Reinforcement Learning Can Be More Efficient with Multiple Rewards

Christoph Dann [1] **Yishay Mansour** [1 2] **Mehryar Mohri** [1 3]

Abstract

Reward design is one of the most critical and challenging aspects when formulating a task as a reinforcement learning (RL) problem. In practice, it often takes several attempts of reward specification and learning with it in order to find one that leads to sample-efficient learning of the desired behavior. Instead, in this work, we study whether directly incorporating multiple alternate reward formulations of the same task in a single agent can lead to faster learning. We analyze multi-reward extensions of action-elimination algorithms and prove more favorable instance-dependent regret bounds compared to their single-reward counterparts, both in multi-armed bandits and in tabular Markov decision processes. Our bounds scale for each state-action pair with the inverse of the largest gap among all reward functions. This suggests that learning with multiple rewards can indeed be more sample-efficient, as long as the rewards agree on an optimal policy. We further prove that when rewards do not agree, multi-reward action elimination in multi-armed bandits still learns a policy that is good across all reward functions.

1. Introduction

Crafting an appropriate reward function often poses one of the most challenging obstacles when implementing reinforcement learning in real-world scenarios. The process typically entails numerous iterations of reward engineering to come up with a definition that both captures the task accurately and enables fast learning. This frequently involves exploring different formulations by training an agent with alternative definitions and testing which yields the fastest learning (Sutton & Barto, 2018, Chapter 17.6).

[1]Google Research [2]Tel Aviv University [3]Courant Institute of Mathematical Sciences. Correspondence to: Christoph Dann <cdann@cdann.net>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Here, we aim to lessen the requirement from the reward designers to supply a single adequate reward definition and instead allow for multiple rewards, which together can help better represent the task and induce fast learning.

To illustrate our motivation, consider the street navigation task, where the objective is to drive from one city center to another in the shortest possible time. A natural way to define the reward in this scenario is to assign a negative value to each road section proportionate to the travel time required to traverse it. While this reward definition specifies the desired behavior, it may pose challenges in terms of learning, as there could be numerous alternative routes with similar travel times. The agent would need to repeatedly test these routes to determine which one is consistently the fastest. Local experts, such as experienced taxi drivers, often possess an in-depth understanding of their neighborhoods. They confidently know which road sections are not the most efficient options. Incorporating their recommendations can be valuable by introducing an additional reward function. This function assigns a significant negative reward to the discouraged road sections and zero everywhere else. Although this reward function does not provide information about other road segments, it can complement the original reward function and expedite the agent's learning process in avoiding those specific segments with higher penalties. Consequently, the agent can optimize its navigation strategy more efficiently.

Another example where multiple reward functions naturally occur is in the context of goal reaching in robotics. In this scenario, the objective is to train an agent to efficiently move the end-effector of a robot to a desired position. There are many natural ways to formulate the reward function for this task. We could give a reward of -1 unless the end-effector is within a certain bounding box around the goal, or alternatively, a negative reward proportional to the Euclidean distance of end-effector to goal. Other choices are negative rewards proportional to the squared Euclidean distance or the Manhattan distance. All of these reward function formulations are reasonable and can lead to optimal or approximately-optimal policies. However, it is not immediately evident which reward function is the easiest to learn, and certain policies may be more or less suboptimal under different reward functions. In fact, existing research (Luo et al., 2020, Table II) has observed significant differ-

ences in training times when using different natural reward formulations for this task, thereby demonstrating the impact of the reward function choice on the learning process.

These scenarios raise the question of how to best learn in the presence of multiple rewards. One simple option is to take the average of the different rewards and use it in a single-reward RL algorithm. We find however, that stronger results can be achieved when we directly work with multiple reward in the algorithm. We study natural extensions of action-elimination algorithms (Even-Dar et al., 2006; Xu et al., 2021) that directly leverage multiple rewards and learn a policy that is optimal for all provided rewards. We investigate them in the multi-armed bandit (MAB) and tabular Markov decision process (MDP) setting and prove regret bounds that are more favorable than those achieved by using just a single reward. More specifically, our main contributions are:

- We analyze action elimination in MABs with multiple rewards $\bar{r}^1, \dots \bar{r}^m$ and induced gaps $\Delta^1(a), \dots, \Delta^m(a)$. Our main regret bound scales for each action as $\frac{1}{\max\{\Delta^1(a),\dots,\Delta^m(a)\}}$ compared to $\frac{1}{\Delta^i(a)}$ in the single reward case. This scaling with the maximum gap (max-gap) shows that there is benefit to learning with multiple rewards as long as there is an action that is optimal under all rewards.
- We prove that, when no action is optimal under all rewards, action elimination still converges to an action with the smallest max-gap up to a constant factor. This shows that the algorithm is robust to inconsistent rewards while still enjoying faster convergence dictated by the max-gap. To relax the consistency assumption further, we allow our algorithm to work with other operators besides the maximum, for a better trade-off between asymptotic quality and convergence speed.
- We extend our result to tabular MDPs and show that action elimination with multiple rewards also achieves regret that scales inversely with the max-gap for each state-action pair. Again, this improves the best known results for single-reward learning in this setting.
- We provide an MDP instance where regret bounds based on max-gap for multi-reward RL are, by a factor of number of rewards, smaller than those based on individual reward gaps for single-reward RL. Importantly, this holds for any individual reward function, suggesting that multi-reward RL may be strictly easier than single-reward RL, depending on the MDP dynamics.

2. Related Work

It has been recognized early on that rewards not only specify the desired behavior but also determine how the behavior is learned by RL approaches. This has led to considerations like reward shaping (Ng et al., 1999) which stipulates conditions under which reward functions have identical optimal policies but may be easier to learn for certain algorithms. There also have been efforts on automatic reward design (Mataric, 1994; Sorg et al., 2010; Zheng et al., 2020) which alter the reward during learning, e.g. by meta gradient ascent. The idea of changing reward during learning is also prevalent in intrinsic rewards (e.g. Chentanez et al., 2004; Bellemare et al., 2016; Pathak et al., 2017) where a bonus is added to the reward to encourage exploration of lesser known parts of the MDP. Our work is orthogonal to these efforts and multiple reward RL could be combined with intrinsic rewards in optimistic algorithms.

To provide meaningful guarantees and demonstrate the benefits of RL with multiple rewards, we prove gap-dependent regret bounds. In the single-reward multi-armed bandit setting, such bounds go back to Lai et al. (1985), but in the MDP setting, they are a recent development (Simchowitz & Jamieson, 2019; Dann et al., 2021; Xu et al., 2021; Jin et al., 2021). Even more recently, similar quantities have been used to characterize instance-dependent PAC bounds (Wagenmaker et al., 2022; Tirinzoni et al., 2022a;b) for policy identification in MDPs. Fine-grained gap-dependent guarantees are largely restricted to tabular algorithms that employ strategic exploration. A notable exception is Dann et al. (2022) who show instance-dependent guarantees for ε-greedy exploration through the notion of policy-based gaps. All of these works consider the single-reward case.

Reinforcement learning with auxiliary tasks has been studied extensively empirically (Jaderberg et al., 2016; Veeriah et al., 2019). Here, the agent aims to learn several additional tasks that are qualitatively different from the main task, to learn better representations that in turn improve performance on the main task. These observations are specific to the function-approximation setting and we are not aware of a theoretical analysis of such approaches.

Several early studies have explored reinforcement learning with multiple rewards. For instance, Shelton (2000) put forward a game-theoretic method to determine a policy in the presence of multiple rewards that may not align on an optimal policy. Lizotte et al. (2010), on the other hand, propose a computationally efficient value iteration algorithm for computing all actions that may be optimal under any convex combination of reward functions. Thus, although these studies delve into the realm of multiple rewards, their primary focus differs from our work. Unlike our research, they do not demonstrate a speed-up of learning with multiple rewards compared to single-reward learning. Furthermore, they do not provide any sample-complexity guarantees, which sets our study apart.

A related setting to ours is that of multi-task (Teh et al.,

2017), transfer and concurrent reinforcement learning. Here, either a single or multiple, possibly concurrently acting agents aim to learn a set of different tasks. In comparison to our setting, the agent aims to learn one policy per task and only receives one reward signal for each interaction. Instead, we receive multiple reward signals per interaction and aim to learn a single policy. RL in multi-task, transfer and concurrent settings has been studied theoretically Brunskill & Li (2013); Guo & Brunskill (2015); Lazaric (2012); Lazaric & Restelli (2011); Hu et al. (2021); Pazis & Parr (2016); Zhang & Wang (2021). Perhaps the closest to our work is Zhang & Wang (2021) who analyze multi-task RL and provide gap-dependent bounds for an optimistic algorithm. Their bounds show that there is a benefit to sharing information between tasks compared to learning them separately but they do not have the resolution to show that learning a single task is faster with multiple rewards.

3. Setup and Notation

We study reinforcement learning with multiple rewards in two settings: that of multi-armed bandit (MAB) and that of tabular Markov decision process (MDP). While we could use the same notation for both, since the MAB setting is a special case of the MDP one, we will use a separate (though largely consistent) notation to aid readability.

3.1. Multi-Armed Bandits with Multiple Rewards

The agent interacts with the environment in rounds. In each round $t \in \mathbb{N}$, the agent chooses an action a_t from the set of available action $\mathcal{A}$ with size $A = |\mathcal{A}|$. The agent then receives a vector r_t of m rewards $r_t^1, \dots, r_t^m$. This vector is drawn from a joint distribution that depends on the chosen action, with means $\mathbb{E}[r_t^i | a_t = a] = \bar{r}^i(a)$ for all $i \in [m]$. Here $\bar{r}^i \colon \mathcal{A} \to [0,1]$ is the i-th (average) reward function. We also assume that the marginal distribution $\mathbb{P}(r_t^i | a_t = a)$ is 1-sub-Gaussian as is common. Importantly, we do not require that the different components of the observed reward are independent (conditioned on a). Thus, an independent "noise" for the different reward functions is not necessary; that would trivially speed up learning, even when all the reward functions coincide, since the effective noise level is then reduced by a factor of $\sqrt{m}$.

For each action $a \in \mathcal{A}$ and reward index $i \in [m]$, we define the gap against the action set $\mathcal{A}$ as $\Delta^i(a; \mathcal{A}) = \max_{a' \in \mathcal{A}} \bar{r}^i(a') - \bar{r}^i(a)$ and entire gap vector as

$$\Delta(a; \mathcal{A}) = \begin{bmatrix} \Delta^1(a; \mathcal{A}) \\ \vdots \\ \Delta^m(a; \mathcal{A}) \end{bmatrix}.$$

We will use the shorthand $\Delta(a) = \Delta(a; \mathcal{A})$ when $\mathcal{A}$ is the set are all actions and will shorten the notation when the comparator set is a single action and write $\Delta(a; b)$ instead of $\Delta(a; \{b\})$. Similarly, the regret after T rounds of an algorithm choosing actions $a_1, \dots, a_T$ is defined for each reward $i \in [m]$ as

$$\mathrm{Reg}^i(T) = T \max_{a \in \mathcal{A}} \bar{r}^i(a) - \sum_{t=1}^{T} \bar{r}^i(a_t),$$

and the regret-vector as

$$\mathrm{Reg}(T) = \begin{bmatrix} \mathrm{Reg}^1(T) \\ \vdots \\ \mathrm{Reg}^m(T) \end{bmatrix}.$$

Throughout the paper, whenever we omit superscript i for the reward function on a quantity v^i, we mean the vector $v = (v^1, \dots, v^m)$. The goal of the algorithm is to achieve small regret with respect to all reward functions. This can mean that the maximum regret under any reward function should be small but, as we will see in the next section, it can also be beneficial to aim for a weaker goal. We will make this precise later when we introduce the operator choice $\mathtt{op}$.

