Oracle-Efficient Adversarial Reinforcement Learning via Max-Following

Sikata Bela Sengupta[*]
University of Pennsylvania
sikata@seas.upenn.edu

Zakaria Mhammedi[*]
Google Research
mhammedi@google.com

Teodor V. Marinov[*]
Google Research
tvmarinov@google.com

Abstract

Learning the optimal policy in reinforcement learning (RL) with large state and action spaces remains a notoriously difficult problem from both computational and statistical perspectives. A recent line of work addresses this challenge by aiming to compete with, or improve upon, a given base class of policies. One approach, known as max-following, selects at each state the policy from the base class whose estimated value function is highest. In this paper, we extend the max-following framework to the setting of regret minimization under adversarial initial states and limited feedback. Our algorithm is oracle-efficient, achieves no-regret guarantees with respect to the base class (and to the worst approximate max-following policy), and avoids any dependence on the size of the state or action space. It also attains the optimal rate in terms of the number of episodes. Additionally, we establish a lower bound on the regret of any max-following algorithm as a function of β, a parameter that quantifies the approximation slack in the benchmark policy class. Finally, we empirically validate our theoretical findings on the Linear Quadratic Regulator (LQR) problem.

1 Introduction

In many practical reinforcement learning applications, handling large or continuous state and action spaces is essential. However, beyond the tabular MDP setting, designing algorithms that are both computationally and statistically efficient remains a major challenge Kane et al. [2022]. To address these difficulties, several lines of work have explored imposing additional structural assumptions on the MDP in order to avoid dependence on the size of the state space. For example, Jin et al. [2020] provide an algorithm (LSVI-UCB) with a polynomial run-time and sample complexity with regret independent of the size of the state or action space for Linear MDPs (with linear dynamics and rewards). An assumption that subsumes linear MDPs is that of finite coverability Xie et al. [2023] which includes classes of MDPs like Low-Rank MDPs and Block MDPs as well. Coverability asks for the existence of a distribution with good concentrability which loosely asks that the data distribution (say from an offline dataset) uniformly covers all the possible induced state distributions from some policy in the class. Mhammedi et al. [2024] recently showed that given local-simulator access, if the optimal policy's state-action value function is realizable (Q^*-realizable), then low-coverability MDPs can be learned in a sample-efficient manner. Moreover, Xie et al. [2023] provide the complexity measure known as the Sequential Extrapolation Coefficient (SEC) which subsumes coverability, and measures that make various representational assumptions on the Bellman residuals like Bellman-Eluder Jin et al. [2021] and Bellman/Bilinear Rank Jiang et al. [2017], Du et al. [2021] for sample-efficient online RL.

Due to the difficulty of exploration from scratch in large state spaces without these structural assumptions, there has been growing interest in leveraging prior knowledge to guide exploration

[*]Equal Contribution

Preprint. Under review.

more effectively. Several works have also explored ensembling techniques in RL to better scale to large state spaces Lee et al. [2021]. Some of these methods aim to compete with the optimal policy using boosting-style approaches Brukhim et al. [2022], but they often rely on strong weak learnability assumptions. When a base class of sub-optimal policies is available, recent work has focused on competing with weaker benchmarks rather than the optimal policy itself Cheng et al. [2020], Liu et al. [2023], Hussing et al. [2024]. One such approach is max-following, where at each state, the learner selects the policy from the base class with the highest estimated value. The MaxIteration algorithm proposed by Hussing et al. [2024] provide theoretical guarantees for this strategy in settings where the initial state distribution is fixed. That is, their algorithm returns a policy that is competitive with the base class of policies and they provide examples where the max-following policy is able to significantly outperform any policy in the base class. However, in many practical scenarios, the learner may not encounter states from the same distribution across episodes. Instead, initial states may be drawn from arbitrary distributions—potentially selected by "Nature" or an adversary—requiring the learner to behave competitively without the ability to resample from a consistent starting distribution. In particular, certain safety or robotics applications may require exploration in settings with adversarially generated start states to consider worst-case guarantees. There are a variety of ways in which one could model the behavior of the adversary. In this work, we focus on initial start states because we view it as important in improving the robustness of the learner by possibly exposing it to challenging initial states or configurations.

2 Related Work

Due to the challenges of reinforcement learning in large state spaces, a growing body of work has focused on competing against weaker objectives than the optimal policy. In particular, several approaches provide performance guarantees relative to a weaker baseline in the batch or stochastic setting, where initial states are drawn from a fixed distribution. Cheng et al. [2020] introduce the max-aggregation benchmark, which performs a one-step look ahead and selects the action with the largest advantage relative to the best policy in the base class. Unlike max-following, which commits to a single policy from the base class at each state, max-aggregation allows state-dependent action selection, potentially offering more flexibility but requiring more fine-grained information. The authors note that max-following and max-aggregation are generally incomparable in terms of their guarantees. For details of the incompatibility we refer to Appendix A in Cheng et al. [2020]. They ultimately propose the MAMBA algorithm, which provably competes with the best policy in the base class for states drawn from a given initial distribution. Liu et al. [2023] build upon the work of Cheng et al. [2020] with an active state exploration criterion. It is worth noting that policy-gradient methods generally may face challenges from higher variance Wu et al. [2018], and both Cheng et al. [2020] and Liu et al. [2023] have benchmarks that depend on the bias and variance of their policy gradients.

Marinov et al. [2024] also consider a similar benchmark based on the state-wise maximum over a base class of policies in the stochastic setting, using behavioral cloning under sparse feedback. Barreto et al. [2020] study a form of generalized policy improvement in the context of transfer learning across tasks that share a common representation structure. Their benchmark selects the best action at each state according to the Q-values of the base class of policies. Hussing et al. [2024] propose an oracle-efficient algorithm, MaxIteration, for the stochastic setting, which allows the learner to compete with an approximate max-following policy class using access to a regression oracle. Hussing et al. [2024] show that any policy in this approximate max-following policy class is ϵ-competitive with the base class of policies.

Several lines of work have investigated no-regret algorithms in adversarial reinforcement learning settings. Liu et al. [2024] study regret minimization with respect to the best policy in a given class for low-rank MDPs with adversarial (i.e., arbitrary) losses and fixed transitions. They propose oracle-efficient, model-free algorithms under bandit feedback, improving upon existing regret bounds for the full-information setting. Sekhari et al. [2023] examine online imitation learning with multiple noisy experts, using selective sampling, preference-based feedback, and an online regression oracle to guide learning. They make an assumption on the realizability of the preference model, which in this context could be reduced to referring to the max-following policy. However, as shown in Hussing et al. [2024] in Observation 4.7, the parametric class of the value functions of the max-following policy can be more complex than simply the parametric class of the base class of policies. Therefore, they make a stronger assumption than we will need to make in this paper. We extend the max-following approach

to these adversarial RL settings and are able to compete with the approximate max-following policy benchmark class from Hussing et al. [2024].

