ **do**
4 **for** $w = 1, 2..., W$ **do**
5 $\overline{\mu}_{\text{Lo},w}^k(i) \leftarrow \widehat{\mu}_{\text{Lo},w}^k(i) + \sqrt{\frac{2\alpha \log f(k)}{N_{\text{Lo},w}(i)}}$, $\overline{\pi}_{\text{Lo},w}^k \leftarrow \arg\max_i \overline{\mu}_{\text{Lo},w}^k(i)$, $\pi_{\text{Lo},w}^k \leftarrow \overline{\pi}_{\text{Lo},w}^k$.
6 $\underline{\mu}_{\text{Lo},w}^k(i) \leftarrow \widehat{\mu}_{\text{Lo},w}^k(i) - \sqrt{\frac{2\alpha \log f(k)}{N_{\text{Lo},w}(i)}}$, $\underline{\pi}_{\text{Lo},w}^k \leftarrow \arg\max_i \underline{\mu}_{\text{Lo},w}^k(i)$.
7 **end**
8 $\overline{\mu}_{\text{Hi}}^k(i) \leftarrow \widehat{\mu}_{\text{Hi}}^k(i) + \sqrt{\frac{2\alpha \log f(k)}{N_{\text{Hi}}^k(i)}}$, $\overline{\pi}_{\text{Hi}}^k \leftarrow \arg\max_i \overline{\mu}_{\text{Hi}}^k(i)$.
9 $\mathcal{I}^k \leftarrow \{w \in [W] | \underline{\mu}_{\text{Lo},w}^k(\underline{\pi}_{\text{Lo},w}^k) \le \overline{\mu}_{\text{Hi}}^k(\underline{\pi}_{\text{Lo},w}^k) + \varepsilon \text{ and } N_{\text{Lo},w}^k(\underline{\pi}_{\text{Lo},w}^k) > k/2\}$
10 **if** $\mathcal{I}^k = \emptyset$ **then** $w^k \leftarrow$ `Null`, $\pi_{\text{Hi}}^k \leftarrow \overline{\pi}_{\text{Hi}}^k$;
11 **else**
12 **if** $w^{k-1} \neq$ *Null and* $w^{k-1} \in \mathcal{I}^k$ **then** $w^k \leftarrow w^{k-1}$, $\pi_{\text{Hi}}^k \leftarrow \underline{\pi}_{\text{Lo},w^k}^k$;
13 **else if** $w^{k-1} \neq$ *Null and* $\exists w \in \mathcal{I}^k$ *such that* $\pi_{Hi}^{k-1} = \arg\max_i N_{Lo,w}^k(i)$ **then**
14 $w^k \leftarrow w$, $\pi_{\text{Hi}}^k \leftarrow \underline{\pi}_{\text{Lo},w}^k$
15 **end**
16 **else** $w^k \sim \text{Unif}(\mathcal{I}^k)$, $\pi_{\text{Hi}}^k \leftarrow \underline{\pi}_{\text{Lo},w^k}^k$;
17 **end**
18 Interact with $M_{\text{Hi}}/\{M_{\text{Lo},}\}_{w=1}^W$ by $\pi_{\text{Hi}}^k/\{\pi_{\text{Lo},w}^k\}_{w=1}^W$;
19 Update $\{N_{\text{Lo},w}^{k+1}\}_{w=1}^W, N_{\text{Hi}}^{k+1}$ and empirical mean $\{\widehat{\mu}_{\text{Lo},w}^{k+1}\}_{w=1}^W, \widehat{\mu}_{\text{Hi}}^{k+1}$ for each arm.
20 **end**

For the lack of space, in the following, we only analyze the bandit setting to explain the key idea of our task selection strategy. For the RL setting, we defer to Appx. F the algorithm Alg. 7, detailed version of Thm. 5.2 (Thm. D.16), defintion of transferable set $\mathcal{C}_h^{\lambda,[W]}$ (Def. F.2), and technical details.

Algorithm Design and Proof Sketch for Bandit Setting The algorithm is provided in Alg. 3. Comparing with Alg. 1 in single task setting, the main difference is the task selection strategy from line 9 to line 17. We first examine each source task with a checking event similar to single task setting, and collect those feasible tasks passing the test to $\mathcal{I}^k$. Intuitively, for those M_{Lo,w^*} close to M_{Lo}, we expect $M_{\text{Lo},w} \in \mathcal{I}^k$ holds almostly for arbitrary $k > 0$, while for the other $M_{\text{Lo},w'}$, if it takes the position of w^k, following $M_{\text{Lo},w'}$ will reduce the uncertainty and it will be ruled out from $\mathcal{I}^k$, eventually. So we expect w^k can "escape" from dissimilar source tasks but be absorbed to the similar task if exists. Therefore, if $\mathcal{I}^k$ is empty, Alg^{Hi} will do exploration by itself. Otherwise, we choose one from $\mathcal{I}^k$ to transfer the action until it fails on the checking event. However, for any ε the algorithm chosen, those "marginally similar" source tasks (denoted as $M_{\text{Lo},\widetilde{w}}$), which are ε'-close to M_{Hi} for some ε' only slightly larger than ε, may cause some trouble. Because the checking event