3.2. Markov Decision Processes with Multiple Rewards

We consider episodic tabular Markov decision processes $(\mathcal{X}, \mathcal{A}, P, (\bar{r}^i)_{i \in [m]}, H)$ with state space $\mathcal{X}$, action space $\mathcal{A}$ and horizon H. The number of states and actions are denoted by $S = |\mathcal{X}|$ and $A = |\mathcal{A}|$. The transition kernel $P \colon \mathcal{X} \times \mathcal{A} \to \mathcal{P}(\mathcal{X})$ maps state-action pairs to distributions $\mathcal{P}(\mathcal{X})$ over next states. Deviating from the typical setup, we assume that the process is equipped with m reward functions $\bar{r}^i \colon \mathcal{X} \times \mathcal{A} \to [0,1]$ that each maps state-action pairs to an immediate reward. For ease of notation, we follow the common layered state space assumption, where $\mathcal{X}$ is partitioned into sets $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_{H+1}$ and the transition kernel is such that only $\mathcal{X}_h$ can be visited at level h. We further assume that $\mathcal{X}_{H+1} = \{x_{H+1}\}$ where x_{H+1} is a special state that every policy visits in a virtual level $H+1$, with $\bar{r}^i(x_{H+1}, a) = 0$.

The agent interacts with the environments in T episodes indexed by k. The initial state $x_{k,1}$ is chosen arbitrarily (and possibly adversarially) in each episode. For H timesteps indexed by h, the agent plays an action $a_{k,h}$ and transitions to the next state $x_{k,h+1} \sim P(\cdot | x_{k,h}, a_{k,h})$. For ease of notation, we follow the common assumption that the immediate rewards are directly generated by the reward functions known to the agent.

The agent's action choices in episode k are governed by policy $\pi_k \colon \mathcal{X} \to \mathcal{A}$ which maps states to actions. For our purposes, considering deterministic policies is sufficient. The value (and Q-value) of a policy π at a state $x \in \mathcal{X}_{h'}$

Algorithm 1: Action Elimination in Bandits

Input: operator $\mathtt{op}\colon \mathbb{R}^m \to \mathbb{R}$, failure prob. $\delta \in (0,1)$
Initialize active set $\mathcal{A}_0 = \mathcal{A}$
Initialize $\widehat{r}_0(a) = 0$, $n_0(a) = 0$ for all $a \in \mathcal{A}$
for $t = 1, 2, \dots$ **do**
  Select least played action
  $a_t = \operatorname{argmin}_{a \in \mathcal{A}_t} n_{t-1}(a)$
  Play action a_t and receive reward vector r_t
  Update counters, reward estimates and gaps:

$$n_t(a) = \begin{cases} n_{t-1}(a) + 1 & \text{if } a = a_t \\ n_{t-1}(a) & \text{otherwise} \end{cases}$$

$$\widehat{r}_t(a) = \begin{cases} \frac{1}{n_t(a)} r_t + \frac{n_{t-1}(a)}{n_t(a)} \widehat{r}_{t-1}(a) & \text{if } a = a_t \\ \widehat{r}_{t-1}(a) & \text{otherwise} \end{cases}$$

$$\widehat{\Delta}_t(a; \mathcal{A}_t) = \begin{bmatrix} \max_{b \in \mathcal{A}_{t-1}} \widehat{r}^1_t(b) - \widehat{r}^1_t(a) \\ \vdots \\ \max_{b \in \mathcal{A}_{t-1}} \widehat{r}^m_t(b) - \widehat{r}^m_t(a) \end{bmatrix}$$

  Eliminate suboptimal actions from active set

$$\mathcal{A}_{t+1} = \left\{ a \in \mathcal{A}_t \colon \mathtt{op}\left(\widehat{\Delta}_t(a; \mathcal{A}_t)\right) \le c' \sqrt{\frac{\ln \frac{mA \ln(n_t(a))}{\delta}}{n_t(a)}} \right\}$$

(and action a) with respect to reward $\bar{r}^i$ is defined as

$$V^{\pi,i}(x) = \mathbb{E}^\pi \left[\sum_{h=h'}^{H} \bar{r}^i(x_h, a_h) \mid x_{h'} = x \right]$$

$$Q^{\pi,i}(x,a) = \mathbb{E}^\pi \left[\sum_{h=h'}^{H} \bar{r}^i(x_h, a_h) \mid x_{h'} = x, a_{h'} = a \right],$$

where $\mathbb{E}^\pi$ denotes the expectation over trajectories $x_1, a_1, \dots, x_{H+1}$ that are generated by following π. We denote by $\pi^\star_i$ an optimal policy for reward $\bar{r}^i$ and by $Q^{\star,i} = Q^{\pi^\star_i,i}$ and $V^{\star,i} = V^{\pi^\star_i,i}$ the optimal Q- and state-value function respectively. The suboptimality associated with a state-action pair can be formalized by the *value-function gap*

$$\Delta^i(x,a) = V^{\star,i}(x) - Q^{\star,i}(x,a).$$

There are other, more accurate characterizations of suboptimality that govern regret and sample-complexity (Dann et al., 2021; Tirinzoni et al., 2022b) but we adopt this most common notion of gap for ease of comparison.

4. Multi-Reward Action Elimination in Multi-Armed Bandits

Multiple reward functions can be naturally incorporated in various algorithms. We find, however, that action elimination algorithms are particularly suited to multi-reward learning. We therefore present and study a natural extension of action elimination (Even-Dar et al., 2006) in Algorithm 1.

This algorithm takes as input an operator $\mathtt{op}$ that maps m values to a scalar. First consider the case where $\mathtt{op}$ is the maximum operator, thus, $\mathtt{op}(v) = \max\{v_1, v_2, \dots, v_m\}$. Algorithm 1 maintains an active set of actions and selects always the least often played action to ensure all active actions are roughly played equally often. It then receives the reward vector, updates an empirical average reward per reward function and arm as well as an estimate of the gap. It then eliminates an action as soon as it is sure that an action has a positive gap *under any reward function* by comparing the maximum gap of an action to a confidence term.

Since an elimination happens as soon as a gap is detected under any reward function, it is clear that the algorithm may eliminate actions faster than had it just used one single reward function. Further, different actions may be eliminated due to being highly suboptimal for different rewards. This suggests that learning with multiple rewards can be more sample-efficient than just learning with the best among the m rewards. Our analysis will indeed confirm this.

However, aggressively eliminating arms based on evidence from any reward function can also be harmful if not all rewards agree with each other on an optimal policy. In this case, an optimal arm according to one reward can could be eliminated because it is suboptimal under another. To allow for a more moderate elimination, we allow the user to specify an operator $\mathtt{op}$ as input to the algorithm. As we will see in our regret bound, using a different operator than $\max$ may be beneficial. This operator has to satisfy the following assumption.

Assumption 1. The operator $\mathtt{op}\colon \mathbb{R}^m \to \mathbb{R}$ is a *coherent risk measure*,[1] that is, it is monotone, positively homogeneous, sub-additive and translationally invariant:

- for any $x, y \in \mathbb{R}^m$ with $x \le y$, $\mathtt{op}(x) \le \mathtt{op}(y)$;
- for any $x \in \mathbb{R}^m, \alpha \in \mathbb{R}_+$, $\mathtt{op}(\alpha x) = \alpha\, \mathtt{op}(x)$;
- for any $x, y \in \mathbb{R}^m$, $\mathtt{op}(x+y) \le \mathtt{op}(x) + \mathtt{op}(y)$;
- for any $\alpha \in \mathbb{R}, x \in \mathbb{R}^m$, $\mathtt{op}(x + \alpha \mathbf{1}) = \mathtt{op}(x) + \alpha$.

The most prominent examples for operators that satisfy this assumption is the class of CVaR risk measures, with $\max$ and mean as important special cases.

Example 1 (CVaR with average and maximum as special cases)**.** *A popular coherent risk measure is conditional value at risk (CVaR).*

$$\mathrm{CVaR}_\alpha(x) = \min_{c \in \mathbb{R}} \left\{ c + \frac{1}{1-\alpha} \frac{1}{n} \sum_{i=1}^{n} [x_i - c]_+ \right\}.$$

[1] Strictly speaking, $\mathtt{op}$ is the negative of the common definition of a coherent risk measure.

For $\alpha = 0$, this matches mean *and for $\alpha = 1$, it matches* max.

Importantly, quantiles are generally not sub-additive and therefore do not satisfy Assumption 1. This is well known for value-at-risk and easy to see for the special case of min (which is supra-additive) since for example $\min\{1+2, 2+1\} \not\leq \min\{1,2\} + \min\{2,1\}$.

Under this assumption on the operator, we prove the following main result for Algorithm 1.

Theorem 1. *Let ε be a threshold that can chosen as small as the suboptimality of the best action under $\mathtt{op}$ that satisfies Assumption 1,*

$$\varepsilon \geq \min_{a \in \mathcal{A}} \mathtt{op}(\Delta(a)),$$

and let $\mathcal{A}^\varepsilon = \{a \in \mathcal{A} \colon \mathtt{op}(\Delta(a)) > \varepsilon\}$ be the set of actions that are worse than ε. Then, with probability at least $1-\delta$, the regret of Algorithm 1 satisfies

$$\mathtt{op}(\mathrm{Reg}(T)) = O\left(\sum_{a \in \mathcal{A}^\varepsilon} \frac{1}{\mathtt{op}(\Delta(a))} \ln \frac{mAT}{\delta} + T\varepsilon\right).$$

This guarantee bounds the $\mathtt{op}$-operator of the regret vector, that is, for example, the maximum or average regret under the different reward functions. Our bound consists of a $\ln(T)$-term which captures how quickly suboptimal arms can be eliminated and a linear T-term for the asymptotic suboptimality. When all reward functions agree, we can choose $\varepsilon = 0$ and the linear T term vanishes. Otherwise, we can choose ε as the smallest operator-gap which we have to accept as a factor in the linear regret component. This is unimprovable since there is no single arm that achieves better instantaneous regret (aggregated according to $\mathtt{op}$). Importantly, our result shows that action elimination is robust towards disagreement among the reward functions.