Broadly, our work builds upon these results in a couple of different ways. First, our algorithm is able to handle the adversary selecting arbitrary initial start states. Second, we utilize value-based methods rather than policy-gradient based methods and so we avoid some of the associated policy-gradient bias and variance dependencies. Third, our algorithm competes against the slightly stronger benchmark of the approximate-max-following policy class. Fourth, we provide lower bounds with respect to this benchmark class (as will be specified below). Finally, we only require online learnability of the base class of policy's value functions.

3 Background

We consider a horizon H MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, H, \{P_h\}_{h\in[H]}, \{r_h\}_{h\in[H]})$, where $P_h : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ from h to $h+1$, $\mathcal{X}$ is the (potentially large) state space, $\mathcal{A}$ is the action space, H is the horizon length of each episode, and $r_H : \mathcal{X} \times \mathcal{A} \to [0,1]$. A (non-stationary) policy π is a mapping of the form $\pi : [H] \times \mathcal{X} \to \Delta(\mathcal{A})$. Throughout, we will use the notation $\{\pi_h\}_{h\in[H]}$ to denote such a policy. When a policy $\{\pi_h\}_{h\in[H]}$ is executed, it generates the trajectory $(\boldsymbol{x}_1, \boldsymbol{a}_1, \boldsymbol{x}_2, \boldsymbol{a}_2, \ldots, \boldsymbol{x}_H, \boldsymbol{a}_H)$ via the process $\boldsymbol{a}_h \sim \pi_h(\boldsymbol{x}_h)$, $\boldsymbol{x}_{h+1} \sim P_h(\cdot \mid \boldsymbol{x}_h, \boldsymbol{a}_h)$, initialized from $\boldsymbol{x}_1 = x_1$. We let $\mathbb{P}^\pi[\cdot \mid \boldsymbol{x}_1 = x_1]$ and $\mathbb{E}^\pi[\cdot \mid \boldsymbol{x}_1 = x_1]$ denote the law and expectation under this process. Similar to Liu et al. [2024], $\pi \circ_h \pi'$ will denote a policy that follows $\pi_k(\cdot \mid \cdot)$ for $k < h$, and then $\pi'_k(\cdot \mid \cdot)$ for $k \geq h$.

We define a value function for policy π, time step h, and transitions set by P, to be of the form

$$V_h^\pi(x) = \mathbb{E}^\pi\left[\sum_{s=h}^{H} r_s(\boldsymbol{x}_s, \pi_s(\boldsymbol{x}_s)) \mid \boldsymbol{x}_h = x\right].$$

Definition 3.1 (Argmax)**.** *Consider a function $f : D \to \mathbb{R}$. We define the argmax operator* $\arg\max : \{(f, D) \mid f : D \to \mathbb{R}\} \to 2^D$ *to be*

$$\arg\max_{x\in D} f(x) \;:=\; \left\{ x \in D \;\middle|\; f(x) = \sup_{y\in D} f(y) \right\}.$$

Building on Hussing et al. [2024], we are interested in studying the following setting. Consider a base class of policies Π_{Base}, where $K = |\Pi_{\mathsf{Base}}|$. This collection of policies could be gathered by any means and are not necessarily heuristic. They consider a *max-following* policy of the form

$$\pi^{\max}(x) = \pi_x(x), \quad \text{where} \quad \pi_x \in \arg\max_{\pi\in\Pi_{\mathsf{Base}}} V^\pi(x)$$

That is, this is the policy that at every state follows the action of the policy from the base class with the maximum value function at that state. More formally, due to the presence of tie-breaking they provide the following definitions.

Definition 3.2. *(Max-following policy class Hussing et al. [2024]) Let Π_{Base} be a finite set of policies. The class of max-following policies $\Pi_{\max}$ is defined*

$$\Pi_{\max} = \left\{ \pi_{1:H} \mid \forall h \in [H], \forall x \in \mathcal{X}, \exists \pi_x \in \arg\max_{\tilde{\pi}\in\Pi_{\mathsf{Base}}} V_h^{\tilde{\pi}}(x) : \;\; \pi_h(x) = \pi_x(x) \right\}.$$

In practice, it can be challenging to compete against policies in $\Pi_{\max}$ without further assumptions on the MDP. To make the RL task more tractable, Hussing et al. [2024] introduce the class of approximate max-following policies.

Definition 3.3. *(Approximate max-following policies Hussing et al. [2024]). Let Π_{Base} be a finite base policy class and let $\beta > 0$ be given. The class of β-approximate max-following policies relative to Π_{Base} is defined as*

$$\Pi_{\mathsf{Bench}}(\beta) = \{\pi \mid \forall h \in [H], \forall x \in \mathcal{X}, \exists \pi_x \in \Pi_{\beta,h}(x) : \;\; \pi_h(x) = \pi_x(x)\},$$

where for $x \in \mathcal{X}$ and $h \in [H]$:

$$\Pi_{\beta,h}(x) = \left\{ \pi \in \Pi_{\mathsf{Base}} \mid V_h^\pi(x) \geq \max_{\tilde{\pi}\in\Pi_{\mathsf{Base}}} V_h^{\tilde{\pi}}(x) - \beta \right\}.$$

Online RL framework. Our goal in this paper is to design an algorithm that produces policies that compete against the best policy in the β-approximate policy class in Definition 3.3. Unlike previous works, we allow the initial state $\boldsymbol{x}_1^{(t)}$ to be chosen by an adversary at the beginning of each episode t. After this initial state is revealed, the learner (or algorithm) interacts with the MDP $\mathcal{M}$ and generates a trajectory by executing some policy.

