Unpacking Reward Shaping: Understanding the Benefits of Reward Engineering on Sample Complexity

Anonymous Author(s)
Affiliation
Address
email

Abstract

The success of reinforcement learning in a variety of challenging sequential decision-making problems has been much discussed, but often ignored in this discussion is the consideration of how the choice of reward function affects the behavior of these algorithms. Most practical RL algorithms require copious amounts of reward engineering in order to successfully solve challenging tasks. The idea of this type of "reward-shaping" has been often discussed in the literature and is used in practical instantiations, but there is relatively little formal characterization of how the choice of reward shaping can yield benefits in sample complexity for RL problems. In this work, we build on the framework of novelty-based exploration to provide a simple scheme for incorporating shaped rewards into RL along with an analysis tool to show that particular choices of reward shaping provably improve sample efficiency. We characterize the class of problems where these gains are expected to be significant and show how this can be connected to practical algorithms in the literature. We show that these results hold in practice in experimental evaluations as well, providing an insight into the mechanisms through which reward shaping can significantly improve the complexity of reinforcement learning while retaining asymptotic performance.

1 Introduction

Reinforcement learning (RL) in its most general form presents a very difficult optimization problem: when there are no constraints on the reward function or dynamics, a learning algorithm may need to exhaustively explore the entire state space to discover high-reward regions. Naïve algorithms that rely entirely on random exploration are known to be exponentially expensive [19, 2], and much of the theoretical work on efficient RL algorithms has focused smarter exploration strategies that aim to more efficiently cover state space, typically in time polynomial in the state space cardinality [5, 7, 17, 46, 20, 25, 43]. Much of this work is based on upper confidence bound (UCB) principles and prescribes some kind of exploration bonus to prioritize exploration of rarely visited regions. Analogous strategies have also been employed in a number of practical RL algorithms [32, 31, 9, 11, 15, 27, 34, 33, 28]. However, surprisingly, much of the empirical work on reinforcement learning does not make use of explicit exploration bonuses or other dedicated exploration strategies, despite numerous theoretical results showing them to be essential to attain polynomial sample complexity. Instead, practitioners often incorporate prior knowledge of each task into designing or shaping the reward function [24, 29, 30, 35, 4, 10, 41, 45], preferring this heuristic approach over the more principled exploration strategies. At this point, one may wonder why is reward shaping often practically preferable to dedicated exploration?

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

A likely answer to this question lies in the fact that even the best general-purpose exploration strategies still require visiting every state in the MDP at least once in the worst case. This is of course unavoidable without further assumptions. However, in practice, sample complexity that is polynomial in the size of the entire state space might *still* not be practical, and hence prior knowledge in the form of reward shaping is required to render such tasks tractable for RL. Surprisingly, despite the widespread popularity of reward shaping in RL applications, the analysis of reward shaping has remained limited to proving policy invariance [24] or largely empirical observations. In this work, we take a step towards studying the effect of reward shaping on the efficiency of RL algorithms, by asking the following question:

Can we theoretically justify the sample complexity benefits that reward shaping from prior domain knowledge can provide reinforcement learning?

In this work, we aim to provide a set of tools that formally analyze how reward shaping can improve the complexity of tabula-rasa RL and better direct. exploration. To perform this analysis, we first propose a simple RL algorithm "UCBVI-Shaped" that incorporates shaped rewards into optimism based exploration. We then provide a regret analysis framework that studies how this introduction of shaped rewards can (in certain cases) provide much more directed optimism than uninformed exploration, while maintaining asymptotic performance.

In our analysis, we specifically consider problems where shaping is provided through a term $\widetilde{V}$, a (potentially suboptimal) approximation of the optimal value function $V^\star$. In particular, we assume that the shaping $\widetilde{V}$ is a multiplicatively bounded approximation of the optimal value function $V^\star$, i.e $\widetilde{V}(s) \leq V^\star(s) \leq \beta\widetilde{V}(s) \forall s$ for a finite multiplicative factor β. This type of shaped reward function $\widetilde{V}$ can be incorporated into a standard RL algorithm like UCBVI [7] through two channels - (1) bonus scaling: simply reweighting a standard, decaying count-based bonus $\frac{1}{\sqrt{N_h(s,a)}}$ by the per-state reward shaping and (2) value projection: adaptively projecting learned value functions into ranges of value functions derived from the reward shaping.

We show that this relatively simple algorithmic instantiation provides significant complexity benefits with shaping. Intuitively, the key pieces in our complexity analysis of UCBVI-Shaped are two-fold: 1) The multiplicative sandwich condition (via β) between $\widetilde{V}$ and $V^\star$ allows for the bonus scaling to depend on $\beta\widetilde{V}$ instead of a coarse approximation of $V^\star$ such as H. This allows for a reduction of complexity from a horizon H dependent term to one scaling with $\beta\widetilde{V} \leq \beta V^\star$ by allowing for a faster decay of the exploration bonuses while still providing enough optimism; 2) The projection of the value function prevents "over optimism" by hastening the convergence of the empirical $\widehat{Q}$ functions during value iteration thus allowing for faster detection of sub-optimal actions. This results in the ability to prune out large parts of the state space, as we also validate empirically.

2 Related Work

Regret Analysis in Finite Horizon Episodic Tabular MDPs Recent research on regret analysis has studied both model-based and model-free RL methods. Model-based methods [16, 3, 7, 14, 13, 26] first learn a model from previous experience and use the learned model for future decision making. In contrast, model-free methods [17, 8, 46, 37, 23, 20] aim to learning the value function without the model estimation stage and use the learned value function for decision making. Our analysis lies in the model-based framework and is similar to the setting of [7] but with additional assumptions on knowing $\widetilde{V}$ as a multiplicatively bounded approximation of $V^\star$. The main difference between our results and the aforementioned works is that, the truncated interval assumption can reduce the state dependency S to the some smaller "effective state space" and reduce horizon dependency.