We can now also see why it may be beneficial to choose a different operator than max. Take the example where the best action is optimal under all but one reward function and it has a gap of α there. Then with $\mathtt{op} = \max$, we have to choose $\varepsilon \geq \alpha$ and tolerate αT operator-regret. On the other hand, for $\mathtt{op} = \text{mean}$, we can choose $\varepsilon = \frac{\alpha}{m}$ and only accept $\frac{\alpha T}{m}$ regret contribution. Of course, this comes at the price that the inverse effective gaps may be smaller and the notion of operator regret $\mathtt{op}(\mathrm{Reg}(T))$ itself is weaker.

Comparing our bound against the standard guarantee for action elimination with a single reward $\bar{r}^i$ of

$$O\left(\sum_{a \in \mathcal{A}^\varepsilon} \frac{1}{\Delta^i(a)} \ln \frac{mAT}{\delta} + T\varepsilon\right),$$

where $\varepsilon \geq 0$, we see that our bounds scale inversely with the operator-gap which can be much more favorable compared to the gap of the single reward. This is most apparent for $\mathtt{op} = \max$, where the denominator is the largest gap of the action among all reward functions. While our guarantee is often preferable, the bound in Theorem 1 can sometimes be worse than the bound for learning with a single reward $\bar{r}^i$, due to the linear regret from disagreement of the rewards on the optimal reward and cases where $\mathtt{op}(\Delta(a)) > \Delta^i(a) = 0$. Fortunately, for $\mathtt{op} = \max$ and when there is an arm that is optimal under all rewards, we can show a stronger version of Theorem 1:

Theorem 2. *Assume that $\mathtt{op} = \max$ and there is an action $a^\star$ that is optimal under all reward functions, that is, $\mathtt{op}(\Delta(a^\star)) = 0$. Then with probability at least $1-\delta$, the regret vector of Algorithm 1 satisfies*

$$\mathrm{Reg}(T) = O\left(\sum_{a \in \mathcal{A}^\varepsilon} \frac{\Delta(a)}{\max(\Delta(a))^2} \ln \frac{mAT}{\delta} + T\varepsilon\right),$$

for all $\varepsilon \geq 0$.

This regret guarantee for any individual reward $\bar{r}^i$ is never worse than the guarantee one obtains from learning with only one reward and much smaller in many cases. It is also instructive to compare this result against single-reward learning with the average of all rewards $\mathrm{mean}(\bar{r})$. This would yield

$$\mathrm{mean}(\mathrm{Reg}(T)) = O\left(\sum_{a \in \mathcal{A}^\varepsilon} \frac{1}{\mathrm{mean}(\Delta(a))} \ln \frac{mAT}{\delta} + T\varepsilon\right).$$

This bound is generally worse than that in Theorem 2 and if we want a bound for a specific reward $\bar{r}^i$ from the mixture, we have to pay another factor of m since $\mathrm{Reg}^i(T) \leq m \cdot \mathrm{mean}(\mathrm{Reg}(T))$. All in all, this shows that learning with multiple rewards is often beneficial, as long as there is some consensus among the rewards on the optimal policy.

4.1. Illustrative Examples

To illustrate the gains from learning with multiple rewards, we provide two simple examples from a Bayesian perspective where the different reward functions are drawn from distributions that always agree on the optimal action but have random preferences for different arms. Such a perspective is appropriate when different rewards come from independent sources.

Example 2. *Each of the m reward functions comes from an independent reward designer who always associates a reward of 1 with the action 1. For each other action, with probability $1/2$ she gives a slightly suboptimal reward $1 - 1/\sqrt{T}$ and with probability $1/2$ a reward of 0. Then the expected regret bound (over the designer's randomness) with a single reward function is of order*

$$\sqrt{T} A \ln(AT).$$

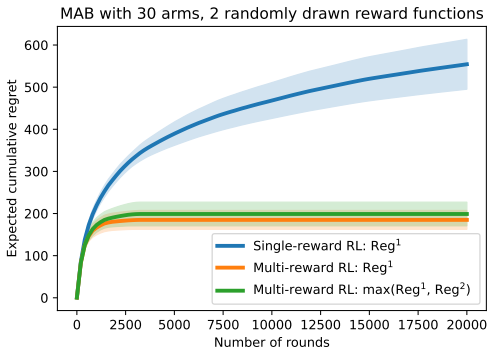

Figure 1. Expected regret of single- and multi-reward RL in Example 3. We used 20 independent runs of Algorithm 1, each on randomly drawn reward functions. Shaded areas are 95% confidence bands. For single-reward RL, we only expose the first reward function to the algorithm, for multi-reward both. Our simulations corroborate our theoretical findings: multi-reward RL incurs significantly less expected regret in this construction.

In contrast, with m independently designed rewards, we have an expected bound on the regret of order

$$\left[\frac{\sqrt{T}}{2^m}+1\right] A\ln(mAT).$$

Thus, with $m=\Omega(\ln(T))$ reward functions, the expected bound for RL with multiple rewards becomes $A\ln(AT)$ which improves the rate in the single reward case by $\sqrt{T}$.

The example above is designed to illustrate the largest possible improvement, from the worst-case regret rate $\sqrt{T}$ to the most favorable $\ln(T)$ where all effective gaps are constant. However, even if we just take a uniform random distribution over rewards from each designer and only two reward functions, we already observe (a more modest) improvement, as the following example illustrates:

Example 3. *Assume there are two reward functions $m=2$. Action 1 has reward 1 in both reward functions. For each action i, the expected reward in reward function is selected independently uniformly at random in $[0,1]$. The expected regret due to action i is $\int_0^1(\log(T)/z)(2zdz)=\Theta(\ln(T))$. (Note that $\Pr[\ell<z]=z^2$.) On the other hand, for a single reward function we have $\int_0^1\min\{(\log(T)/z),zT\}dz=\Theta(\ln^2(T))$. Thus, the second reward function decreases the regret bound by a $\ln(T)$ factor. We also simulate this example in Figure 1.*

4.2. Model-Selection for Operator?

Our results show that the regret of Algorithm 1 degrades gracefully as the disagreement among reward functions $\rho_{\mathrm{op}}=\min_{a\in\mathcal{A}}\mathrm{op}(\Delta(a))$ increases. Fortunately, the algorithm does not need to know the disagreement ρ_{op} for the guarantee to hold and naturally adapts to it. However, the algorithm does take a hyperparameter, the operator op. As discussed above, the operator controls how aggressively the algorithm eliminates arms and determines the *effective gap* or *operator gap* $\mathrm{op}(\Delta(a))$ and thus also ρ_{op}. Thus, we want to choose an operator that yields the tightest regret bound in Theorem 1, optimally trading off ρ_{op}, which determines how small we can choose ε, and $\mathrm{op}(\Delta(a))^{-1}$ for suboptimal arms.

One may wonder whether we can automatically select the best operator from a class using online model selection techniques (e.g. Agarwal et al., 2017; Pacchiano et al., 2020; Arora et al., 2021; Cutkosky et al., 2021). A natural class is for example the family of CVaR_α operators for varying α from Example 1. Another family that is natural in many scenarios are $\max$ operators with different support. Consider the case where we have an ordered list of rewards $\bar{r}^1,\dots\bar{r}^m$, where $\bar{r}^1$ is the reward we are primarily interested in and $\bar{r}^2,\dots\bar{r}^m$ are alternative rewards that, with decreasing confidence, agree with $\bar{r}^1$ on the optimal policy. Then a natural choice is to select an operator from $\max_1,\dots,\max_m$ where $\max_i(v)=\max\{v^1,v^2,\dots,v^i\}$ only considers the first i dimensions of the input.

Unfortunately, we will provide an example that illustrates that model selection for the operator is in general not possible without sacrificing our sharp guarantees from Theorem 1. The reason for this is related to the barrier of model selection in regimes below $\sqrt{T}$ observed in prior work (Pacchiano et al., 2020; 2022). Consider two bandit instances indexed by $I\in\{-1,+1\}$ with arms $\mathcal{A}=\{1,2\}$ and rewards

$$\bar{r}(1)=\begin{bmatrix}\frac{1}{2}\\ \frac{1}{2}\end{bmatrix}\qquad \bar{r}(2)=\begin{bmatrix}\frac{1}{2}-\mathrm{sign}(I)\cdot\gamma\\ 0\end{bmatrix},$$

where $\gamma=T^{-1/3}$. Thus, in instance $I=+1$, the first arm is optimal under both reward functions and the second arm has gaps $\Delta_{+1}(2)=[\gamma,1/2]$. In instance $I=-1$, the two reward functions disagree on the optimal arm and induce gaps $\Delta_{-1}(1)=[\gamma,0]$ and $\Delta_{-1}(2)=[0,1/2]$. Assume that we are interested in choosing from two operators $\max_1$ and $\max_2$. According to Theorem 1, using each operator in each instance yields an expected regret rate of

	$\max_1$	$\max_2$
$I=+1$	$T^{1/3}\ln(T)$	$\ln(T)$
$I=-1$	$T^{1/3}\ln(T)$	$T^{2/3}$

Thus, using model selection, we would like to achieve logarithmic regret in instance $+1$ under both reward functions and still get regret $o(T^{2/3})$ for reward $\bar{r}^1$ in instance $I=-1$. To achieve expected regret at most $O(\ln(T))$ under reward $\bar{r}^2$ in $I=+1$, the expected number of times an algorithm can choose action 2 is at most $O(\ln(T))$. This, however, is not sufficient to distinguish between $I=+1$ and $I=-1$

with high probability. As a result, the algorithm also has to play action 2 at most $O(\ln(T))$ times with constant probability in $I = -1$. Therefore, the expected number of plays of action 1 in instance $I = -1$ is $\Omega(T)$, which yields a regret of $\Omega(\gamma T) = \Omega(T^{2/3})$ according to $\bar{r}^1$. This argument shows that no algorithm can achieve $\mathbb{E}\,\mathrm{Reg}^1_{-1}(T) = o(T^{2/3})$ while maintaining $\mathbb{E}\,\mathrm{Reg}^2_{+1}(T) = O(\ln(T))$. We can therefore not hope to achieve the guarantee of Theorem 1 for several operators simultaneously.

5. Multi-Reward Action Elimination in MDPs

We now extend our results from the multi-armed bandit setting to episodic Markov decision processes. Here too, we find that action-elimination is the most suitable technique. We consider a multi-reward extension of the AMB (Adaptive Multistep Boostrap) algorithm by Xu et al. (2021) shown in Algorithm 2.

This algorithm maintains upper and lower confidence bounds $\overline{Q}_k$, $\underline{Q}_k$ on the optimal Q-function under each reward, as well as $\overline{V}_k$, $\underline{V}_k$ for the optimal value function $V^\star$. To collect data in episode k, the algorithm chooses, in each state x, the action a from the active set $\mathcal{A}_k(x)$ that has the largest uncertainty on $Q^\star$, that is, where $\overline{Q}_{k-1}(x,a) - \underline{Q}_{k-1}(x,a)$ is largest. Since the confidence bounds and their difference can vary across reward functions, the algorithm uses the given operator op to combine the uncertainty measures and select $\mathrm{argmax}_{a\in\mathcal{A}_k(x)} \mathrm{op}(\overline{Q}_k(x,a) - \underline{Q}_k(x,a))$.