The learner receives feedback on the value functions of the base class of policies at every time step in the form of a loss $\ell_h^{(t)}(V)$, for an $h \in [h]$ and $\pi \in \Pi_{\mathsf{Base}}$ (for all other (h, π) the loss is set to 0) which provides feedback on how well every value function $V \in \mathcal{V}$ fits the actual return provided from the given trajectory. Ultimately, the learner is hoping to minimize the regret with respect to the *worst*-approximate max-following policy. Note that Hussing et al. [2024] show that the worst-approximate max-following policy is ϵ-competitive with the performance of the base class of policies themselves.

$$\mathrm{Reg}_T(\tilde{\pi}) = \sum_{t=1}^{T} \left(\inf_{\pi \in \Pi_{\mathsf{Bench}}(\beta)} V^{\pi}(\boldsymbol{x}_1^{(t)}) - V^{\tilde{\pi}^{(t)}}(\boldsymbol{x}_1^{(t)}) \right).$$

The learner provides this loss feedback to an online learning oracle which provides estimated value functions, which are used to construct a policy that is ultimately competitive with an approximate max-following policy.

Definition 3.4 (η-mixability, see Korotin et al. [2021], Cesa-Bianchi and Lugosi [2006])**.** *. For all $\boldsymbol{q} \in \Delta([N])$, there exists a prediction $\hat{p}$ such that for all $y \in \mathcal{Y}$ and forecasts $\hat{\boldsymbol{y}}_i \in \mathcal{Y}, \forall i \in [N]$,*

$$\ell\big(\hat{p}(t), y(t)\big) \leq -\frac{C}{\eta} \log \sum_{i=1}^{N} \boldsymbol{q}_i \exp\left(-\eta \ell\big(\hat{\boldsymbol{y}}_i(t), y(t)\big)\right).$$

Note that square-loss functions are an example of $\frac{1}{2}$-mixable loss functions.

Online learning guarantee. Vovk [1990] There exists an online learning oracle $\mathcal{O}$ over $\mathcal{V}$ such that for all $(h, \pi) \in [H] \times \Pi_{\mathsf{Base}}$ the outputs $\widehat{V}_{h,\pi}^{(1)}, \ldots, \widehat{V}_{h,\pi}^{(T)}$ of $\mathcal{O}$ in response to any sequence of η-mixable losses $\hat{\ell}_{h,\pi}^{(1)}, \ldots, \hat{\ell}_{h,\pi}^{(T)}$ over $\mathcal{V}$:

$$\sum_{t=1}^{T} \hat{\ell}_{h,\pi}^{(t)}(\widehat{V}_{h,\pi}^{(t)}) - \sum_{t=1}^{T} \hat{\ell}_{h,\pi}^{(t)}(V_h^{\pi}) \leq \frac{H^2 \log |\mathcal{V}|}{\eta}. \tag{1}$$

4 Algorithm: Behavior Cloning

At each round, our algorithm either performs an exploration step (with probability $\varepsilon_{\mathsf{exp}}$) or an 'exploitation' step. On exploration rounds, our algorithm gathers trajectory data losses (on cumulative returns vs. value function estimates) for a uniformly sampled policy and time step within an episode. It then feeds these losses to an online learning oracle which is able to provide value function estimates for these policies over different time steps. These value-function estimates are then used to follow an estimated or approximate max-following policy. On an exploitation round, this estimated max-following policy is followed for the duration of the episode.

Algorithm 1 AdversarialMaxIteration (AMI).

input: Base policy class Π_{Base}, value function class $\mathcal{V}$, and number of rounds $T \geq 1$, $\beta > 0$.

initialize: Set $K \leftarrow |\Pi_{\mathsf{Base}}|$, $\widehat{\pi}_h^{(1)} \equiv \pi_{\mathtt{unif}}, \forall h$, and $\varepsilon_{\mathsf{exp}} \leftarrow \beta^{-1}T^{-1/4}$ (probability of exploration).

for $t = 1, \ldots, T$ **do**
  Observe initial state $\boldsymbol{x}_1^{(t)}$.
  Sample $\boldsymbol{b}^{(t)} \sim \mathrm{Ber}(\varepsilon_{\mathsf{exp}})$.
  if $\boldsymbol{b}^{(t)} = 1$ **then**
    Sample $\boldsymbol{h}^{(t)} \sim \mathtt{unif}([H])$ and $\boldsymbol{\pi}^{(t)} \sim \mathtt{unif}(\Pi_{\mathsf{Base}})$.
    For $(h, \pi) = (\boldsymbol{h}^{(t)}, \boldsymbol{\pi}^{(t)})$, sample $(\boldsymbol{x}_1^{(t)}, \boldsymbol{a}_1^{(t)}, \boldsymbol{r}_1^{(t)}, \ldots, \boldsymbol{x}_H^{(t)}, \boldsymbol{a}_H^{(t)}, \boldsymbol{r}_H^{(t)}) \sim \mathbb{P}^{\widehat{\pi}^{(t)} \circ_h \pi}[\cdot \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}]$.
    For $(h, \pi) \in [H] \times \Pi_{\mathsf{Base}}$, define $\hat{\boldsymbol{\ell}}_{h,\pi}^{(t)} : \mathcal{V} \to \mathbb{R}$ such that

$$\hat{\boldsymbol{\ell}}_{h,\pi}^{(t)}(V) \leftarrow \mathbb{I}\{h = \boldsymbol{h}^{(t)}, \pi = \boldsymbol{\pi}^{(t)}\} \cdot \left(V(\boldsymbol{x}_h^{(t)}) - \sum_{\ell=h}^{H} \boldsymbol{r}_\ell^{(t)} \right)^2 . \tag{2}$$

  else
    Execute $\widehat{\pi}^{(t)}$ and observe $(\boldsymbol{x}_1^{(t)}, \boldsymbol{a}_1^{(t)}, \boldsymbol{r}_1^{(t)}, \ldots, \boldsymbol{x}_H^{(t)}, \boldsymbol{a}_H^{(t)}, \boldsymbol{r}_H^{(t)}) \sim \mathbb{P}^{\widehat{\pi}^{(t)}}[\cdot \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}]$.
    Set $\hat{\boldsymbol{\ell}}_{h,\pi}^{(t)} \equiv 0$, for all $(h, \pi) \in [H] \times \Pi_{\mathsf{Base}}$.
  for $h = 1, \ldots, H$ and $\pi \in \Pi_{\mathsf{Base}}$ **do**
    Get $\widehat{V}_{h,\pi}^{(t+1)} \in \mathcal{V}$ from online learning oracle $\mathcal{O}(\hat{\boldsymbol{\ell}}_{h,\pi}^{(1)}, \ldots, \hat{\boldsymbol{\ell}}_{h,\pi}^{(t)}; \mathcal{V})$.
  For $x \in \mathcal{X}$ and $h \in [H]$, define $\widehat{\pi}_h^{(t+1)}(\cdot \mid x) = \widehat{\pi}_{h,x}(\cdot \mid x)$, where $\widehat{\pi}_{h,x} \in \arg\max_{\pi \in \Pi_{\mathsf{Base}}} \widehat{V}_{h,\pi}^{(t+1)}(x)$.