Regret Analysis with Linear Function Approximation A recent line of literature has investigated the linear function approximation setting by assuming the transition kernels or the value functions can be represented by d-dimensional linear features [18, 44, 6, 47, 12, 39, 38, 36] or even general function classes [40]. With the aforementioned assumptions, the regret analysis will only depend on the ambient dimension d (or other intrinsic complexity measure), instead of S,A in the tabular setting [36, 39, 12, 47, 6, 43, 18], which could greatly decrease the complexity of learning. Instead of

posing structural assumptions on the function class representing the MDP's values or dynamics, we ask the question of whether having approximate knowledge of $V^\star$ can improve the speed of learning.

Practical Reward Design and Reward Shaping Ng et al. [24] proposed a potential-based shaping function F that ensures policy invariance under transformation. However, unbiased potential based reward shaping is rarely used in practice. In most applications, heuristic reward design is carefully performed with potentially biased reward functions. In some large-scale practical RL tasks [10, 41, 42], the reward functions are heavily handcrafted based on prior knowledge. Besides reward engineering, a distinct line of work applies uninformed exploration algorithms like count-based reward shaping or intrinsic rewards to encourage greater state visitation [32, 31, 9, 11, 15, 27, 34, 33, 28, 21]. Importantly, these methods optimize for the worst case, as they try to cover *all* states since the exact location of the reward is unknown. In contrast, we look at the problem of incorporating domain knowledge via reward shaping into the exploration process.

3 Overview

We consider an episodic Markov Decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}^\star, r, H)$ where $\mathcal{S}$ corresponds to the state space, $\mathcal{A}$ is the action space, $\mathbb{P}^\star$ is the transition operator, $r : \mathcal{S} \times \mathcal{A} \to [0,1]$ is the reward function and H is the problem horizon. We use $|\mathcal{S}|, |\mathcal{A}|$ to denote the number of state and actions, and we use $\mathbb{P}^\star(\cdot|s,a)$ to denote the transition probability of state action pair (s,a). The value function $V^\pi(s_0)$ of a policy π starting at an initial state s_0 is defined as $V^\pi(s_0) := \mathbb{E}_\pi\left[\sum_{h=0}^H r(s_h, a_h)\right]$, where $\mathbb{E}_\pi$ denote the transition dynamic of $\mathcal{M}$ under π. Similarly, $V^\star(s_0)$ is the value function of the optimal policy $\pi^\star$. We consider a sequential interaction between a learner and the MDP $\mathcal{M}$ occurring in rounds indexed by $t \in \mathbb{N}$. At the start of round t the learner selects a policy π_t that is used to gather a sample trajectory from $\mathcal{M}$. As is standard in the literature, we measure the learner's performance up to round T by $\text{Regret}(T) := \sum_{t=1}^T V^\star(s_0) - V^{\pi_t}(s_0)$.

Our objective is to show that knowing a reward shaping term $\widetilde{V}$ allows for the design of a simple algorithm, that with high probability having a sublinear regret upper bound with a leading term that depends on an "effective state space" of $\mathcal{M}$ which is determined by the quality of a reward shaping term $\widetilde{V}$ and the nature of the particular MDP being solved. This pruned effective state space can be much smaller than $|\mathcal{S}|$. Additionally, we will show an improved horizon dependence as the shaping term $\widetilde{V}$ allows for the bonus terms to be smaller and therefore decay faster thus replacing horizon factors of H with $\beta\widetilde{V}$ ones.

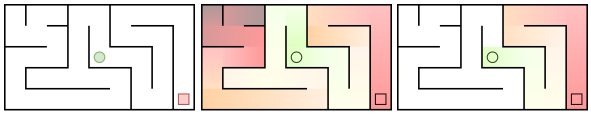


Figure 1: Reward shaping can allow for reduction of the "effective" state space size. In the above maze environment, the initial state is in the middle of the maze so finding the goal only requires solving half of the maze **(Left)**. While uninformed state covering exploration **(Middle)** would go both directions since it has no knowledge of goal location, effective shaping **(Right)** would allow for halving of the effective state space.

To more intuitively understand how reward shaping can allow for the consideration of a reduced effective state space, consider the maze environment in Fig. 1 with agent starting in the middle. Without knowing where the goal is, an uninformed exploration algorithm needs to explore nearly the entire maze. However, an effectively incorporated reward shaping term $\widetilde{V}$ would allow an algorithm to prune out the entire left half of the grid, effectively halving the effective state space size. This can happen in two ways —firstly if the shaping can directly indicate that certain actions are sub-optimal then entire regions of the space can be eliminated from search. Secondly, even if states are not eliminated, if their corresponding bonuses are scaled according to their shaped value $\widetilde{V}$ from the shaping rather than uniformly with H, it limits unnecessary exploration towards suboptimal states.