52081 https://doi.org/10.52202/075280-2268

will finally eliminate $M_{\text{Lo},\widetilde{w}}$ since they are not ε-close, but it may occupy the position w^k for a long time before elimination, especially when ε' is extremely close to ε. After eliminating $M_{\text{Lo},\widetilde{w}}$, Alg^{Hi} needs to re-select one from $\mathcal{I}^k$. Now since other sub-optimal arms in M_{Hi} haven't been chosen for a long time and the confidence level $\delta_k = O(1/k^\alpha)$ is decreasing, $\mathcal{I}^k$ will include those dissimilar tasks again, which causes difficulty to identify the true similar task. To solve this issue, once the previous trusted task fails, we give priority to the task recommending the same action as the previous one (line 14). As a result, since $M_{\text{Lo},\widetilde{w}}$ and M_{Lo,w^*} share the optimal action, after the elimination of $M_{\text{Lo},\widetilde{w}}$, we can expect w^k to only switch among those tasks $M_{\text{Lo},w}$ with $\pi^*_{\text{Lo},w} = \pi^*_{\text{Hi}}$. We highlight this technical novelty to Lem. 5.3 below, and defer all the proofs to Appx. E.

Lemma 5.3. *[Absorbing to Similar Task] Under Assump. A, B and C, there exists a constant c^*, s.t., if there exists at least one $w^* \in [W]$ such that M_{Lo,w^*} is $\frac{\widetilde{\Delta}_{\min}}{4}$-close to M_{Hi}, by running Alg. 3 with $\varepsilon = \frac{\widetilde{\Delta}_{\min}}{4}$ and $\alpha > 2$, for any $k \geq k^* := c^* \frac{\alpha A}{\Delta^2_{\min}} \log \frac{\alpha AW}{\Delta_{\min}}$, we have $\Pr(\pi^k_{Hi} \neq i^*_{Hi}) = O(\frac{A}{k^{2\alpha-2}})$.*

6 Experiments

In this section, we evaluate our most representative algorithm, Alg. 7, in multiple source tasks setting.

Experiments Setting[3] We set $S = A = 3$ and $H = 5$. The details for construction of source and target tasks are defered to Appx. G. We adapt StrongEuler in [23] as online learning algorithm to solve source tasks, and use the bonus function in [23] as the bonus function in our Alg. 7. We evaluate our algorithm when $W = 0, 1, 2, 5$, where $W = 0$ means the high-tier task M_{Hi} is simply solved by normal online learning method (StrongEuler) without any parallel knowledge transfer. We choose $\lambda = 0.3 \approx 1/S$ in Alg. 7, and in the MDP instance we test, across all $S \cdot H = 15$ states, for $W = 1, 2, 5$, the number of transferable states would be 6, 9 and 13, respectively.

We choose iteration number $K = 1e7$, where we start the transfer since $k = 5e5$ to avoid large "burn-in" terms. As we can see, after the transfer starts, the regret in target task will suddenly increase for a while, because the target task has to make some mistakes and learn from it as a result of the model uncertainty. However, because of our algorithm design, the negative transfer will terminate after a very short period. As predicted by our theory, by adding more and more source tasks which can introduce new transferable states, the target task will suffer less and less regret.

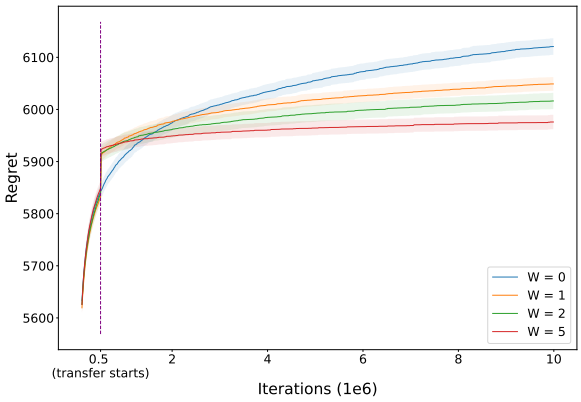

Figure 1: **Regret in the Target Task given Multiple Source Tasks** We report the result when W source tasks are available with $W = 0, 1, 2, 5$. The shadows indicate 96% confidence interval.