5 Theoretical Results

Theorem 5.1 (AMI No-Regret)**.** *Let $\varepsilon_{\mathsf{exp}}$ be an exploration probability parameter and β correspond to the approximate-max following policy class parameter. For any sequence of initial states $\{\boldsymbol{x}_1^{(t)}\}_{t=1}^T$ chosen by the adversary, Algorithm 1 obtains regret bounded by:*

$$Reg_T = \sum_{t=1}^{T} \left(\inf_{\pi \in \Pi_{\mathsf{Bench}}(\beta)} V^{\pi}(\boldsymbol{x}_1^{(t)}) - V^{\widehat{\pi}^{(t)}}(\boldsymbol{x}_1^{(t)}) \right) \leq HT\varepsilon_{\mathsf{exp}} + \frac{1}{\eta} \frac{H^5 K^2}{\beta^2 \varepsilon_{\mathsf{exp}}} \log |\mathcal{V}|$$

Tuning $\varepsilon_{\mathsf{exp}}$ gives us regret that is $O(\sqrt{T}/\beta)$ or more specifically $O(\frac{H^3 K \sqrt{T \log |\mathcal{V}|}}{\beta})$.

Proof. See the proof in Appendix A. □

Recall once again that Hussing et al. [2024] in Lemma 4.1 show that the worst-approximate max-following policy is ϵ-competitive with the base class of policies.

5.1 Lower bound

We now present a lower bound of the order $\tilde{\Omega}(1/\beta^2)$. We note that there is a gap between the regret upper bound that we present of $\tilde{O}(\sqrt{T}/\beta)$ and this lower bound. The intuition behind the lower bound is that there exists a hard instance that requires learning the value function of each of the K baseline policies to reach a state with high reward. In particular, we create an instance where $K-1$ policies have value functions with expected rewards in $\{\frac{1}{2} + \beta, \ldots \frac{1}{2} + K\beta\}$ and one policy has value $\frac{3}{4} \gg \frac{1}{2} + K\beta$ in a state in the last layer of the MDP which is only reachable by playing the correct sequence of the $K-1$ policies in each layer of the MDP. Information theoretically, we need to play each of the $K-1$ policies for $\tilde{\Omega}(1/\beta^2)$ episodes before it is possible to distinguish between them and thus determine the correct order of play. The above intuition is realized by a tree-structured MDP with H layers where each non-leaf state has K children. Only leaf nodes have nonzero rewards and to reach the leaf with expected reward $\frac{3}{4}$ the player needs to distinguish policy π_k from π_{k+1} at layer k. An MDP instance can be found in Figure 1.

Theorem 5.2. *Any algorithm competing against any policy* $\Pi_{\text{Bench}}(\beta)$ *has regret of at least* $\Omega(\frac{K\log(K)}{\beta^2})$ *for* $\beta \le \frac{1}{8H} \wedge \frac{1}{10}$, $K \le H$.

The proof provided in Appendix B.3 together with further discussion on the tightness of the lower bound.

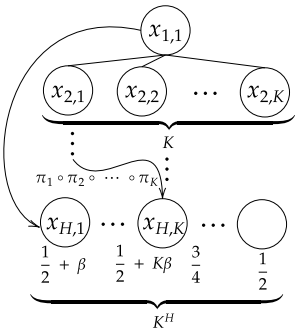


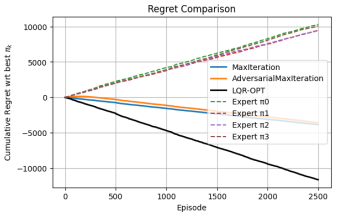
Figure 1: Hard instance

6 Experiments

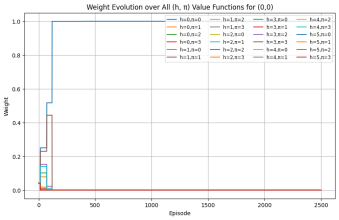
To test our algorithm in a continuous state-space setting, we empirically test our algorithm for the case of the Discrete Time Finite Horizon Linear Quadratic Regulator (LQR, Recht [2019]). Our base policy class consists of policies that only act along one dimension of the state space. In our example, our adversary is simulating playing distributions according to a specified normal distribution D. By using our method in the stochastic setting, we are able to compare this method with that of the MaxIteration algorithm presented in Hussing et al. [2024].

$$\min_{\{u_h\}_{h=0}^{H-1}} \mathbb{E}[x_H^T Q x_H + \sum_{h=0}^{H-1} (x_h^T Q x_h + u_h^T R u_h)]$$
$$\text{subject to } x_{h+1} = Ax_h + Bu_h, \quad x_0 \sim D$$

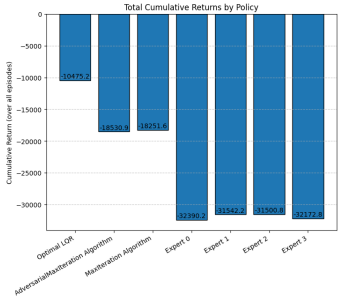
Note that for controllers specified of the form $u = -Kx$, the objective functions for the corresponding policy are quadratic functions of the state. For our experiments, we run with $d = 4, K = d, H = 5, T = 20000, \epsilon = 0.1, \eta = 0.01$. We also use $A = \begin{bmatrix} 0.9 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.9 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.9 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.9 \end{bmatrix}$, $B = Q = \mathbb{I}(d)$, and $R = 0.1 * \mathbb{I}(d)$. Also we denote $\pi_i(x) = -K_i x$, where $K_i[i,i] = 0.5$, and 0 everywhere else. Rewards will correspond to negative costs, so $r(x,u) = -(x^T Q x + u^T R u)$. For this synthetic experiment, we are able to precisely construct $\mathcal{V}$ to contain the value function of each base policy for every time step. Of course, in larger scale applications this would be replaced by a broader value function class which would require us making broader learnability assumptions about our value functions and more generally parametrizing our value function class, thereby increasing its cardinality. For the following experiments, we consider the following quantities. Over the course of states sampled across episodes, we measure the cumulative regret of the optimal LQR policy, the AdversarialMaxIteration policy, the MaxIteration policy, and any fixed base class policies against the best of the base class of policies for each state provided at the start of the episode. See Appendix C for more details on the experiment.