Based on this intuition, we introduce UCBVI-Shaped, a modification of the UCBVI algorithm [7] that uses an additional reward shaping term $\widetilde{V}$ in two ways: (1) Bonus scaling: $\widetilde{V}$ is used to provide an exploration bonus that combines inverse state visitation counts with reward shaping. This allows for the reduction of overoptimism and the dropping of the horizon H dependence (2) Value projection: the shaping term is used to clip the empirical value function $\widehat{V}_h(s)$, which enables pruning unnecessary

elements of the state space and allows for a bound that depends only on the effective state space size. Informally, our main result for UCBVI-Shaped can be summarized as follows:

Theorem 3.1 (Main Informal)**.** *With probability at least* $1-\delta$*, the regret of UCBVI - Shaped satisfies*[1]

$$\text{Regret}(T) = \mathcal{O}\left(B(\widetilde{V},\mathcal{M})\log(T/\delta) + \text{Regret}-\text{UCBVI}(\mathcal{S}_{\text{remain}},\mathcal{A},T)\right).$$

Here, $\mathcal{S}_{\text{remain}}$ *corresponds to the set of states in* $\mathcal{M}$ *where the information contained in* $\widetilde{V}$ *was not enough to rule them out, and* $\text{Regret}-\text{UCBVI}(\mathcal{S}_{\text{remain}}\mathcal{A},T)$ *is a UCBVI regret upper bound function evaluated on* $\mathcal{S}_{\text{remain}}$ *and* $\mathcal{A}$*.* $B(\widetilde{V},\mathcal{M})$ *is a time-agnostic complexity measure that depends solely on the* $\mathcal{M}$ *and the quality of the shaping* $\widetilde{V}$*.*

We will define $\mathcal{S}_{\text{remain}}$ and $B(\widetilde{V},\mathcal{M})$ and describe their relationship to $\widetilde{V}$ precisely in Section 5. If $\mathcal{S}_{\text{remain}} \ll \mathcal{S}$, the regret of UCBVI - Shaped can be substantially smaller than $\text{Regret}-\text{UCBVI}(\mathcal{S},\mathcal{A})$, particularly since the first term of the regret grows logarithmically while the second scales with $\sqrt{T}$. We state our assumptions next

Assumption 1. *The quality of the shaping term* $\widetilde{V}$ *is described by a parameter* β*. We assume access to shaping "value function" estimators* $\widetilde{V}_h : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ *such that,* $\widetilde{V}_h(s) \le V^\star_h(s) \le \beta\widetilde{V}_h(s)$*, for all* $s\in\mathcal{S}$*,* $h\in[H]$ *and for some* $\beta\ge 1$*.*

Assumption 2. *We assume the reward functions satisfy* $r(s,a)\in[0,1], \forall(s,a)\in\mathcal{S}\times\mathcal{A}$*.*

Assumption 3. *The states* $\mathcal{S}$ *are* $h-$*indexed i.e. the states reachable at time* $h\in[H]$ *are disjoint from the states reachable at time* $h'\in[H]$ *when* $h\neq h'$*.*

Contributions. Our analysis introduces the notions of pseudosuboptimal and path-pseudosuboptimal states to quantify the "effective" size of the state space as a function of the quality of the shaping term $\widetilde{V}$, and we use these notions to show how UCBVI-Shaped can attain significantly improved regret rates. We further show that incorporating shaping into the exploration bonus term improves the horizon-dependence in the bound.

4 The UCBVI-Shaped Algorithm

To perform analysis of reward shaping, we build on the framework of the UCBVI algorithm [7]. This forms a convenient base algorithm for incorporating shaped rewards in a way that admits faster learning while maintaining asymptotic performance (as we will discuss in Section 5). Our algorithm, UCBVI-Shaped, is a combination of two changes to the upper confidence bound algorithm - first we modify the bonus scaling to depend on $\widetilde{V}$ and second, we introduce a projection subroutine into value iteration, that is implemented as a standard clipping operation.

Algorithm 1 UCBVI - Shaped

```
Input reward function r (assumed to be known), confidence parameters
for t = 1, ..., T
    Compute P̂_t using all previous empirical transition data as P̂_t(s'|s,a) := N_h^t(s,a,s') / N_h^t(s,a), ∀h,s,a,s'.
    Compute reward bonus b_h^t(s,a) from Eqn. 1                    ▷ Bonus scaling
    Run Value-Iteration with Projection (Algorithm 2).
    Set π_t as the returned policy of VI.
End for
```

As in standard UCBVI, we define $N_h^t(s,a)$ to be the visitation count for the state-action pair (s,a) at iteration $t-1$ for horizon h: $N_h^t(s,a) := \sum_{i=0}^{t-1}\mathbf{1}\left\{(s_h^i,a_h^i)=(s,a)\right\}$. Similarly to $N_h^t(s,a)$, we use $N_h^t(s,a,s') := \sum_{i=0}^{t-1}\mathbf{1}\left\{(s_h^i,a_h^i,s_{h+1}^i)=(s,a,s')\right\}$ as the visitation count of state-action pair (s,a) and the subsequent state s'. We then use $\widehat{\mathbb{P}}_t(s'|s,a) := \frac{N_h^t(s,a,s')}{N_h^t(s,a)}$, $\forall h,s,a,s'$ to denote the empirical transition

[1] Our main result, Theorem 5.2, is slightly more complex than this statement. We have chosen this simplified form to aid the reader to form the right intuition.

kernels at iteration t. UCBVI uses value iteration with the empirical transition function $\widehat{\mathbb{P}}_t$ and a reward function augmented with an exploration bonus, given by $r_h + b_h^t$. This is defined as a dynamic programming procedure that starts at H and then proceeds backward in time to $h = 0$, updating according to Algorithm 1 and 2. The key algorithmic changes between UCBVI and UCBVI-Shaped are highlighted in blue - (1) scaling bonus b_h^t by the shaping term $\widetilde{V}$ and (2) projecting the empirical value function $\widehat{V}_h^t(s)$ by an upper bound based on the sandwiched shaping, $\beta\widetilde{V}(s,a)$.