After collecting the data, the algorithm updates the confidence bounds on the optimal Q- and value function for each reward function in a model-free manner, following the single reward algorithm (Xu et al., 2021). Finally, to determine the active sets $\mathcal{A}_{k+1}(x)$ for the next episode, the algorithm only retains actions that satisfy $\mathrm{op}(\underline{V}_k(x) - \overline{Q}_k(x,a) - \varepsilon) \le 0$. To gain intuition for this criterion, first consider the case where $\varepsilon = \mathbf{0}$ and $\mathrm{op} = \max$. An action then gets eliminated as soon as $\underline{V}^i_k(x) - \overline{Q}^i_k(x,a) > 0$ holds for any reward functions i. Since $\underline{V}^i_k(x) - \overline{Q}^i_k(x,a)$ is a lower confidence bound on $\Delta^i(x,a) = V^{\star,i}(x) - Q^{\star,i}(x,a)$, this condition tests, just as in the bandit case, whether the gap associated with an action is strictly positive.

The hyperparameter ε quantifies how much the individual rewards agree on an optimal policy. When there is a policy that is optimal in all (reachable) states for all reward functions, then we can set $\varepsilon = \mathbf{0}$. Otherwise, ε_i specifies how much expected reward can be lost according to reward $\bar{r}^i$, when taking an action that is optimal under another reward function. The following assumption formalizes this:

Assumption 2. For each reward function, there is an optimal policy $\pi^\star_i \in \Pi$ such that for all $x \in \mathcal{X}, j \in [m]$

$$\Delta^j(x, \pi^\star_i(x)) \le \begin{cases} 0 & \text{if } i = j \\ \varepsilon_j & \text{otherwise.} \end{cases}$$

Since there are H steps per episode, this assumption implies that $\pi^\star_i$ is at most $H \times \varepsilon_j$ suboptimal under reward $\bar{r}_j$. Algorithm 2 receives $\varepsilon \in \mathbb{R}^m$ as an input and uses it as an additional slack when testing for a positive gap. This ensures that only actions with $\mathrm{op}(\Delta(x,a) - \varepsilon) > 0$ can be eliminated and the actions of all policies $\pi^\star_i$ from Assumption 2 always survive.

In the bandit case, Algorithm 1 did not need the additional slack. There, we could argue that if the best action (the one with the smallest operator-gap) is eliminated, then only actions can survive that have the same operator gap up to a constant factor. So the algorithm will incur the best possible regret from that point on, again up the a constant factor. In the MDP case, there is the additional challenge that the gap at level h depends on the optimal behavior in later layers through the optimal value functions. If we eliminate all actions that are optimal for some reward functions, then we will underestimate the optimal value for that policy in earlier states. Accounting for such errors with known arguments unfortunately would yield an exponential in H error. We therefore opted for the additional slack of size ε in Algorithm 2. Note that related work on multi-task RL (Zhang & Wang, 2021) also assumes knowledge of the equivalent quantity in that setting (similarity of tasks).

Our main result in the MDP setting is:

Theorem 3. *If Assumption 2 holds for some $\varepsilon \in \mathbb{R}^m$ and op is a coherent risk measure, then Algorithm 2 with parameters op, ε and δ has, with probability at least $1-\delta$, expected operator-regret*

$$\tilde{O}\left(\sum_{x,a} \frac{H^5 \ln(mSAT/\delta)}{\max\{\mathrm{op}(\varepsilon), \mathrm{op}(\Delta(x,a)), \Delta_2^{\mathrm{op}}(x)\}} + TH\varepsilon\right),$$

where $\Delta_2^{\mathrm{op}}(x)$ is the second-smallest operator-gap in state x, that is, the second-smallest entry of $\mathrm{op}(\Delta(x,\cdot))$.

Thus, for each state-action pair, we always pay the inverse of its operator gap, second smallest operator gap in that state or just $\mathrm{op}(\varepsilon)$, whichever is most favorable. To ease the comparison with the single reward case, we again look at the case where $\mathrm{op} = \max$, $\varepsilon = 0$ and where there is a unique optimal action in each state. Then, our bound evaluates to

$$\tilde{O}\left(\sum_{x,a} \frac{H^5 \ln(mSAT/\delta)}{\max\{\max(\Delta(x,a)), \Delta_2^{\max}(x)\}}\right)$$

Algorithm 2: Multi-Reward Action Elimination in Tabular MDPs

Input: m reward functions $(\bar{r}^i)_{i\in[m]}$, operator op, slack parameters $(\varepsilon_i)_{i\in[m]}$, failure prob. δ
Initialize $\underline{Q}_0^i(x,a)=0$, $\overline{Q}_0^i(x,a)=H+1-h$ for all $x\in\mathcal{X}_h$, $a\in\mathcal{A}$, $h\in[H]$
Initialize $\underline{V}_0^i(x_{H+1})=\overline{V}_0^i(x_{H+1})=0$
Initialize $\mathcal{G}_1=\varnothing$ and $\mathcal{A}_1(x)=\mathcal{A}$ for all $x\in\mathcal{X}$
Set learning rate $\alpha_k=\frac{H+1}{H+k}$ for all $k\in\mathbb{N}$ and bonuses $b_n=c\sqrt{\frac{H^3\ln(mSAHK/\delta)}{n}}$ with a universal const. c for all $n\in\mathbb{N}$
for $k=1,2,3,\dots,T$ **do**
  Collect data: sample one episode $x_{k,1},a_{k,1},x_{k,2},a_{k,2},\dots,x_{k,H+1}$ with policy

$$\pi_k(x)\in\operatorname*{argmax}_{a\in\mathcal{A}_k(x)}\mathrm{op}\left(\overline{Q}_{k-1}(x,a)-\underline{Q}_{k-1}(x,a)\right)\qquad\forall x\in\mathcal{X}\tag{1}$$

  for $h=H,H-1,\dots,1$ **do**
    if $x_{k,h}\in\mathcal{G}_k$ **then**
      continue
    Let $n=n_k(x_{k,h},a_{k,h})$ be the number of visits to $(x_{k,h},a_{k,h})$
    Let $x_{k,h'}$ be the first state after $x_{k,h}$ that is not in $\mathcal{G}_k$
    Perform regular Q-function update for each reward function:
    for $i\in[m]$ **do**
      Set $\widehat{R}_k^i(x_{k,h},a_{k,h})=\sum_{j=h}^{h'-1}\bar{r}^i(x_{k,j},a_{k,j})$
      $\underline{Q}_k^i(x_{k,h},a_{k,h})=\max\{0,(1-\alpha_n)\underline{Q}_{k-1}^i(x_{k,h},a_{k,h})+\alpha_n(\widehat{R}_k^i(x_{k,h},a_{k,h})+\underline{V}_{k-1}^i(x_{k,h'})-b_n)\}$
      $\overline{Q}^ik(x_{k,h},a_{k,h})=\min\{H-h+1,(1-\alpha_n)\overline{Q}_{k-1}^i(x_{k,h},a_{k,h})+\alpha_n(\widehat{R}_k^i(x_{k,h},a_{k,h})+\overline{V}_{k-1}^i(x_{k,h'})+b_n)\}$
      $\underline{V}_k^i(x_{k,h})=\max_{a\in\mathcal{A}_k(x_{k,h})}\underline{Q}_k^i(x_{k,h},a)$
      $\overline{V}_k^i(x_{k,h})=\max_{a\in\mathcal{A}_k(x_{k,h})}\overline{Q}_k^i(x_{k,h},a)$
  for $(x,a)\in(\mathcal{X}\times A_k(x))\setminus\{x_{k,h},a_{k,h}\}_{h\in[H]}$ **do**
    $\underline{Q}_k(x,a)=\underline{Q}_{k-1}(x,a)$, $\overline{Q}_k(x,a)=\overline{Q}_{k-1}(x,a)$
    $\underline{V}_k(x)=\underline{V}_{k-1}(x)$, $\overline{V}_k(x)=\overline{V}_{k-1}(x)$
  Eliminate Actions
  Set $\mathcal{A}_{k+1}(x)=\{a\in\mathcal{A}_k(x)\colon 0\geq\mathrm{op}(\underline{V}_k(x)-\overline{Q}_k(x,a)-\varepsilon)\}$ for all $x\in\mathcal{X}$
  Set $\mathcal{G}_{k+1}=\{x\in\mathcal{X}\colon|\mathcal{A}_{k+1}(x)|=1\}$

and the bound for learning with a single reward $\bar{r}^i$ is (Xu et al., 2021)

$$\tilde{O}\left(\sum_{x,a}\frac{H^5\ln(SAT/\delta)}{\max\{\Delta^i(x,a),\Delta_2^i(x)\}}\right)$$

where $\Delta_2^i(x)$ is the smallest nonzero gap according to $\bar{r}^i$ in x. Just as in the bandit case, we see that we benefit from the largest gap across all reward functions for each state-action pair. Hence, as long as there is agreement among reward functions on an optimal policy, $\varepsilon=0$, multi-reward RL enjoys more favorable guarantees compared to using a single reward function.

The proof of Theorem 1 in the appendix builds largely on the single-reward analysis. However, naively applying the operator op to various parts of the existing analysis would require exchanging max and op at some point, which only works for op = max. To handle the $\mathrm{op}\neq\max$ case, we first had to carefully decompose the op-gap before applying the max. Further, when $\varepsilon>0$, we effectively deal with a misspecified model. Although common in UCB analyses, misspecification was absent from existing action elimination theory and required particular care in our proofs.

5.1. Illustrative Example

We now provide an example of an MDP and optimal policy where the structure of the MDP is such that the effective gaps for multiple rewards are much larger, compared to the gaps of *any individual reward*, due to the structure of the MDP. Let $n\in\mathbb{N}$ and consider an MDP with $|\mathcal{X}_h|=n$ states on each of the $H=n$ layers. We denote the j-th state in layer h by $x^{j,h}$. All states in layers $h\geq 2$ have a single action which transitions the agent from $x^{j,h}$ to $x^{j,h+1}$, that is, states

form n chains of length $H-1$. States $x^{1,1},\dots,x^{n-1,1}$ in the initial layer have two actions 1 and 2 that transition the agent from $x^{j,1}$ to $x^{j,2}$ and $x^{j+1,2}$ respectively. Initial states are chosen uniformly at random from this set. We would like the policy that takes action 1 in all states to be optimal.