(a) Cumulative Regret of Algorithm vs. Base Experts

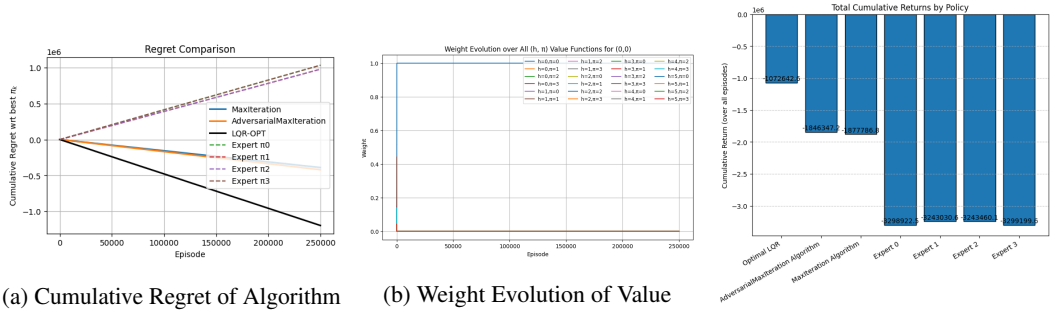


(b) Weight Evolution of Value Function for $h = 0, \pi_0$



(c) Bar Chart of Cumulative Rewards

Figure 2: Linear Quadratic Regulator Experiments for $\beta = \frac{1}{5}$

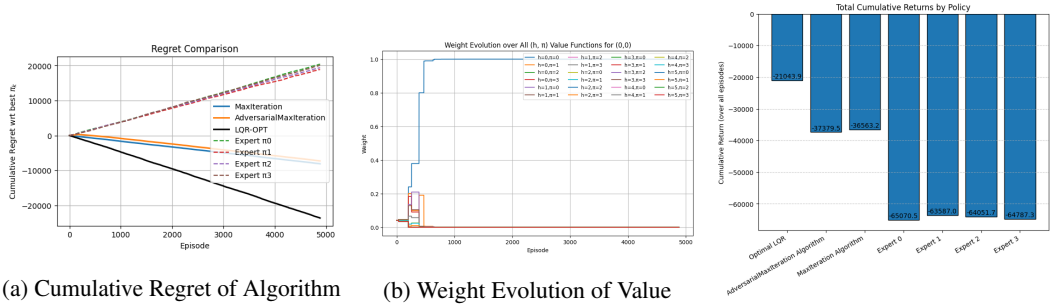


(a) Cumulative Regret of Algorithm vs. Base Experts

(b) Weight Evolution of Value Function for $h = 0, \pi_0$

(c) Bar Chart of Cumulative Rewards

Figure 3: Linear Quadratic Regulator Experiments for $\beta = \frac{1}{5}$ with 100 times longer T than Figure 2



(a) Cumulative Regret of Algorithm vs. Base Experts

(b) Weight Evolution of Value Function for $h = 0, \pi_0$

(c) Bar Chart of Cumulative Rewards

Figure 4: Linear Quadratic Regulator Experiments for $\beta = \frac{4}{25}$

We can see from Figure 2 that the cumulative regret of the algorithm is larger than the cumulative regret of the optimal LQR policy (which max-following is unable to compete with here) and slightly higher than that of the MaxIteration algorithm, but lower than the base class of policies, which seem to scale linearly in T. We can also see that the weight evolution of the value function corresponding to $h = 0, \pi_0$ increases significantly over time, which indicates that the correct value function is being learned for that given (h, π) pair over time. Finally, we compare the total cumulative rewards of the different policies collected over all episodes. One can broadly see through Figures 2,3,4,and 5 that MaxIteration and AdversarialMaxIteration are broadly comparable in performance, but the relative difference in which algorithm outperforms the other is affected by the length of the number of episodes AdversarialMaxIteration is run for (with shorter lengths making MaxIteration somewhat stronger and longer lengths making AdversarialMaxIteration stronger).

7 Conclusion and Future Work

Ultimately, we provide an algorithm **AMI** (1) against adversarial start states with access to a base class of policies, which is oracle-efficient and provably achieves no-regret with respect to the worst approximate-max-following policy and therefore the base class of policies itself. We provide a lower bound on β and experimentally validate our results. In future work, we would hope to remove the enumeration based approach to our current online-learning assumption and instead focus on broader value function classes. Moreover, it would be interested in considering other active/adaptive exploration-based algorithms. Finally, if we restricted ourselves to the setting of Smoothed Online Learning Block et al. [2022], it would be interesting to further explore implementations of no-regret learning algorithms like Follow the Perturbed Leader (FTPL), which make use of a regression oracle.

Acknowledgments and Disclosure of Funding

Omitted for blind review.

References

André Barreto, Shaobo Hou, Diana Borsa, David Silver, and Doina Precup. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences*, 117 (48):30079–30087, 2020.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory*, pages 1716–1786. PMLR, 2022.

Nataly Brukhim, Elad Hazan, and Karan Singh. A boosting approach to reinforcement learning. *Advances in Neural Information Processing Systems*, 35:33806–33817, 2022.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Ching-An Cheng, Andrey Kolobov, and Alekh Agarwal. Policy improvement from multiple experts. *arXiv preprint arXiv:2007.00795*, 2020.

Simon S Du, Sham M Kakade, Jason D Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in RL. *International Conference on Machine Learning*, 2021.

Marcel Hussing, Michael Kearns, Aaron Roth, Sikata B Sengupta, and Jessica Sorrell. Oracle-efficient reinforcement learning for max value ensembles. *Advances in Neural Information Processing Systems*, 37:117657–117681, 2024.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pages 1704–1713, 2017.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. *Neural Information Processing Systems*, 2021.

Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gap in reinforcement learning. In *Conference on Learning Theory*, pages 1282–1302. PMLR, 2022.

Alexander Korotin, Vladimir V'yugin, and Evgeny Burnaev. Mixability of integral losses: A key to efficient online aggregation of functional and probabilistic forecasts. *Pattern Recognition*, 120: 108175, 2021.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pages 6131–6141. PMLR, 2021.

Haolin Liu, Zak Mhammedi, Chen-Yu Wei, and Julian Zimmert. Beating adversarial low-rank mdps with unknown transition and bandit feedback. *Advances in Neural Information Processing Systems*, 37:134645–134700, 2024.