Algorithm 2 Value Iteration with Projection

1: **Input** $\left\{\widehat{P}_t, r + b_h^t\right\}_{h=0}^{H-1}$.
2: $\widehat{V}_H^t(s) = 0,\ \forall s, \widehat{Q}_h^t(s,a)$
3: **While** not converged
4: $\quad \widehat{Q}_h^t(s,a) = \min\left\{r_h(s,a) + b_h^t(s,a) + \widehat{\mathbb{P}}_t(\cdot|s,a)\cdot\widehat{V}_{h+1}^t, H\right\}$
5: $\quad \widehat{V}_h^k(s,a) = \min\left(\max_a \widehat{Q}_h^t(s,a), \beta\widetilde{V}(s,a)\right)$ ▷ Value projection
6: $\pi_h^t(s) = \arg\max_a \widehat{Q}_h^t(s,a),\ \forall h,s,a.$

5 Analyzing UCBVI-Shaped

In this section, we will derive our main result on the sample complexity of the UCBVI-Shaped algorithm, and along the way introduce pseudosuboptimal and path-pseudosuboptimal states as a tool for deriving bounds that depend only on the "effective" state space as determined by the provided reward shaping. We will first introduce some notation to use in our analysis. As described in the previous section, UCBVI-Shaped proceeds in rounds. At the beginning of round t, the learner has access to an empirical model $\widehat{\mathbb{P}}_t$ built from the data collected up to iteration $t-1$. The bonus terms we consider are built by taking the empirical second moment of $\widetilde{V}$. This is related to the definition of bonus_2 in [7]. Importantly, the empirical value functions $\widehat{V}_h^t$ are clipped above by $\beta\widetilde{V}_h$.

$$b_h^t(s,a) = 16\beta\sqrt{\frac{\widehat{\mathbb{E}}_{s'\sim\widehat{\mathbb{P}}_t(\cdot|s,a)}[\widetilde{V}_{h+1}^2(s')|s,a]\ln\frac{2|\mathcal{S}||\mathcal{A}|}{\delta}}{N_h^t(s,a)}} + \frac{12\widetilde{V}^{\max}}{N_t(s,a)}\ln\frac{2|\mathcal{S}||\mathcal{A}|t}{\delta} \tag{1}$$

Where $\widetilde{V}_h^{\max} = \max_{s'}\widetilde{V}_h(s')$ and $\widetilde{V}^{\max} = \max_{s',h'}\widetilde{V}_{h'}(s')$.

Despite clipping and a modified bonus term, the $\widehat{Q}$ and $\widehat{V}$ values of UCBVI-Shaped are optimistic:

Lemma 5.1 (Optimism)**.** *With probability at least $1-\delta$ we have*

$$\widehat{V}_0^t(s_0) \geq V_0^\star(s_0),\ \forall s_0 \in \mathcal{S}; \qquad \textit{and} \qquad \widehat{Q}_h^t(s,a) \geq Q_h^\star(s,a), \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}. \tag{2}$$

for all $t \in \mathbb{N}$, where $\widehat{V}_h^t$ is computed via Algorithm 2.

The proof of Lemma 5.1 can be found in Appendix A.3. Optimism (Lemma 5.1) and the simulation Lemma [1] implies that:

$$V^\star(s_0) - V^{\pi_t}(s_0) \leq \widehat{V}_1^t(s_0) - V^{\pi_t}(s_0) = \mathbb{E}_{\tau\sim\pi_t}\left[\sum_{h=1}^H b_h^t(s_h,a_h) + \left(\widehat{\mathbb{P}}_t(\cdot|s_h,a_h) - \mathbb{P}^\star(\cdot|s_h,a_h)\right)\cdot\widehat{V}_{h+1}^{\pi_t}\right]. \tag{3}$$

We now turn our attention to characterize the information contained in $\widetilde{V}$.

5.1 Pruning of the State Space

Consider the following "surrogate" Q functions, $\widetilde{Q}_h^l, \widetilde{Q}_h^u : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ induced by $\widetilde{V}$ via:

$$\widetilde{Q}_h^l(s,a) = \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}\left[r(s,a) + \widetilde{V}_{h+1}(s')\right], \qquad \widetilde{Q}_h^u(s,a) = \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}\left[r(s,a) + \beta\widetilde{V}_{h+1}(s')\right].$$

By Assumption 1, we can box $Q^\star$ as $Q_h^\star(s,a) \in \left[\widetilde{Q}_h^l(s,a), \widetilde{Q}_h^u(s,a)\right]$. The basis of our results is the following observation. If an action a satisfies $\widetilde{Q}_h^u(s,a) < V_h^\star(s)$ for state s, then a is a suboptimal

action for state s. Given a parameter $\Delta > 0$, we say that an action a is $\Delta-$pseudosuboptimal[2] for state s if $V_h^\star(s) \geq \Delta + \widetilde{Q}_h^u(s,a)$. We denote the set of $\Delta-$pseudosuboptimal state action pairs as:

$$\text{PseudoSub}_\Delta = \{(s,a) \in \mathcal{S} \times \mathcal{A} \text{ s.t. } V_h^\star(s) \geq \Delta + \widetilde{Q}_h^u(s,a)\}$$

The intuition we want to capture is that all states $\tilde{s}$ that can be accessed only through traversing a state action pair in PseudoSub_Δ, can be safely ignored, because we can determine their suboptimality, as soon as the state action pairs (or set of state action pairs) in PseudoSub_Δ that lead to $\tilde{s}$ are visited enough.

Now we define the set of Δ-path-pseudosuboptimal states as the states that can be reached only by traversing a $\Delta-$suboptimal state action pair:

$$\text{PathPseudoSub}_\Delta = \{s \in \mathcal{S} \text{ s.t. all feasible paths from initial states to } s \text{ intersect } \text{PseudoSub}_\Delta\}.$$

We will show that states in $\text{PathPseudoSub}_\Delta$ will not be explored by UCBVI-Shaped after a few iterations. To provide intuition of why this is the case,note the support of the optimal policy distribution does not contain any state in $\text{PathPseudoSub}_\Delta$ for all $\Delta > 0$, i.e., $\text{Support}(d^{\pi^\star}(s,a)) \cap \text{PathPseudoSub}_\Delta = \emptyset, \forall \Delta > 0$.