Consider a single reward function r' and denote by v'_j the value that it associates with state $x^{j,2}$ and by $\Delta'\colon \mathcal{X}\times\mathcal{A}\to\mathbb{R}$ its associated gap function. The only relevant gaps in this example are $\Delta'(x^{j,1},2)$ for $j\in[n-1]$ which satisfy

$$\Delta'(x^{j,1},2)\le v'_j - v'_{j+1} + 1,$$

since the reward function may assign an immediate reward of 0 to action 2 and 1 to action 1. Furthermore, since rewards are normalized in $[0,1]$, we have $v'_j\in[0,H-1]=[0,n-1]$. Considering the sum of relevant inverse gaps that govern the regret (with a $\ln(T)$ factor) in this problem. By convexity of $x\mapsto 1/x$, we have

$$\sum_{j=1}^{n-1}\frac{1}{\Delta'(x^{j,1},2)}\ge\frac{(n-1)^2}{\sum_{j=1}^{n-1}\Delta'(x^{j,1},2)}\ge\frac{(n-1)^2}{2n}.$$

On the other hand, consider multi-reward RL with $\mathrm{op}=\max$ and $n-1$ reward functions $\bar{r}^i$ with their respective gaps Δ^i and values v^i_j at level 2. Each $\bar{r}^i$ assigns a reward of 1 and 0 to actions 1 and 2 in the first layer respectively. It further assigns values $v^i_j=\mathbb{1}\{j\le i\}(H-1)$ to the second layer. We then have $\max(\Delta(x^{j,1},2))=H=n$ and the sum of inverse effective gaps are

$$\sum_{j=1}^{n-1}\frac{1}{\mathrm{op}\,\Delta(x^{j,1},2)}=\frac{n-1}{n}.$$

Hence, there is a difference of $\Omega(n)$ between the regret bound for RL with any arbitrary individual reward on this problem and the bound for RL with multiple rewards. Note that if we compare the two in terms of sample complexity (number of times a suboptimal arm was chosen), then the difference becomes even larger $\Omega(n^2)$ since we pay roughly constant regret when choosing action 2 according to the individual reward vs. n in the multi-reward case. This example shows that the guarantees for multi-reward RL can be substantially better in MDPs, even when compared to RL with the single best reward function.

We validate our theoretical findings by simulating this example. We compare the performance of single-reward RL against multi-reward RL for different values of n. As a single-reward algorithm, we use Algorithm 2 but where we only expose the reward function r^1 that assigns a value $v_j=j-1$ to $x^{j,2}$. Our computation above suggests that this is the optimal choice for a single reward. For multi-reward RL we use Algorithm 2 with n rewards: the single reward function just described and in addition the $n-1$ reward functions $\bar{r}^i$ from above. We compare the total regret after $100k$ steps in Figure 2. We observe a substantially faster learning when we compare the performance of single- and multi-reward RL measured by the regret in the first reward function (r^1 that both have access to). Even if we consider the largest regret under any of the reward functions for multi-reward RL, we still observe a factor of 2 improvement in regret when $n\ge 20$.

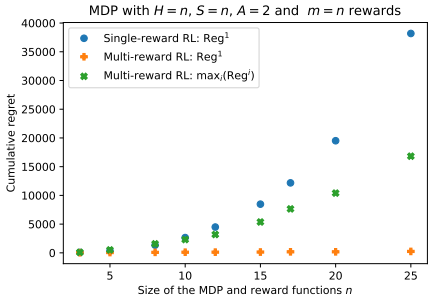

Figure 2. Simulation of the example in Section 5.1.

6. Conclusion

We have shown that directly incorporating different rewards for the same task into the reinforcement learning algorithm can significantly speed up learning. We have studied this both in multi-armed bandits and tabular MDPs and provided improved regret bounds for action-elimination based algorithms. We further illustrated with several examples the benefits of multiple rewards. Notably, for MDPs, depending on the transition structure and the optimal policy, we can achieve guarantees for multi-reward RL that are strictly better than what is possible with any single reward.

To the best of our knowledge, this is the first work to show provable benefits for incorporating multiple rewards for the same task. It motivates several directions for future research. We focused, for example, on action-elimination algorithms due to their simplicity but it would be desirable to adapt other algorithms, e.g. optimism-based ones, to work with multiple rewards for provably faster learning. This would also be a good step towards studying this setting in combination with function approximation.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation (grant number 993/17), the Yandex Initiative for Machine Learning at Tel Aviv University and a grant from the Tel Aviv University Center for AI and Data Science (TAD).

References

Agarwal, A., Luo, H., Neyshabur, B., and Schapire, R. E. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pp. 12–38. PMLR, 2017.

Arora, R., Marinov, T. V., and Mohri, M. Corralling stochastic bandit algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 2116–2124. PMLR, 2021.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Brunskill, E. and Li, L. Sample complexity of multi-task reinforcement learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 122–131, 2013.

Chentanez, N., Barto, A., and Singh, S. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17, 2004.

Cutkosky, A., Dann, C., Das, A., Gentile, C., Pacchiano, A., and Purohit, M. Dynamic balancing for model selection in bandits and rl. In *International Conference on Machine Learning*, pp. 2276–2285. PMLR, 2021.

Dann, C., Marinov, T. V., Mohri, M., and Zimmert, J. Beyond value-function gaps: Improved instance-dependent regret bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 1–12, 2021.

Dann, C., Mansour, Y., Mohri, M., Sekhari, A., and Sridharan, K. Guarantees for epsilon-greedy reinforcement learning with function approximation. In *International Conference on Machine Learning*, pp. 4666–4689. PMLR, 2022.

Even-Dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

Guo, Z. and Brunskill, E. Concurrent pac rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Hu, J., Chen, X., Jin, C., Li, L., and Wang, L. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pp. 4349–4358. PMLR, 2021.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Jin, T., Huang, L., and Luo, H. The best of both worlds: stochastic and adversarial episodic mdps with unknown transition. *Advances in Neural Information Processing Systems*, 34:20491–20502, 2021.

Lai, T. L., Robbins, H., et al. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6 (1):4–22, 1985.

Lazaric, A. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pp. 143–173. Springer, 2012.

Lazaric, A. and Restelli, M. Transfer from multiple mdps. *Advances in neural information processing systems*, 24, 2011.

Lizotte, D. J., Bowling, M. H., and Murphy, S. A. Efficient reinforcement learning with multiple reward functions for randomized controlled trial analysis. In *ICML*, volume 10, pp. 695–702, 2010.

Luo, S., Kasaei, H., and Schomaker, L. Accelerating reinforcement learning for reaching using continuous curriculum learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Mataric, M. J. Reward functions for accelerated learning. In *Machine learning proceedings 1994*, pp. 181–189. Elsevier, 1994.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287, 1999.

Pacchiano, A., Phan, M., Abbasi Yadkori, Y., Rao, A., Zimmert, J., Lattimore, T., and Szepesvari, C. Model selection in contextual stochastic bandit problems. *Advances in Neural Information Processing Systems*, 33:10328–10337, 2020.

Pacchiano, A., Dann, C., and Gentile, C. Best of both worlds model selection. In *Advances in Neural Information Processing Systems*, 2022.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

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

Shelton, C. Balancing multiple sources of reward in reinforcement learning. *Advances in Neural Information Processing Systems*, 13, 2000.

Simchowitz, M. and Jamieson, K. G. Non-asymptotic gap-dependent regret bounds for tabular mdps. *Advances in Neural Information Processing Systems*, 32, 2019.

Sorg, J., Lewis, R. L., and Singh, S. Reward design via online gradient ascent. *Advances in Neural Information Processing Systems*, 23, 2010.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Tirinzoni, A., Al-Marjani, A., and Kaufmann, E. Near instance-optimal pac reinforcement learning for deterministic mdps. *arXiv preprint arXiv:2203.09251*, 2022a.

Tirinzoni, A., Al-Marjani, A., and Kaufmann, E. Optimistic pac reinforcement learning: the instance-dependent view. *arXiv preprint arXiv:2207.05852*, 2022b.

Veeriah, V., Hessel, M., Xu, Z., Rajendran, J., Lewis, R. L., Oh, J., van Hasselt, H. P., Silver, D., and Singh, S. Discovery of useful questions as auxiliary tasks. *Advances in Neural Information Processing Systems*, 32, 2019.

Wagenmaker, A. J., Simchowitz, M., and Jamieson, K. Beyond no regret: Instance-dependent pac reinforcement learning. In *Conference on Learning Theory*, pp. 358–418. PMLR, 2022.

Xu, H., Ma, T., and Du, S. Fine-grained gap-dependent bounds for tabular mdps via adaptive multi-step bootstrap. In *Conference on Learning Theory*, pp. 4438–4472. PMLR, 2021.

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

Zheng, Z., Oh, J., Hessel, M., Xu, Z., Kroiss, M., Van Hasselt, H., Silver, D., and Singh, S. What can learned intrinsic rewards capture? In *International Conference on Machine Learning*, pp. 11436–11446. PMLR, 2020.

Contents of Appendix

A. Proofs for Multi-Armed Bandits

A.1. Proof of Theorem 1

Proof of Theorem 1. We first write the regret using sub-additivity of op as

$$\operatorname{op}\left(\operatorname{Reg}(T)\right) = \operatorname{op}\left(\sum_{t=1}^{T} \Delta(a_t)\right) \leq \sum_{t=1}^{T} \operatorname{op}\left(\Delta(a_t)\right)$$

Let $a^\star \in \operatorname{argmin}_{a \in \mathcal{A}} \operatorname{op}(\Delta(a))$ an action with minimal operator-gap. Further let $T^\star$ be the round when $a^\star$ gets eliminated and $T' = \min\{T^\star, T\}$. For rounds up to T' where some actions $a \in \mathcal{A}^\varepsilon$ was played, its regret contribution is $n_{T'}(a) \operatorname{op} \Delta(a)$ which can be upper-bounded by $n_{T'}(a) \operatorname{op} \Delta(a; a^\star) + n_{T'}(a) \operatorname{op} \Delta(a^\star)$. For convenience, we denote by $l_t = \ln(mA\ln(t)/\delta)$ the log-term in the UCB after t observations. We then have for every round $t \in [T']$ and $a \in \mathcal{A}_{t+1}$

$$\begin{aligned}
\operatorname{op}(\Delta(a; a^\star)) &\leq \operatorname{op}\left(\widehat{\Delta}(a; a^\star) + c\sqrt{\frac{l_{n_t(a)}}{n_t(a)}}\right) && \text{(Lemma 1)} \\
&= \operatorname{op}\left(\widehat{\Delta}(a; a^\star)\right) + c\sqrt{\frac{l_{n_t(a)}}{n_t(a)}} && \text{(translational invariance)} \\
&\leq (c + c')\sqrt{\frac{l_{n_t(a)}}{n_t(a)}} && (a \in \mathcal{A}_{t+1})
\end{aligned}$$

This holds for a given action a in all rounds up to the round before it was eliminated and $n_t(a)$ increases by at most 1 afterwards. Therefore, we have

$$\operatorname{op}(\Delta(a; a^\star)) \leq (c + c')\sqrt{\frac{l_{T'}}{n_{T'}(a)} - 1} \leq (c + c')\sqrt{\frac{2l_{T'}}{n_{T'}(a)}}$$

where the second equality holds because every action needs to be played at least 2 times before being eliminated. As a result, it holds that

$$n_{T'}(a) \operatorname{op} \Delta(a; a^\star) \leq (c + c')\sqrt{2l_{T'} n_{T'}(a)}.$$