7 Conclusion and Future Work

In this paper, we study how to do robust parallel transfer RL when single or multiple source tasks are avilable, without knowledge on models' similarity. The possible future directions include relaxing assumptions, better strategies to leveraging multiple source tasks, and identifying mild structural assumptions allowing for more aggressive transfer, and we defer to Appx. A.4 for more details.

[3] Code is available at `https://github.com/jiaweihhuang/Robust-Tiered-RL`

https://doi.org/10.52202/075280-2268

Acknowledgments and Disclosure of Funding

The authors would like to thank Andreas Krause for valuable discussion. The work is supported by ETH research grant and Swiss National Science Foundation (SNSF) Project Funding No. 200021-207343.

References

[1] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pages 10–4, 2019.

[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] Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning, 2013.

[4] Jinglin Chen and Nan Jiang. Offline reinforcement learning under value and density-ratio realizability: the power of gaps. *arXiv preprint arXiv:2203.13935*, 2022.

[5] HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, et al. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, 118(1):e1907856118, 2021.

[6] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning, 2017.

[7] Christoph Dann, Teodor Vanislavov Marinov, Mehryar Mohri, and Julian Zimmert. Beyond value-function gaps: Improved instance-dependent regret bounds for episodic reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=CU8qQMhB3dh`.

[8] Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, Jan Peters, et al. Sharing knowledge in multi-task deep reinforcement learning. In *8th International Conference on Learning Representations,{ICLR} 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, pages 1–11. OpenReview. net, 2020.

[9] Noah Golowich and Ankur Moitra. Can q-learning be improved with advice? In *Conference on Learning Theory*, pages 4548–4619. PMLR, 2022.

[10] Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham M. Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=D-X3kH-BkpN`.

[11] Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pages 4349–4358. PMLR, 2021.

[12] Jiawei Huang, Jinglin Chen, Li Zhao, Tao Qin, Nan Jiang, and Tie-Yan Liu. Towards deployment-efficient reinforcement learning: Lower bound and optimality. *arXiv preprint arXiv:2202.06450*, 2022.

[13] Jiawei Huang, Li Zhao, Tao Qin, Wei Chen, Nan Jiang, and Tie-Yan Liu. Tiered reinforcement learning: Pessimism in the face of uncertainty and constant regret. *arXiv preprint arXiv:2205.12418*, 2022.

[14] Nan Jiang and Jiawei Huang. Minimax value interval for off-policy evaluation and policy optimization. *Advances in Neural Information Processing Systems*, 33:2747–2758, 2020.

https://doi.org/10.52202/075280-2268

[15] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *arXiv preprint arXiv:1807.03765*, 2018.

[16] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

[17] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

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

[19] Yongyuan Liang and Bangwei Li. Parallel knowledge transfer in multi-agent reinforcement learning, 2020.

[20] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems*, 33:1264–1274, 2020.

[21] Timothy A Mann and Yoonsuck Choe. Directed exploration in reinforcement learning with transferred knowledge. In *European Workshop on Reinforcement Learning*, pages 59–76. PMLR, 2013.

[22] Chloé Rouyer and Yevgeny Seldin. Tsallis-inf for decoupled exploration and exploitation in multi-armed bandits. In *Conference on Learning Theory*, pages 3227–3249. PMLR, 2020.

[23] Max Simchowitz and Kevin G Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pages 9767–9779. PMLR, 2021.

[25] Adam Taylor, Ivana Dusparic, Maxime Guériau, and Siobhán Clarke. Parallel transfer learning in multi-agent systems: What, when and how to transfer? In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

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

[27] Volodymyr Tkachuk, Sriram Ganapathi Subramanian, and Matthew E. Taylor. The effect of q-function reuse on the total regret of tabular, model-free, reinforcement learning, 2021.

[28] Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[29] Nelson Vithayathil Varghese and Qusay H Mahmoud. A survey of multi-task deep reinforcement learning. *Electronics*, 9(9):1363, 2020.

[30] Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. *arXiv preprint arXiv:2211.04974*, 2022.

[31] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022, 2007.

[32] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

[33] Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021.

https://doi.org/10.52202/075280-2268

[34] Haike Xu, Tengyu Ma, and Simon Du. Fine-grained gap-dependent bounds for tabular mdps via adaptive multi-step bootstrap. In *Conference on Learning Theory*, pages 4438–4472. PMLR, 2021.

[35] Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pages 2730–2775. PMLR, 2022.

[36] Chicheng Zhang and Zhi Wang. Provably efficient multi-task reinforcement learning with model transfer. *Advances in Neural Information Processing Systems*, 34:19771–19783, 2021.

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

https://doi.org/10.52202/075280-2268