For any state action pair (s,a) define its set of neighboring states $\text{Neighbor}(s,a)$ as the set of states with nonzero probability in $\mathbb{P}^\star(\cdot|s,a)$. By definition of $\text{PathPseudoSub}_\Delta$, for all $(s,a) \in \mathcal{S} \backslash ((\text{PathPseudoSub}_\Delta \times \mathcal{A}) \cup \text{PseudoSub}_\Delta)$, we have:

$$\text{Neighbor}(s,a) \subseteq \mathcal{S} \backslash \text{PathPseudoSub}_\Delta. \tag{4}$$

In other words, the neighborhood of any state action pair (s,a) whose state is not in $\text{PathPseudoSub}_\Delta$ and such that (s,a) is not $\Delta-$pseudo-suboptimal, cannot be in $\text{PathPseudoSub}_\Delta$. We also introduce the notion of Boundary pseudosuboptimal state action pairs to capture the set of state action pairs that are $\Delta-$suboptimal but whose states are not in $\text{PathPseudoSub}_\Delta$.

$$\text{BoundaryPseudoSub}_\Delta = \{(s,a) \in \text{PseudoSub}_\Delta \text{ and } s \notin \text{PathPseudoSub}_\Delta\}.$$

One of these states has to be traversed by any trajectory that contains any state in $\text{PathPseudoSub}_\Delta$.

Although we only use $\widetilde{Q}^u$ in the definition of PseudoSub_Δ, $\text{PathPseudoSub}_\Delta$, and $\text{BoundaryPseudoSub}_\Delta$, the size of these sets is modulated by the scale of β and the width of the intervals $\left[\widetilde{Q}^l(s,a), \widetilde{Q}^u(s,a)\right]$. We will show that (in the notation of Theorem 3.1) $\mathcal{S}_{\text{prunned}} \approx \mathcal{S} \backslash \text{PathPseudoSub}_\Delta$. With this notation, the formal version of our main results is stated as follows.

Theorem 5.2. *With probability at least $1-3\delta$, the regret of UCBVI-Shaped is upper bounded by*

$$\sum_{t=1}^{T} V^\star(s_0) - V^{\pi_t}(s_0) = \mathcal{O}\Bigg(\min_\Delta \Bigg(H\beta \widetilde{V}^{\max} |\mathcal{S} \backslash \text{PathPseudoSub}_\Delta| \sqrt{|\mathcal{A}| T \ln \frac{\widetilde{V}^{\max} |\mathcal{S}||\mathcal{A}|T}{\delta}} + \beta^2 \left(\widetilde{V}^{\max}\right)^2 \ln \frac{\widetilde{V}^{\max} |\mathcal{S}||\mathcal{A}|T}{\delta} H^{1/2} \min\left(\frac{|\mathcal{S}|^{3/2}|\mathcal{A}|}{\Delta}, \frac{H^{3/2}|\text{BoundaryPseudoSub}_\Delta|}{\Delta^2} \right) \Bigg) \Bigg).$$

for all $T \in \mathbb{N}$.

Theorem 5.2 instantiates the desiderata of Theorem 3.1. Although this regret upper bound cannot be decomposed into a sum of two term as in Theorem 3.1. For any fixed Δ, the regret upper bound does have two components, one where $\mathcal{S}_{\text{remain}}$ can be identified with $|\mathcal{S} \backslash \text{PathPseudoSub}_\Delta|$ and a second one that scales logarithmically in T. In the next section we flesh out the steps in the proof of this result. The full argument can be found in Appendix A. The bonus scaling also allows us to ameliorate the horizon dependence of the upper bound. Instead of obtaining a H^2 dependence as the bound of theorem 7.1 in [1], the first term depends on $\beta H \widetilde{V}^{\max} \leq \beta H \max_s V^\star(s)$. Notice that the state dependence in the second term may be only be of order $\log(|\mathcal{S}|)$ if there is a Δ such that $|\text{BoundaryPseudoSub}_\Delta| \ll |\mathcal{S}|$ (albeit at the cost of a quadratic dependence on $1/\Delta^2$). Setting Δ to a value smaller than the minimum gap recovers the UCBVI regret upper bounds Theorem 7.1 in [1] showing Theorem 5.2 is a strict generalization.

[2] We add the qualifier pseudo to our name to distinguish between suboptimality as captured by our surrogate Q values and true suboptimality between the true values of $Q_h^\star$.

5.2 Proof Intuitions and Sketch for Theorem 5.2

Improved Horizon Dependence. An empirical Bernstein bound shows that adding a bonus scaling (up to low order terms) with $\mathcal{O}\left(\sqrt{\widehat{\mathrm{Var}}_{s'\sim\widehat{\mathbb{P}}_t(\cdot|s,a)}(V^\star_{h+1}(s')|s,a)\ln\frac{2|\mathcal{S}||\mathcal{A}|t}{\delta}/N^t_h(s,a)}\right)$ is sufficient to ensure optimism. Since $V^\star$ is not known, this variance term can be substituted by the empirical second moment of $\beta\widetilde{V}_{h+1}$. Finally the scaling of these terms can be upper bounded by $\beta\widetilde{V}^{\max}$. Without knowledge of $\widetilde{V}$, achieving this scaling would prove more challenging, since the only proxy for $V^\star_{h+1}$ available is $\widehat{V}^t_{h+1}$ or H both of which may vastly overestimate it.