Hence, we can control the op-regret up to time T' directly as

$$\begin{aligned}
\operatorname{op}\left(\operatorname{Reg}(T')\right) &\leq \sum_{a \in \mathcal{A}} n_{T'}(a) \operatorname{op}(\Delta(a)) \\
&= \sum_{a \in \mathcal{A}^\varepsilon} \left(2n_{T'}(a) \operatorname{op}(\Delta(a)) - n_{T'}(a) \operatorname{op}(\Delta(a))\right) + \sum_{a \notin \mathcal{A}^\varepsilon} n_{T'}(a) \operatorname{op}(\Delta(a)) \\
&\leq \sum_{a \in \mathcal{A}^\varepsilon} \left[2(c' + c)\sqrt{2l_{T'} n_{T'}(a)} - n_{T'}(a) \operatorname{op}(\Delta(a))\right] \\
&\quad + \sum_{a \in \mathcal{A}^\varepsilon} 2n_{T'}(a) \operatorname{op}(\Delta(a^\star)) + \sum_{a \notin \mathcal{A}^\varepsilon} n_{T'}(a) \operatorname{op}(\Delta(a)) && \text{(by inequality above)} \\
&\leq \sum_{a \in \mathcal{A}^\varepsilon} \left[2(c' + c)\sqrt{2l_{T'} n_{T'}(a)} - n_{T'}(a) \operatorname{op}(\Delta(a))\right] + 2T'\varepsilon \\
& && \text{(since } \operatorname{op}(\Delta(a^\star)) \leq \max_{a \notin \mathcal{A}^\varepsilon} \operatorname{op}(\Delta(a)) \leq \varepsilon\text{)} \\
&\leq \sum_{a \in \mathcal{A}^\varepsilon} \max_{x \in \mathbb{R}} \left[x - x^2 \frac{\operatorname{op}(\Delta(a))}{8(c' + c)^2 l_{T'}}\right] + 2T'\varepsilon \\
& && \text{(substitute } x = 2(c' + c)\sqrt{2l_{T'} n_{T'}(a)} \text{ and maximize over } x\text{)} \\
&= \sum_{a \in \mathcal{A}^\varepsilon} \frac{4(c' + c)^2 l_{T'}}{\operatorname{op}(\Delta(a))} + 2T'\varepsilon
\end{aligned}$$

To bound the regret after $a^\star$ gets eliminated (if it does), we will argue that only actions that are not much worse than $a^\star$ could have survived up to this point. To this end, note that at the time of elimination $T^\star$ we have

$$\begin{aligned}
c'\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} &< \operatorname{op}\left(\widehat{\Delta}_{T^\star}(a^\star, \mathcal{A}_{T^\star})\right) \\
&\leq \operatorname{op}\left(\Delta(a^\star, \mathcal{A}_{T^\star}) + c\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}}\right) && \text{(Lemma 1)} \\
&= \operatorname{op}\left(\Delta(a^\star, \mathcal{A}_{T^\star})\right) + c\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} && \text{(translational invariance)} \\
&\leq \operatorname{op}\left(\Delta(a^\star)\right) + c\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} && \text{(monotonicity)}
\end{aligned}$$

Rearranging terms gives

$$\operatorname{op}\left(\Delta(a^\star)\right) > (c' - c)\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}}.$$

Let now $a \in A_{T^\star+1}$ be any action that survived $a^\star$. This action must have passed the elimination test in round $T^\star$ and thus satisfy

$$\begin{aligned}
c'\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} &\geq \operatorname{op}\left(\widehat{\Delta}_{T^\star}(a, \mathcal{A}_{T^\star})\right) \\
&\geq \operatorname{op}\left(\Delta(a, \mathcal{A}_{T^\star}) - c\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}}\right) && \text{(Lemma 1)} \\
&= \operatorname{op}\left(\Delta(a, \mathcal{A}_{T^\star})\right) - c\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} && \text{(translational invariance)} \\
&\geq \operatorname{op}\left(\Delta_{T^\star}(a; a^\star)\right) + c\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} && \text{(monotonicity)}
\end{aligned}$$

Rearranging terms gives

$$\operatorname{op}\left(\Delta(a; a^\star)\right) \leq (c' + c)\sqrt{\frac{l_{n_{T^\star}(a)}}{n_{T^\star}(a)}} \leq \frac{c' + c}{c' - c} \operatorname{op}\left(\Delta(a^\star)\right)$$

which allows us to bound the gap of a as

$$\operatorname{op}(\Delta(a)) \leq \operatorname{op}(\Delta(a; a^\star) + \operatorname{op}(\Delta(a^\star)) \leq \left[1 + \frac{c' + c}{c' - c}\right] \operatorname{op}\left(\Delta(a^\star)\right).$$

Since this holds for any action $a \in \mathcal{A}_{T^\star+1}$, we can bound the regret incurred in rounds $T' + 1, \ldots, T$ as

$$(T - T^\star)\left[1 + \frac{c' + c}{c' - c}\right] \operatorname{op}\left(\Delta(a^\star)\right) \leq (T - T')\left[1 + \frac{c' + c}{c' - c}\right] \varepsilon$$

Choosing $c' = 2c$ and putting everything together gives a regret bound of

$$\operatorname{op}(\operatorname{Reg}(T)) \leq 36c \sum_{a \in \mathcal{A}^\varepsilon} \frac{l_T}{\operatorname{op}(\Delta(a))} + 4T\varepsilon$$

□

A.2. Proof of Theorem 2

Proof of Theorem 2. First note that $a^\star$ is never eliminated. This is true because for any active action a

$$\begin{aligned}
\operatorname{op}(\widehat{\Delta}_t(a^\star; a)) &\leq \operatorname{op}\left(\Delta(a^\star; a) + c\sqrt{\frac{l_{n_t(a^\star)}}{n_t(a^\star)}}\right) && \text{(Lemma 1)} \\
&\leq \operatorname{op}\left(\Delta(a^\star; a)\right) + c\sqrt{\frac{l_{n_t(a^\star)}}{n_t(a^\star)}} && \text{(translational invariance)} \\
&\leq \operatorname{op}(\Delta(a^\star)) + c\sqrt{\frac{l_{n_t(a^\star)}}{n_t(a^\star)}} && \text{(monotonicity of op)} \\
&\leq c\sqrt{\frac{l_{n_t(a^\star)}}{n_t(a^\star)}} && (a^\star \text{ optimal}) \\
&\leq c'\sqrt{\frac{l_{n_t(a^\star)}}{n_t(a^\star)}} && (c' = 2c > c)
\end{aligned}$$

where $l_t = \ln(mA\ln(t)/\delta)$. As a result, have for all actions a from the proof of Theorem 1 that

$$\begin{aligned}
n_T(a)\operatorname{op}(\Delta(a)) &= n_T(a)\operatorname{op}(\Delta(a; a^\star)) \\
&\leq (c' + c)\sqrt{n_T(a) 2 l_T} = 3c\sqrt{2 n_T(a) l_T}.
\end{aligned}$$

Rerranging terms and resolving the quadratic form yields

$$n_T(a) \leq 18\frac{c^2 l_T}{\operatorname{op}(\Delta(a))^2}$$

Hence, the regret vector can be readily bounded as

$$\begin{aligned}
\operatorname{Reg}(T) &= \sum_{a\in\mathcal{A}^\varepsilon} n_T(a)\Delta(a) + \sum_{a\notin\mathcal{A}^\varepsilon} n_T(a)\Delta(a) \\
&\leq \sum_{a\in\mathcal{A}^\varepsilon} n_T(a)\Delta(a) + \sum_{a\notin\mathcal{A}^\varepsilon} n_T(a)\operatorname{op}(\Delta(a)) && (\operatorname{op} = \max) \\
&\leq \sum_{a\in\mathcal{A}^\varepsilon} n_T(a)\Delta(a) + T\varepsilon && (\mathcal{A}^\varepsilon \text{ definition}) \\
&\leq 18c^2 l_T \sum_{a\in\mathcal{A}^\varepsilon} \frac{\Delta(a)}{\operatorname{op}(\Delta(a))^2} + T\varepsilon
\end{aligned}$$

□

Lemma 1 (Concentration)**.** *Let $\widehat{\Delta}_t(a; a') = \widehat{r}_t(a') - \widehat{r}_t(a)$ be the empirical gap between actions a and a'. Then with probability at least $1-\delta$ for all $t \in \mathbb{N}$ and $a \in \mathcal{A}_{t-1}$*

$$|\widehat{\Delta}_t(a; a') - \Delta(a; a')| \leq c\sqrt{\frac{\ln(mA\ln(n_t(a))/\delta}{n_t(a)}}$$

where $c > 0$ is an absolute constant.

Proof. This follows from a time-uniform Hoeffding bound, a union bound and the observation that the number of observations $n_t(a)$ and $n_t(a')$ between two active actions can deviate at most by 1. □

B. Proofs for Markov Decision Processes

B.1. Proof of Theorem 3

Proof of Theorem 3. The regret under each reward function can then be bounded as

$$
\begin{aligned}
&\operatorname{op}(\operatorname{Reg}(T)) \\
&= \operatorname{op}\left(\sum_{k=1}^{T}(V^\star(x_1) - V^{\pi_k}(x_1))\right) \\
&= \operatorname{op}\left(\sum_{k=1}^{T}\sum_{h=1}^{H}\mathbb{E}^{\pi_k}\left[V^\star(x_h) - Q^\star(x_h,a_h)\right]\right) = \operatorname{op}\left(\sum_{k=1}^{T}\sum_{h=1}^{H}\mathbb{E}^{\pi_k}\left[\Delta(x_h,a_h)\right]\right) \\
&\leq \sum_{k=1}^{T}\sum_{h=1}^{H}\mathbb{E}^{\pi_k}\left[\operatorname{op}\left(\Delta(x_h,a_h)\right)\right] && \text{(sub-additivity of op)} \\
&= \sum_{k=1}^{T}\sum_{h=1}^{H}\mathbb{E}^{\pi_k}\left[\mathbb{1}\{x_h \in \mathcal{G}_k\}\operatorname{op}\left(\Delta(x_h,a_h)\right)\right] + \sum_{k=1}^{T}\sum_{h=1}^{H}\mathbb{E}^{\pi_k}\left[\mathbb{1}\{x_h \notin \mathcal{G}_k\}\operatorname{op}\left(\Delta(x_h,a_h)\right)\right].
\end{aligned}
$$

By Lemma 3, we know that $\pi_i^\star$ is never eliminated for all $i \in [m]$ in event $\mathcal{E}$. Hence, all $\{\pi_i^\star\}_{i\in[m]]}$ need to agree on states $x \in \mathcal{G}_k$. Further, since $\pi_i^\star$ is optimal for reward i, the remaining action needs to be optimal under all reward functions and incur no regret. Hence, the regret is bounded in event $\mathcal{E}$ as

$$
\operatorname{op}(\operatorname{Reg}(T)) \leq \sum_{k=1}^{T}\sum_{h=1}^{H}\mathbb{E}^{\pi_k}\left[\mathbb{1}\{x_h \notin \mathcal{G}_k\}\operatorname{op}\left(\Delta(x_h,a_h)\right)\right]
$$