Xuefeng Liu, Takuma Yoneda, Chaoqi Wang, Matthew Walter, and Yuxin Chen. Active policy improvement from multiple black-box oracles. In *International Conference on Machine Learning*, pages 22320–22337. PMLR, 2023.

Teodor V Marinov, Alekh Agarwal, and Mircea Trofin. Offline imitation learning from multiple baselines with applications to compiler optimization. *arXiv preprint arXiv:2403.19462*, 2024.

Zakaria Mhammedi, Dylan J Foster, and Alexander Rakhlin. The power of resets in online reinforcement learning. *arXiv preprint arXiv:2404.15417*, 2024.

Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):253–279, 2019.

Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Contextual bandits and imitation learning with preference-based active queries. *Advances in Neural Information Processing Systems*, 36:11261–11295, 2023.

Vladimir Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, 1990.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

A Upper Bound Proof

Theorem A.1 (AMI No-Regret)**.** *Let $\varepsilon_{\mathsf{exp}}$ be an exploration probability parameter and β correspond to the approximate-max following policy class parameter. For any sequence of initial states $\{\boldsymbol{x}_1^{(t)}\}_{t=1}^T$ chosen by the adversary, Algorithm 1 obtains regret bounded by:*

$$Reg_T = \sum_{t=1}^{T}\left(\inf_{\pi\in\Pi_{\mathsf{Bench}}(\beta)} V^{\pi}(\boldsymbol{x}_1^{(t)}) - V^{\tilde{\pi}^{(t)}}(\boldsymbol{x}_1^{(t)})\right) \le HT\varepsilon_{\mathsf{exp}} + \frac{1}{\eta}\frac{H^5K^2}{\beta^2\varepsilon_{\mathsf{exp}}}\log|\mathcal{V}|$$

Tuning $\varepsilon_{\mathsf{exp}}$ gives us regret that is $O(\sqrt{T}/\beta)$ or more specifically $O(\frac{H^3K\sqrt{T\log|\mathcal{V}|}}{\beta})$.

Proof. Now, for $(h,\pi)\in[H]\times\Pi_{\mathsf{Base}}$, let's define $\ell_{h,\pi}^{(t)}:\mathcal{V}\to\mathbb{R}$ as

$$\ell_{h,\pi}^{(t)}(V) \coloneqq \mathbb{E}^{\widehat{\pi}^{(t)}\circ_h\pi}\left[\left(V(\boldsymbol{x}_h) - \sum_{\ell=h}^{H}\boldsymbol{r}_\ell\right)^2 \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}\right],$$

and note for the random losses defined $\hat{\boldsymbol{\ell}}_{h,\pi}^{(t)}$ in Algorithm 1 (see (2) and Line 10), we have for all $(h,\pi)\in[H]\times\Pi_{\mathsf{Base}}$ and $V\in\mathcal{V}$:

$$\mathbb{E}[\hat{\boldsymbol{\ell}}_{h,\pi}^{(t)}(V) \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}, \mathcal{G}_{t-1}] = \frac{\varepsilon_{\mathsf{exp}}}{KH}\cdot\ell_{h,\pi}^{(t)}(V), \tag{3}$$

where $\mathcal{G}_{t-1}$ is the σ-algebra induced by all the random variables in Algorithm 1 in iterations 1 to $t-1$. Thus, using (3) (with $V=\widehat{V}_{h,\pi}^{(t)}$), instantiating the online regret bound in (1) with $\hat{\ell}_{h,\pi}^{(t)} = \hat{\boldsymbol{\ell}}_{h,\pi}^{(t)}$, and applying Freedman's inequality, we get that with high probability

$$\begin{aligned}\frac{H^4K^2\log|\mathcal{V}|}{\eta\varepsilon_{\mathsf{exp}}} &\ge \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{\pi\in\Pi_{\mathsf{Base}}}\mathbb{E}^{\widehat{\pi}^{(t)}\circ_h\pi}\left[\left(\widehat{V}_{h,\pi}^{(t)}(\boldsymbol{x}_h) - \sum_{\ell=h}^{H}\boldsymbol{r}_\ell\right)^2 - \left(\sum_{\ell=h}^{H}\boldsymbol{r}_\ell - V_h^{\pi}(\boldsymbol{x}_h)\right)^2 \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}\right],\\ &= \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{\pi\in\Pi_{\mathsf{Base}}}\mathbb{E}^{\widehat{\pi}^{(t)}}\left[\left(\widehat{V}_{h,\pi}^{(t)}(\boldsymbol{x}_h) - V_h^{\pi}(\boldsymbol{x}_h)\right)^2 \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}\right].\end{aligned} \tag{4}$$

where $\varepsilon_{\mathsf{exp}}$ is the probability of exploration as in Algorithm 1.

The performance difference lemma. Fix $\beta>0$ and define $\pi_{h,\star}^{(t)}$ to be the policy satisfying for all $x\in\mathcal{X}$:

$$\pi_{h,\star}^{(t)}(\cdot\mid x) \coloneqq \pi_{h,x}^{(t)}(\cdot\mid x),$$

where

$$\pi_{h,x}^{(t)} = \begin{cases}\arg\max_{\pi\in\Pi_{\mathsf{Base}}}\widehat{V}_{h,\pi}^{(t)}(x), & \text{if } \max_{\pi\in\Pi_{\mathsf{Base}}}\left|\widehat{V}_{h,\pi}^{(t)}(x) - V_h^{\pi}(x)\right| \le \beta;\\ \pi_{h,\star}(\cdot\mid x), & \text{otherwise,}\end{cases}$$

where $\pi_\star$ is the max-following policy. We note that $\pi_{h,\star}^{(t)}$ is in $\Pi_{\mathsf{Bench}}(\beta)$ (this is the same benchmark policy class as in the max-following paper).