State Pruning. The value function clipping mechanism ensures that $\mathbb{E}_{s'\sim\widehat{\mathbb{P}}_h(\cdot|s,a)}\left[\widehat{V}^t_{h+1}(s')\right] \leq \beta\mathbb{E}_{s'\sim\widehat{\mathbb{P}}_h(\cdot|s,a)}\left[\widetilde{V}_{h+1}(s')\right]$ and therefore the empirical gap between $\widehat{Q}^t_h(s,a)$ and $V^\star(s)$ is decreasing at a rate of at most $\mathcal{O}\left(\beta\widetilde{V}^{\max}/\sqrt{N^t_h(s,a)}\right)$. Since optimism ensures that $\widehat{Q}^t_h(s,\pi^\star(a)) \geq V^\star(s)$, once $\widehat{Q}(s,a) < V^\star(s)$, action a will not be pulled anymore. Thus for any Δ, the number of times any state action pair in $\mathrm{PseudoSub}_\Delta$ may be visited by UCBVI-Shaped is upper bounded by $\beta^2\left(\widetilde{V}^{\max}\right)^2/\Delta^2$. Since any visit to a state action pair in $\mathrm{PathPseudoSub}_\Delta \times \mathcal{A} \cup \mathrm{PseudoSub}_\Delta$ requires a visit to a state in $\mathrm{BoundaryPseudoSub}_\Delta$, this allows us to bound the total number of visits to a state (or a trajectory containing such a state) in $\mathrm{PathPseudoSub}_\Delta \times \mathcal{A} \cup \mathrm{PseudoSub}_\Delta$ by $H|\mathrm{BoundaryPseudoSub}_\Delta|\beta^2\left(\widetilde{V}^{\max}\right)^2/\Delta^2$. Finally, we can split the regret decomposition of Eqn. 3 into two sums, one over state action pairs in $\mathcal{U} = \mathrm{PathPseudoSub}_\Delta \times \mathcal{A} \cup \mathrm{PseudoSub}_\Delta$ (or trajectories intersecting $\mathcal{U}$) and a second one over state action pairs in $\mathcal{U}^c$ (or trajectories without intersection with $\mathcal{U}$). We can then apply the upper bound on the visitation of state action pairs (or trajectories) in $\mathrm{PathPseudoSub}_\Delta \times \mathcal{A} \cup \mathrm{PseudoSub}_\Delta$ to derive a regret upper bound over these states scaling with $1/\Delta$. Using the trajectory decomposition yields the term scaling with $1/\Delta^2$ (but having only logarithmic state dependence) in Theorem 5.2. We handle the upper bound for the term summing over intersections with $\mathcal{U}^c$ by noting that Eqn. 4 implies the transition operators over state action pairs in $\mathcal{U}^c$ have support only over $\mathcal{S}\backslash\mathrm{PathPseudoSub}_\Delta$. This implies the error terms $\left(\widehat{\mathbb{P}}_t(\cdot|s_h,a_h) - \mathbb{P}^\star(\cdot|s_h,a_h)\right)\cdot\widehat{V}^{\pi_t}_{h+1}$ can be bounded by a factor depending on $|\mathcal{S}\backslash\mathrm{PathPseudoSub}_\Delta|$ and not $|\mathcal{S}|$. Detailed proofs can be found in Appendix A.

6 Numerical Simulations

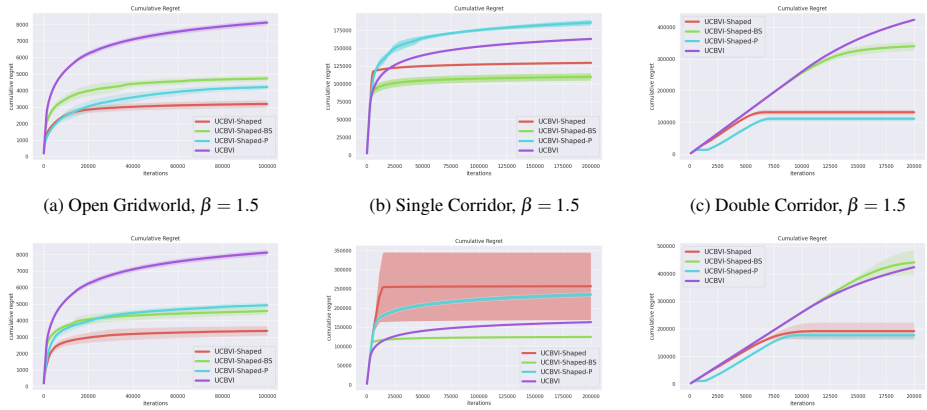
To show the practical relevance of our analysis on reward shaping we perform some numerical simulations on a family of maze environments with tabular state-action representations, as shown in Fig. 2. These environments have deterministic dynamics and reward. The reward is 0 at all states except a goal sink state, which has reward 1. These simulations are aimed at studying the following questions: **(1)** Does reward shaping improves sample complexity in these types of maze environments over uninformed UCBVI? **(2)** What is the relative importance of the bonus reweighting and the $\widetilde{V}$ projection? **(3)** How does the "suboptimality" of $\widetilde{V}$ impact the resulting sample complexity? **(4)** Does introducing decayed shaping actually allow for policy convergence? In these experiments, $\widetilde{V}$ is constructed by scaling the optimal value function $V^\star$ by per-state scaling factors sampled independently within the range β. .

(a) Value function of Gridworld (b) Value function of Single Corridor (c) Value function of Double Corridor

Figure 2: Environments used for numerical simulations. **(Left)** Open Gridworld **(Middle)** Single corridor, agent starts bottom right has to reach a goal in top left **(Right)** Double corridor, agent starts in the middle and has to reach a goal on the left, with many irrelevant states on the right hand side

6.1 Does reward shaping help direct exploration over optimism under uncertainty?

We conducted numerical simulations on tabular environments to understand how reward shaping via UCBVI-Shaped provides benefits over standard UCBVI. Fig. 3 shows cumulative regret accumulated with different variants of UCBVI-Shaped (with both projection and bonus scaling), UCBVI-Shaped-P (with only projection), UCBVI-Shaped-BS (with only bonus scaling), UCBVI (standard UCBVI without shaping, as described in [7]). This is benchmarked across the three environments described in Fig. 2, with various levels of imperfect shaping applied by varying $\beta = \{1.5, 1.9\}$. As seen from Fig. 3, across all environments UCBVI-shaped with projection and bonus scaling performs most favorably, followed typically by UCBVI-Shaped-P, followed by UCBVI-Shaped-BS and UCBVI, suggesting that reward shaping can significantly help with learning efficiency.