Uncertainty potential. We now upper-bound the max-gap by the maximum uncertainty for state-action pairs (x,a) that is not yet eliminated, i.e., $a \in \mathcal{A}_k(x)$

$$
\begin{aligned}
\operatorname{op}\left(\Delta(x,a)\right) &= \operatorname{op}\left(V^\star(x) - Q^\star(x,a)\right) \\
&\leq \operatorname{op}\left(\overline{V}_{k-1}(x) - \underline{Q}_{k-1}(x,a)\right) && \text{(Lemma 5 and monotonicity of op)} \\
&= \operatorname{op}\left(\overline{Q}_{k-1}(x,\hat{a}) - \underline{Q}_{k-1}(x,a)\right) && \text{(where } \hat{a} = \operatorname{argmax}_{a'}\overline{Q}_{k-1}(x,a')\text{)} \\
&= \operatorname{op}\big(\overline{Q}_{k-1}(x,\hat{a}) - \underline{Q}_{k-1}(x,\hat{a}) + \overline{Q}_{k-1}(x,a) - \underline{Q}_{k-1}(x,a) \\
&\qquad + \underline{Q}_{k-1}(x,\hat{a}) - \overline{Q}_{k-1}(x,a)\big) \\
&\leq \operatorname{op}\left(\overline{Q}_{k-1}(x,\hat{a}) - \underline{Q}_{k-1}(x,\hat{a})\right) + \operatorname{op}\left(\overline{Q}_{k-1}(x,a) - \underline{Q}_{k-1}(x,a)\right) \\
&\quad + \operatorname{op}\left(\underline{Q}_{k-1}(x,\hat{a}) - \overline{Q}_{k-1}(x,a)\right) && \text{(sub-additivity)} \\
&\leq 2\max_{a'\in A_k(x)}\operatorname{op}\left(\overline{Q}_{k-1}(x,a') - \underline{Q}_{k-1}(x,a')\right) + \operatorname{op}\left(\underline{Q}_{k-1}(x,\hat{a}) - \overline{Q}_{k-1}(x,a)\right) \\
&= 2\operatorname{op}\left(\overline{Q}_{k-1}(x,\pi_k(x)) - \underline{Q}_{k-1}(x,\pi_k(x))\right) + \operatorname{op}\left(\underline{Q}_{k-1}(x,\hat{a}) - \overline{Q}_{k-1}(x,a)\right) && \text{(definition of } \pi_k\text{)}
\end{aligned}
$$

The second term can further be bounded using the elimination condition as

$$
\begin{aligned}
\operatorname{op}\left(\underline{Q}_{k-1}(x,\hat{a}) - \overline{Q}_{k-1}(x,a)\right) &\leq \operatorname{op}\left(\underline{V}_{k-1}(x) - \overline{Q}_{k-1}(x,a)\right) && \text{(monotonicity of op)} \\
&= \operatorname{op}\left(\underline{V}_{k-1}(x) - \overline{Q}_{k-1}(x,a) - \varepsilon + \varepsilon\right) \\
&\leq \operatorname{op}\left(\underline{V}_{k-1}(x) - \overline{Q}_{k-1}(x,a) - \varepsilon\right) + \operatorname{op}\left(\varepsilon\right) && \text{(sub-additivity of op)} \\
&\leq \operatorname{op}\left(\varepsilon\right) && \text{(elimination condition)}
\end{aligned}
$$

We define the potential function

$$W_k(x,a) = \mathbb{1}\{x \notin \mathcal{G}_k\}\left[\mathtt{op}\left(\overline{Q}_{k-1}(x,a) - \underline{Q}_{k-1}(x,a)\right) + \frac{\mathtt{op}(\varepsilon)}{2}\right]$$

which we have shown satisfies for all active actions $a' \in \mathcal{A}_k(x)$ and $a = \pi_k(x)$

$$\frac{\mathbb{1}\{x \notin \mathcal{G}_k\}}{2}\,\mathtt{op}(\Delta(x,a')) \le W_k(x,a).$$

Since $\overline{Q}_k \ge \underline{Q}_k$ uniformly, it also holds that

$$\frac{\mathbb{1}\{x \notin \mathcal{G}_k\}}{2}\varepsilon \le W_k(x,a).$$

For each state x, let $(\mathtt{op}(\Delta(x,a^i)))_{i\in[A]}$ be the sorted max-gaps, that is $\mathtt{op}(\Delta(x,a^i)) \le \mathtt{op}(\Delta(x,a^j))$ for $i<j$. We then denote the second-smallest max-gaps as $\Delta_2^{\mathtt{op}}(x) = \mathtt{op}(\Delta(x,a^2))$. This may be zero if there a multiple actions that are optimal under all reward functions. We then also have

$$\frac{\mathbb{1}\{x \notin \mathcal{G}_k\}}{2}\Delta_2^{\mathtt{op}}(x) \le W_k(x,a).$$

Putting the pieces together, the expected regret, conditioned on the event $\mathcal{E}$ where all the concentration argument hold, can be bounded as

$$\mathbb{E}[\mathtt{op}(\mathrm{Reg}(T))|\mathcal{E}] \le 2\mathbb{E}\left[\sum_{k=1}^{T}\sum_{h=1}^{H} W_k(x_h,a_h) \mid \mathcal{E}\right]. \tag{2}$$

Recursive form of potential. We can recursively bound the uncertainty $\overline{Q}^i_{k-1}(x,a) - \underline{Q}^i_{k-1}(x,a)$ for all $i \in [m]$ as

$$\begin{aligned}
&\overline{Q}^i_{k-1}(x,a) - \underline{Q}^i_{k-1}(x,a) \\
&= \alpha^0_{n_{k-1}}H + \sum_{t=1}^{n_{k-1}} \alpha^t_{n_{k-1}}\left(\overline{V}^i_{k[t]-1}(x'_{k[t]}) - \underline{V}^i_{k[t]-1}(x'_{k[t]}) + 2b_t(x,a)\right) && \text{(update rule)} \\
&\le \alpha^0_{n_{k-1}}H + 2b_{n_{k-1}}(x,a) + \sum_{t=1}^{n_{k-1}} \alpha^t_{n_{k-1}}\left(\overline{V}^i_{k[t]-1}(x'_{k[t]}) - \underline{V}^i_{k[t]-1}(x'_{k[t]})\right) \\
&\le \alpha^0_{n_{k-1}}H + 2b_{n_{k-1}}(x,a) + \sum_{t=1}^{n_{k-1}} \alpha^t_{n_{k-1}}\left(\overline{Q}^i_{k[t]-1}(x'_{k[t]},a'_{k[t]}) - \underline{Q}^i_{k[t]-1}(x'_{k[t]},a'_{k[t]})\right) \\
&\le \alpha^0_{n_{k-1}}H + 2b_{n_{k-1}}(x,a) + \sum_{t=1}^{n_{k-1}} \alpha^t_{n_{k-1}}\left(\overline{Q}^i_{k[t]-1}(x'_{k[t]},a'_{k[t]}) - \underline{Q}^i_{k[t]-1}(x'_{k[t]},a'_{k[t]})\right)\mathbb{1}\left\{x'_{k[t]} \notin \mathcal{G}_k\right\} \\
& && (x'_{k[t]} \text{ never in } \mathcal{G}_k)
\end{aligned}$$

where n_{k-1} is the number of visits to (x,a) up to episode $k-1$, $k[t]$ is the episode index of the t-th visit to (x,a) and x'_t is the t-th successor state observed for (x,a). Applying $\mathtt{op}$ on both sides yields

$$W_k(x,a) \le \alpha^0_{n_{k-1}}H + 2b_{n_{k-1}}(x,a) + \frac{\mathtt{op}(\varepsilon)}{2} + \sum_{t=1}^{n_{k-1}} \alpha^t_{n_{k-1}} W_{k[t]}(x'_{k[t]},a'_{k[t]}).$$

Clipping. Since W_k is only non-zero for (x,a) pairs with $x \notin \mathcal{G}_k$ and for those (x,a) we have $\frac{\mathtt{op}(\Delta(x,a))\vee\mathtt{op}(\varepsilon)\vee\Delta_2^{\mathtt{op}}(x)}{2} \le W_k(x,a)$, we can apply the following lemma with $c = \frac{\mathtt{op}(\Delta(x,a))\vee\mathtt{op}(\varepsilon)\vee\Delta_2^{\mathtt{op}}(x)}{2}$ and $x = 1/H$.

Lemma 2 (Claim A.8 of Xu et al. (2021))**.** *For any three positive numbers $c \le a+b$, the following holds for all $x \in (0,1)$*

$$a + b \le \mathrm{clip}\left[a \mid \frac{xc}{2}\right] + (1+x)b$$

This gives

$$W_k(x,a) \le \left(1+\frac{1}{H}\right)\left(\operatorname{op}(\varepsilon)+\alpha^0_{n_{k-1}}H+\sum_{t=1}^{n_{k-1}}\alpha^t_{n_{k-1}}W_{k[t]}(x'_{k[t]},a'_{k[t]})\right) + \operatorname{clip}\left[2b_{n_{k-1}}(x,a) \mid \frac{\operatorname{op}(\varepsilon)\vee\operatorname{op}(\Delta(x,a))\vee\Delta_2^{\operatorname{op}}(x)}{4H}\right]$$

We denote $\phi_n(x,a)=\left(1+\frac{1}{H}\right)\left(\operatorname{op}(\varepsilon)+\alpha^0_n\right)+\operatorname{clip}\left[2b_n(x,a) \mid \frac{\operatorname{op}(\varepsilon)\vee\operatorname{op}(\Delta(x,a))\vee\Delta_2^{\operatorname{op}}(x)}{4H}\right]$ and show for any h

$$\sum_{k=1}^{T}W_k(x_{k,h},a_{k,h}) \le \sum_{k'=1}^{T}\sum_{h'=h}^{H}w(h,h')\phi_{n_{k'-1}}(x_{k',h'},a_{k',h'})$$

where $w(h,h')=(1+1/H)^{2(h'-h)}$ using the argument in Proposition 4.6 of Xu et al. (2021). This yields that

$$\begin{aligned}
&\sum_{k=1}^{T}\sum_{h=1}^{H}W_k(x_{k,h},a_{k,h})\\
&\le e^3SAH^2+e^3TH\operatorname{op}(\varepsilon)\\
&\quad+2e^2H\sum_{k=1}^{T}\sum_{h=1}^{H}\operatorname{clip}\left[b_{n_{k-1}}(x_{k,h},a_{k,h}) \mid \frac{\operatorname{op}(\varepsilon)\vee\operatorname{op}(\Delta(x,a))\vee\Delta_2(x)}{4H}\right]\\
&\lesssim SAH^2+TH\operatorname{op}(\varepsilon)+\sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}}\frac{H^5}{\operatorname{op}(\varepsilon)\vee\operatorname{op}(\Delta(x,a))\vee\Delta_2^{\operatorname{op}}(x)}\log\left(\frac{mSAHT}{\delta}\right)
\end{aligned}$$