Let $\tilde{\pi}^{(t)} = (1-\varepsilon_{\mathsf{exp}})\cdot\widehat{\pi}^{(t)} + \frac{\varepsilon_{\mathsf{exp}}}{H}\sum_{h=1}^{H}\widehat{\pi}^{(t)}\circ_h\boldsymbol{\pi}^{(t)}$ denote the policy mixture executed at iteration t of Algorithm 1. By the performance difference lemma, we have that for all $t\in[T]$:

$$\begin{aligned}V^{\pi_{h,\star}^{(t)}}(\boldsymbol{x}_1^{(t)}) - V^{\tilde{\pi}^{(t)}}(\boldsymbol{x}_1^{(t)}) &\le \left(V^{\pi_{h,\star}^{(t)}}(\boldsymbol{x}_1^{(t)}) - V^{\widehat{\pi}^{(t)}}(\boldsymbol{x}_1^{(t)})\right) + H\varepsilon_{\mathsf{exp}},\\ &\le H\varepsilon_{\mathsf{exp}} + \sum_{h=1}^{H}\mathbb{E}^{\widehat{\pi}^{(t)}}\left[Q_h^{\pi_\star^{(t)}}(\boldsymbol{x}_h,\pi_{h,\star}^{(t)}(\boldsymbol{x}_h)) - Q_h^{\pi_\star^{(t)}}(\boldsymbol{x}_h,\widehat{\pi}_h^{(t)}(\boldsymbol{x}_h)) \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}\right],\\ &\le H\varepsilon_{\mathsf{exp}} + H\sum_{h=1}^{H}\mathbb{P}^{\widehat{\pi}^{(t)}}\left[\max_{\pi\in\Pi_{\mathsf{Base}}}\left|\widehat{V}_{h,\pi}^{(t)}(\boldsymbol{x}_h) - V_h^{\pi}(\boldsymbol{x}_h)\right| > \beta \mid \boldsymbol{x}_1 = \boldsymbol{x}_1^{(t)}\right],\end{aligned}$$

$$\leq H\varepsilon_{\mathsf{exp}} + \frac{H}{\beta^2}\sum_{h=1}^{H}\sum_{\pi\in\Pi_{\mathsf{Base}}}\mathbb{E}^{\bar{\pi}^{(t)}}\left[\left(\widehat{V}^{(t)}_{h,\pi}(\boldsymbol{x}_h) - V^{\pi}_h(\boldsymbol{x}_h)\right)^2 \mid \boldsymbol{x}_1 = \boldsymbol{x}^{(t)}_1\right].$$

Summing this over $t = 1, \ldots, T$, and using (4), we get

$$\sum_{t=1}^{T}\left(\inf_{\pi\in\Pi_{\mathsf{Bench}}(\beta)} V^{\pi}(\boldsymbol{x}^{(t)}_1) - V^{\tilde{\pi}^{(t)}}(\boldsymbol{x}^{(t)}_1)\right) \leq \sum_{t=1}^{T}\left(V^{\pi^{(t)}_{h,\star}}(\boldsymbol{x}^{(t)}_1) - V^{\tilde{\pi}^{(t)}}(\boldsymbol{x}^{(t)}_1)\right) \leq HT\varepsilon_{\mathsf{exp}} + \frac{1}{\eta}\frac{H^3K^2}{\beta^2\varepsilon_{\mathsf{exp}}}H^2\log|\mathcal{V}|.$$

Tuning $\varepsilon_{\mathsf{exp}} \propto \beta^{-1}T^{-1/2}$, gives us a $O(T^{1/2}/\beta)$ regret bound. □

B Lower Bound Proofs

B.1 The tree construction

Consider the following family of MDPs with H layers. Each MDP is a tree where each node of the tree has K children. The transition from each state $x^{(t)}_{h,i}$ to its children $x^{(t)}_{h+1,j}$ is deterministic and each of the K policies in Π_{Base} plays an action that transitions to a different state. We are going to assume that the $K+1$-st baseline policy always transitions to an absorbing state for all layers $h < H$ with 0 reward. We now define a single MDP instance from the family of instances. We always assume that at each $x^{(t)}_{h,i}$ it holds that for all $i, j \in [K]$ with $i \neq j$ $\pi_i(x^{(t)}_{h,s}) \neq \pi_j(x^{(t)}_{h,s}), \forall h, s$. Further, as stated above, we have that if $\mathbb{P}(x^{(t)}_{h+i,s'}|\pi_i(x^{(t)}_{h,s}), x^{(t)}_{h,s}) = 1$ then $\mathbb{P}(x^{(t)}_{h+i,s'}|\pi_j(x^{(t)}_{h,s}), x^{(t)}_{h,s}) = 0$ and further there exists a single $x^{(t)}_{h+i,s'}$ for each $\pi_i, x^{(t)}_{h,s}$ pair for which it holds $\mathbb{P}(x^{(t)}_{h+i,s'}|\pi_i(x^{(t)}_{h,s}), x^{(t)}_{h,s}) = 1$.

We define an instance of the class of MDPs as follows. By definition of the MDP class each π_i follows a single trajectory throughout the tree. Let $\rho^{(t)} : [K] \times \mathcal{X} \to K^H$ be the function which maps policy π_i to the index of the leaf that is the end of the trajectory for π_i starting from state x. That is $x_{H,\rho(\pi_i,x)}$ is the state to which π_i transitions after starting at x. The instance construction begins by sampling a policy $\pi^{(t)}_1$ is chosen uniformly at random from the $[K]$ baseline policies. The reward of $r(x^{(t)}_{H,\rho(\pi^{(t)}_1,x^{(t)}_{1,1})}, \pi^{(t)}_1(x^{(t)}_{H,\rho(\pi^{(t)}_1,x^{(t)}_{1,1})}))$ is then sampled from a Bernoulli r.v. with mean $\frac{1}{2} + \beta$ and the remaining rewards on the trajectory of $\pi^{(t)}_1$ are set to 0 so that $V^{\pi^{(t)}_1}(x^{(t)}_{1,1}) = \frac{1}{2} + \beta$. WLOG assume that $\pi^{(t)}_1$ transitions to state $x^{(t)}_{2,1}$ at the second layer. Sample a policy $\pi^{(t)}_2$ uniformly at random from $[K] \setminus \{\pi^{(t)}_1\}$ and sample the reward of $r(x^{(t)}_{H,\rho(\pi^{(t)}_2,x^{(t)}_{2,1})}, \pi^{(t)}_2(x^{(t)}_{H,\rho(\pi^{(t)}_2,x^{(t)}_{2,1})}))$ from a Bernoulli r.v. with mean $\frac{1}{2} + 2\beta$, the remaining rewards on the trajectory are set to 0 so that $V^{\pi^{(t)}_2}(x^{(t)}_{2,1}) = \frac{1}{2} + 2\beta$. The construction continues in a similar way where at layer h we have $V^{\pi^{(t)}_h}(x^{(t)}_{h,1}) = \frac{1}{2} + h\beta$ for $\pi^{(t)}_h$ sampled uniformly at random from $[K] \setminus \{\pi^{(t)}_{h-1}\}$. Note that this construction allows for $\pi^{(t)}_1 = \pi^{(t)}_h, h \neq 2$. Finally, at state $x^{(t)}_{H,1}$ we set the reward of π_{K+1} to be $\mathrm{Ber}(3/4)$, that is $r(\pi^{(t)}_H(x^{(t)}_{H,1}), x^{(t)}_{H,1}) \sim \mathrm{Ber}(\frac{3}{4})$. In summary, with the indexing that we have adopted above