(a) Open Gridworld, $\beta = 1.5$ (b) Single Corridor, $\beta = 1.5$ (c) Double Corridor, $\beta = 1.5$

(d) Open Gridworld, $\beta = 1.9$ (e) Single Corridor, $\beta = 1.9$ (f) Double Corridor, $\beta = 1.9$

Figure 3: Cumulative regret for learning in various environments with varying amounts of shaping, as compared with UCBVI, and ablations UCBVI-Shaped-BS (no projection) and UCBVI-Shaped-P (no bonus scaling)

6.2 How does the effectiveness of reward shaping vary across environments?

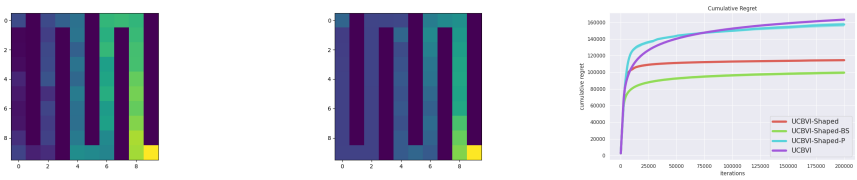
We next conducted some numerical simulations across environments to understand how the nature of the environment itself affects the sample complexity of learning with shaping via UCBVI-Shaped. As shown in Fig. 4, we see that UCBVI shaped can be very effective in environments with many irrelevant sub-optimal paths like the double corridor environment in Fig. 2, but is relatively less effective in environments where all exploration is directed the same way such as the single corridor. Even incorrect but optimistic shaping will provide guidance towards the goal, making UCBVI relatively less dominant in the single corridor environment as compared to double corridor. This suggests that in environments where ruling out an entire part of the exploration space is easy from the shaping, we can expect to see larger benefits, as in double corridor.

6.3 How does the suboptimality of reward shaping affect learning?

We next compared how different levels of suboptimality of the β sandwich term in the reward shaping affect cumulative regret across environments. As shown in Fig. 5, we see that for environments with open paths, the shaping degradation has minimal negative effect until it gets very suboptimal. On the other hand, for corridor and double corridor, where there are only a few paths to the goal, suboptimal reward shaping

(a) Single Corridor (b) Double Corridor

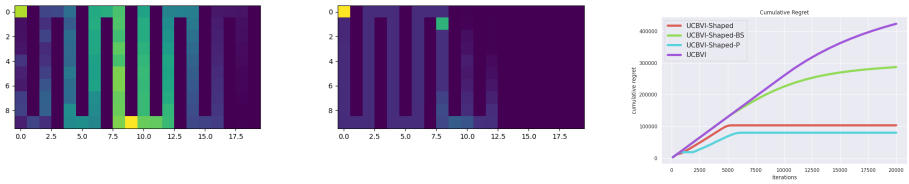
Figure 5: Effect of suboptimality of reward shaping on the performance of UCBVI-Shaped. While $\beta = 1.2, 1.5$ don't make much of a difference, very large β leads to performance degradation



(a) Heatmap of intermediate visitations of UCBVI on single corridor (no shaping)

(b) Heatmap of intermediate visitations of UCBVI-Shaped on single corridor

(c) Learning progress of UCBVI-Shaped compared to UCBVI on single corridor



(d) Heatmap of intermediate visitations of UCBVI on double corridor (no shaping)

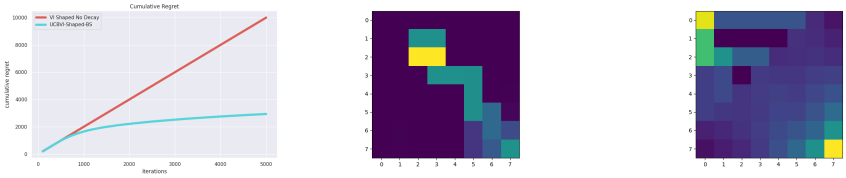
(e) Heatmap of intermediate visitations of UCBVI-Shaped on double corridor

(f) Learning progress of UCBVI-Shaped compared to UCBVI on double corridor

Figure 4: Visualization of how different environments are affected by reward shaping differently. (Left) intermediate visitations (Right) learning progress of UCBVI vs UCBVI-Shaped. The single corridor environment on the top sees much smaller gains for UCBVI-Shaped compared to double corridor environment.

along those paths can significantly hamper progress. This suggests that suboptimal reward shaping can hurt learning, but mainly when β is quite large and in environments with bottlenecks.

6.4 Does decaying suboptimal reward shaping help over standard shaped rewards?



(a) Quantitative comparison of effect of bonus decay

(b) Visualization of non decaying agent getting stuck

(c) Visualization of decaying agent (UCBVI-Shaped) reaching goal

Figure 6: Effect of bonus decay on the performance of UCBVI-Shaped vs standard reward shaping under shaping misspecification. As we can see, without decay the agent can stuck in arbitrarily sub-optimal points, whereas with decay the agent easily converges to an optimal solution.