Plugging the bound on the total sum of potential functions back in Equation 2 gives the desired result

$$\begin{aligned}
\mathbb{E}[\operatorname{op}(\operatorname{Reg}(T))|\mathcal{E}] &\le O\left(\sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}}\frac{H^5}{\operatorname{op}(\varepsilon)\vee\operatorname{op}(\Delta(x,a))\vee\Delta_2^{\operatorname{op}}(x)}\log\left(\frac{mSAHT}{\delta}\right)+SAH^2+TH\operatorname{op}(\varepsilon)\right)\\
&= O\left(\sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}}\frac{H^5}{\operatorname{op}(\varepsilon)\vee\operatorname{op}(\Delta(x,a))\vee\Delta_2^{\operatorname{op}}(x)}\log\left(\frac{mSAHT}{\delta}\right)+TH\operatorname{op}(\varepsilon)\right)
\end{aligned}$$

□

B.2. Supporting lemmas

Following the single-reward analysis of Xu et al. (2021), we decompose the optimal Q-function for a reward i as

$$Q^{\star,i}(x,a)=R^i_k(x,a)+J^i_k(x,a).$$

To define the two terms, we use the random variables x' and h' which denote the first unknown state encountered when following $\pi^\star_i$ from (x,a) on and the time step when x' is encountered, respectively. Then we have

$$\begin{aligned}
R^i_k(x,a)&=\mathbb{E}^{\pi^\star_i}\left[\sum_{\ell=h}^{h'-1}\bar{r}^i(x_\ell,a_\ell) \mid x_h=x,a_h=a\right]\\
J^i_k(x,a)&=\mathbb{E}^{\pi^\star_i}\left[V^{\star,i}(x') \mid x_h=x,a_h=a\right]
\end{aligned}$$

where $x\in\mathcal{X}_h$. The empirical versions of these quantities are

$$\widehat{R}^i_k(x,a)=\sum_{\ell=h}^{h'-1}\bar{r}^i(x_{k,\ell},a_{k,\ell})$$

$$\widehat{J}_k^i(x,a) = V^{\star,i}(x_{k,h'})$$

where again h' is the first time in episode k that an unknown state is encountered since from h on.

Lemma 3 (Concentration)**.** *With probability at least $1-\delta$ the following conditions hold for all episodes $k \in [K]$, $i \in [m]$, $x \in \mathcal{X} \setminus \mathcal{G}_k$, $a \in \mathcal{A}_k(x)$*

$$\left|\sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\widehat{R}_{k[t]}^i(x,a) - R_{k[t]}^i(x,a)\right)\right| \le \frac{1}{2} b_{n_k}(x,a) \tag{3}$$

$$\left|\sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\widehat{J}_{k[t]}^i(x,a) - J_{k[t]}^i(x,a)\right)\right| \le \frac{1}{2} b_{n_k}(x,a) \tag{4}$$

where $n_k = n_k(x,a)$ is the number of visits to (x,a) before episode k and $k[t] = k[t](x,a)$ is the episode index of the t-th visit to (x,a). The event where the inequalities above hold is denoted by $\mathcal{E}$.

Proof. This follows from a simple martingale concentration argument in combination with union bounds. See also Lemma 4.1 of Xu et al. (2021). □

The following quantities are a standard tool in the analysis of model-free RL algorithms and also used in our analysis:

$$\alpha_n^t = \alpha_t \prod_{\ell=t+1}^{n} (1-\alpha_\ell) \qquad\qquad \alpha_n^0 = \prod_{\ell=1}^{n} (1-\alpha_\ell)$$

for $0 < t \le n$. These quantites satisfy the following properties

Lemma 4 (Lemma 4.1 by Jin et al. (2018))**.** *The following properties hold for all $n \ge 1$, $t \ge 1$*

1. $\frac{1}{\sqrt{n}} \le \sum_{t=1}^n \frac{\alpha_n^t}{\sqrt{t}} \le \frac{2}{\sqrt{n}}$
2. $\alpha_0^0 = 1$ *and* $\sum_{t=1}^0 \alpha_0^t = 0$
3. $\alpha_n^0 = 0$ *and* $\sum_{t=1}^n \alpha_n^t = 1$
4. $\sum_{n=t}^\infty \alpha_n^t = 1 + \frac{1}{H}$
5. $\sum_{t=1}^n (\alpha_n^t)^2 \le \frac{2H}{n}$

Lemma 5 (Valid Confidence Bounds)**.** *In event $\mathcal{E}$, we have for all $k \in [K], i \in [m], x \in \mathcal{X}$ that the following statements hold:*

- $\pi_i^\star(x) \in \mathcal{A}_k(x)$
- *for all $a \in \mathcal{A}_k(x)$*

$$\underline{V}_{k,i}(x) \le V_i^\star(x) \le \overline{V}_{k,i}(x)$$
$$\underline{Q}_{k,i}(x,a) \le Q_i^\star(x,a) \le \overline{Q}_{k,i}(x,a)$$

Proof. We prove this inductively. First assume both claims holds for episodes up to including k. We will now show that the first claim holds for $k+1$. Let $x \in \mathcal{X}_h$ for arbitrary h and assume $\pi_i^\star(x) \notin \mathcal{A}_{k+1}(x)$. Since $\pi_i^\star(x) \in \mathcal{A}_k(x)$, we have

$$\begin{aligned}
\mathtt{op}\left(\underline{V}_k(x) - \overline{Q}_k(x, \pi_i^\star(x)) - \varepsilon\right) &\le \mathtt{op}\left(V^\star(x) - Q^\star(x,\pi_i^\star(x)) - \varepsilon\right) && \text{(induction hypothesis and monotonicity)} \\
&= \mathtt{op}\left(\Delta(x,\pi_i^\star(x)) - \varepsilon\right) && \text{(gap definition)} \\
&\le \mathtt{op}\left(0\right) && \text{(Assumption 2)}
\end{aligned}$$

$$= 0 \qquad \text{(transitional invariance)}$$

Hence, $\pi_i^\star(x) \in \mathcal{A}_{k+1}(x)$ since it satisfies the condition in the active set update of Algorithm 2.

Note that the second claim holds trivially for all $x \in \mathcal{X}_{H+1}$ by definition. Assume now that the first claim holds for all episodes up to k and the second claim holds for all episodes up to $k-1$ and for all time steps $h+1, \dots, H+1$ in episode k. We will show that it also holds for time step h in episode k. Consider first $\overline{Q}_k$. Since it only changes for $(x_{k,h}, a_{k,h})$, the condition holds by the induction hypothesis for all other (x, a) with $x \in \mathcal{X}_h$ and $a \in \mathcal{A}_k(x)$. By the update rule in the algorithm, we can write $\overline{Q}_k^i(x_{k,h}, a_{k,h})$ as

$$\overline{Q}_k^i(x_{k,h}, a_{k,h}) = \min \Bigg\{ H - h + 1, \tag{5}$$

$$\mathbb{1}\{n_k = 0\}(H-h+1) + \sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\widehat{R}_{k[t]}^i(x_{k[t],h}, a_{k[t],h}) + \overline{V}_{k[t]-1}^i(x_{k[t],h'}) + b_t \right) \Bigg\}, \tag{6}$$

where $n_k = n_k(x_{k,h}, a_{k,h}$ is the number of visits of the state-action pair, $k[t]$ is the episode of the t-th visit to $(x_{k,h}, a_{k,h}$ and $h'[t]$ is the time step of the next unknown state encountered after that t-th visit. By the properties of the learning rate, we can also write $Q^{\star,i}$ in a similar form:

$$Q^{\star,i}(x_{k,h}, a_{k,h}) = \mathbb{1}\{n_k = 0\} Q^{\star,i}(x_{k,h}, a_{k,h}) + \sum_{t=1}^{n_k} \alpha_{n_k}^t Q^{\star,i}(x_{k[t],h}, a_{k[t],h})$$

Since $Q^{\star,i}(x_{k,h}, a_{k,h}) \le H + 1 - h$ by the normalization of the rewards, the desired condition holds whenever the first term in the min is tight in Equation 6 or $n_k = 0$. For the remaining case, we have

$$\begin{aligned}
\overline{Q}_k^i(x_{k,h}, a_{k,h}) - Q^{\star,i}(x_{k,h}, a_{k,h}) &\ge \sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\widehat{R}_{k[t]}^i(x_{k[t],h}, a_{k[t],h}) + \overline{V}_{k[t]-1}^i(x_{k[t],h'[t]} + b_t - Q^{\star,i}(x_{k[t],h}, a_{k[t],h}) \right) \\
&= \sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\widehat{R}_{k[t]}^i(x_{k[t],h}, a_{k[t],h}) - R_{k[t]}^i(x_{k[t],h}, a_{k[t],h}) \right) \\
&\quad + \sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\overline{V}_{k[t]-1}^i(x_{k[t],h'} - J_{k[t]}^i(x_{k[t],h}, a_{k[t],h}) \right) + \sum_{t=1}^{n_k} \alpha_{n_k}^t b_t \quad \text{(Equation 3)} \\
&\ge \sum_{t=1}^{n_k} \alpha_{n_k}^t \left(\overline{V}_{k[t]-1}^i(x_{k[t],h'[t]} - V^{\star,i}(x_{k[t],h'[t]}) \right) - b_{n_k} + \sum_{t=1}^{n_k} \alpha_{n_k}^t b_t \quad \text{(Lemma 3)} \\
&\ge -b_{n_k} + \sum_{t=1}^{n_k} \alpha_{n_k}^t b_t \quad \text{(induction hypothesis)} \\
&\ge 0. \quad \text{(Lemma 4)}
\end{aligned}$$

Analogously, we can show that $\underline{Q}_k^i(x_{k,h}, a_{k,h}) \le Q^{\star,i}(x_{k,h}, a_{k,h})$. Finally, since $\underline{Q}_k^i(x, a) \le Q^{\star,i}(x, a) \le \overline{Q}_k^i(x, a)$ for all $x \in \mathcal{X}_h$ and $a \in \mathcal{A}_k(x)$, we have

$$\begin{aligned}
\overline{V}_k^i(x) - V^{\star,i}(x) &= \max_{a \in \mathcal{A}_k(x)} \overline{Q}_k^i(x, a) - \max_{a' \in \mathcal{A}} Q^{\star,i}(x, a) \\
&\ge \max_{a \in \mathcal{A}_k(x)} Q^{\star,i}(x, a) - \max_{a' \in \mathcal{A}} Q^{\star,i}(x, a) \\
&\ge 0. \quad \text{(first claim holds: } \pi_i^\star \in \mathcal{A}_k(x))
\end{aligned}$$

Analogously, the condition for $\underline{V}$ can be shown. Hence, the second claim holds also for time step h in episode k which completes the induction argument. □