We ran simulations to understand whether the adaptive decay of $\widetilde{V}$ by $\frac{1}{\sqrt{N(s,a)}}$ actually provides tangible benefits, specifically when the shaping is suboptimal. We compared UCBVI-Shaped with a variant where $\widetilde{V}$ is added to the reward linearly, as done in many practical RL implementations. As seen in Fig. 6, with suboptimal shaping, UCBVI-shaped dampens the effect of the suboptimal shaping and succeed, whereas simple addition of shaping leads to the agent becoming trapped. This suggests that not only can reward shaping improve sample efficiency, but incorporating reward shaping via UCBVI-Shaped can mitigate the potential bias from suboptimal shaping.

7 Discussion

In this work, we take a step towards formally analyzing the benefits of reward shaping, proving that effective reward shaping can lead to more efficient learning than uninformed exploration strategies. In our analysis, we study an algorithm that incorporates reward shaping into a modified version of UCBVI, using it to modify bonuses and value function projection. Our analysis shows that incorporating shaped rewards allows for pruning significant parts of the state space and sharpening of optimism in a task directed way. This reduces the dependence of the regret bound on the state space size and on the horizon, depending on the quality of the shaping term and parameters of the MDP. This shows how reward shaping can direct exploration and provide significant sample complexity benefits while retaining asymptotic performance.

References

[1] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 2019.

[2] A. Agarwal, M. Henaff, S. Kakade, and W. Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 33: 13399–13412, 2020.

[3] S. Agrawal and R. Jia. Posterior sampling for reinforcement learning: worst-case regret bounds. *arXiv preprint arXiv:1705.07041*, 2017.

[4] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[5] P. Auer, T. Jaksch, and R. Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

[6] A. Ayoub, Z. Jia, C. Szepesvari, M. Wang, and L. Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

[7] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

[8] Y. Bai, T. Xie, N. Jiang, and Y.-X. Wang. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[10] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[11] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[12] Q. Cai, Z. Yang, C. Jin, and Z. Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

[13] Y. Efroni, N. Merlis, M. Ghavamzadeh, and S. Mannor. Tight regret bounds for model-based reinforcement learning with greedy policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[14] R. Fruit, M. Pirotta, A. Lazaric, and R. Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *International Conference on Machine Learning*, pages 1578–1586. PMLR, 2018.

[15] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

[16] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

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

[18] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

[19] S. M. Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, 2003.

[20] G. Li, L. Shi, Y. Chen, Y. Gu, and Y. Chi. Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] K. Li, A. Gupta, A. Reddy, V. H. Pong, A. Zhou, J. Yu, and S. Levine. Mural: Meta-learning uncertainty-aware rewards for outcome-driven reinforcement learning. In *International Conference on Machine Learning*, pages 6346–6356. PMLR, 2021.

[22] A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

[23] P. Ménard, O. D. Domingues, X. Shang, and M. Valko. Ucb momentum q-learning: Correcting the bias without forgetting. In *International Conference on Machine Learning*, pages 7609–7618. PMLR, 2021.

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

[25] A. Pacchiano, P. Ball, J. Parker-Holder, K. Choromanski, and S. Roberts. On optimism in model-based reinforcement learning. *arXiv preprint arXiv:2006.11911*, 2020.

[26] A. Pacchiano, P. Ball, J. Parker-Holder, K. Choromanski, and S. Roberts. Towards tractable optimism in model-based reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 1413–1423. PMLR, 2021.

[27] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[28] R. Raileanu and T. Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkg-TJBFPB.

[29] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[30] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550 (7676):354–359, 2017.

[31] B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[32] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. X. Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pages 2753–2762, 2017.

[33] H. Van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, and J. Tsang. Hybrid reward architecture for reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/1264a061d82a2edae1574b07249800d6-Paper.pdf.

[34] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3540–3549. JMLR. org, 2017.

[35] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[36] Y. Wang, R. Wang, S. S. Du, and A. Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

[37] K. Yang, L. Yang, and S. Du. Q-learning with logarithmic regret. In *International Conference on Artificial Intelligence and Statistics*, pages 1576–1584. PMLR, 2021.

[38] L. Yang and M. Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

[39] L. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

[40] Z. Yang, C. Jin, Z. Wang, M. Wang, and M. I. Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces. *arXiv preprint arXiv:2011.04622*, 2020.

[41] D. Ye, G. Chen, W. Zhang, S. Chen, B. Yuan, B. Liu, J. Chen, Z. Liu, F. Qiu, H. Yu, Y. Yin, B. Shi, L. Wang, T. Shi, Q. Fu, W. Yang, L. Huang, and W. Liu. Towards playing full moba games with deep reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 621–632. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/06d5ae105ea1bea4d800bc96491876e9-Paper.pdf.

[42] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1094–1100. PMLR, 2019. URL http://proceedings.mlr.press/v100/yu20a.html.

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

[44] A. Zanette, D. Brandfonbrener, E. Brunskill, M. Pirotta, and A. Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1964. PMLR, 2020.

[45] Y. Zhai, C. Baek, Z. Zhou, J. Jiao, and Y. Ma. Computational benefits of intermediate rewards for goal-reaching policy learning. *Journal of Artificial Intelligence Research*, 73:847–896, 2022.

[46] Z. Zhang, Y. Zhou, and X. Ji. Almost optimal model-free reinforcement learningvia reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207, 2020.

[47] D. Zhou, J. He, and Q. Gu. Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pages 12793–12802. PMLR, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] This work does not actually use human subjects, and is largely done in simulation. But we have included a discussion in Section 7
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A] Math is used as a theory/formalism, but we don't make any provable claims about it.

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix A for link to URL and run instructions in the README in the github repo.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All plots were created with 3 random seeds with std error bars.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Envs we used are cited in section 6

(b) Did you mention the license of the assets? [Yes] This is in Appendix B

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We published the code and included all environments and assets as a part of this

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We used three open source domains and collected our own data on these domains.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] We did not include full text since we didn't use an exact script, but we summarized the instructions and included images of the environments used.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Only human involvement was data collection with our system.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Human testers were volunteers