$$r(x^{(t)}_{H,\rho(\pi^{(t)}_h,x^{(t)}_{h,1})}, \pi^{(t)}_h(x^{(t)}_{H,\rho(\pi^{(t)}_h,x^{(t)}_{h,1})})) \sim \mathrm{Ber}(\frac{1}{2} + h\beta)$$
$$r(\pi^{(t)}_H(x^{(t)}_{H,1}), x^{(t)}_{H,1}) \sim \mathrm{Ber}(\frac{3}{4}).$$

The rewards of all other leaves are sampled from $\mathrm{Ber}(\frac{1}{2})$, that is, $r(a, x_{H,i}) \sim \mathrm{Ber}(\frac{1}{2})$ for all other actions a. Finally, all other rewards are set to 0.

To receive the maximum reward, a max-following policy has to learn to distinguish the value function of $\pi^{(t)}_h$ at $x^{(t)}_{h,1}$ at all $h \in H$ from that of $\pi^{(t)}_{h-1}$ which roughly requires $\tilde{\Theta}(1/\beta^2)$ samples. Furthermore, depending unless the player observes a reward from one of the $\mathrm{Ber}(\frac{1}{2} + h\beta)$ distributions or a reward of 1 from playing π_{K+1} at $x^{(t)}_{H,1}$, the player receives no information to distinguish from an MDP with the same transition dynamics but with all leaf rewards sampled from $\mathrm{Ber}(\frac{1}{2})$. Observing a reward

from $\mathrm{Ber}(\frac{1}{2}+h\beta)$, roughly reveals information about the value of all policies $\pi_1^{(t)},\ldots,\pi_h^{(t)}$, however, unless the player has already learned the value functions of $\pi_1^{(t)},\ldots,\pi_{h-j}^{(t)}$ for some $j=O(1)$, then it is highly unlikely that the player can play according to policies $\pi_{h-j}^{(t)},\ldots,\pi_h^{(t)}$. In particular, because the player has no information about the value functions, they can not distinguish from the MDP with uniform rewards and the best they can do is play uniformly at random for j layers. This in turn implies that they only guess the correct path up to $\pi_h^{(t)}$ w.p. at most $\left(\frac{1}{K-1}\right)^j$.

To make the above formal we proceed with a standard information-theoretic argument. Let $\mathcal{E}$ denote the environment of the sampled MDP above and let $\mathcal{E}'$ be the environment with all leaf rewards sampled according to $\mathrm{Ber}(\frac{1}{2})$. Let $\mathbb{P}_{\mathcal{E}}$ and $\mathbb{P}_{\mathcal{E}'}$ be the induced measures of composing the environment with the agent's algorithm for T rounds.

B.2 KL Upper Bound

Lemma B.1. *For any game with T rounds and agent's strategy it holds that*

$$\mathrm{KL}(\mathbb{P}_{\mathcal{E}'}\|\mathbb{P}_{\mathcal{E}})\le\frac{10T\beta^2}{(K-1)\log(K-1)}.$$

Proof. Let $\pi_{h,t}$ be the base policy selected by algorithm at layer h of the MDP during the t-th round of the game. We have

$$\begin{aligned}&\mathbb{P}_{\mathcal{E}}\left(r_{H,t},\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)},r_{H,t-1},\pi_{H,t-1},x_H^{(t-1)},\ldots,\pi_{1,1},x_1^{(1)}\right)\\ =&\Pi_{s=1}^t\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)}\right)\mathbb{P}_{\mathcal{E}}\left(\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)}|r_{H,t-1},\pi_{H,t-1},x_H^{(t-1)},\ldots,\pi_{1,1},x_1^{(1)}\right).\end{aligned}$$

We unpack $\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)}\right)$ which is a Bernoulli r.v. First, because transitions are deterministic we have $\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)}\right)=\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},\pi_{H-1,t},\ldots,\pi_{1,t},x_1^{(t)}\right)$. Next, let $x_{h,t}$ be the first layer at which it holds that $\pi_{h,t}\neq\pi_h^{(t)}$, that is the agent does not play the max-value policy. If it also holds that $\pi_{h,t}\neq\pi_{h-1}^{(t)}$ we have $\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},\pi_{H-1,t},\ldots,\pi_{1,t},x_1^{(t)}\right)=\mathrm{Ber}(\frac{1}{2})$. If at all remaining states in the trajectory it holds that $\pi_{h',t}=\pi_{h-1}^{(t)},h'\ge h-1$, then $\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},\pi_{H-1,t},\ldots,\pi_{1,t},x_1^{(t)}\right)=\mathrm{Ber}(\frac{1}{2}+(h-1)\beta)$, otherwise again it holds that $\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},\pi_{H-1,t},\ldots,\pi_{1,t},x_1^{(t)}\right)=\mathrm{Ber}(\frac{1}{2})$. Finally, if the agent plays $\pi_h^{(t)},h\le H-1$ and π_{K+1} at $x_H^{(t)}$ then the reward is just equal to 1. We denote the event $\pi_{h',t}=\pi_{h'}^{(t)},h'\le h,\pi_{h',t}=\pi_h^{(t)},h'>h$ by $\mathcal{A}_h^{(t)}$ and let $\mathcal{A}_0^{(t)}=\left(\bigcup_{h=1}^H\mathcal{A}_h^{(t)}\right)^{\mathsf{C}}$, $\mathcal{A}_{H+1}^{(t)}=\{\pi_{h,t}=\pi_h^{(t)},h\le H-1,\pi_{H,t}=\pi_{K+1}\}$. The above decomposition of $\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)}\right)$ can be summarized as

$$\mathbb{P}_{\mathcal{E}}\left(r_{H,t}|\pi_{H,t},x_H^{(t)},\ldots,\pi_{1,t},x_1^{(t)}\right)=\sum_{h=0}^H\mathbb{I}(\mathcal{A}_h^{(t)})\mathrm{Ber}\left(\frac{1}{2}+h\beta\right)+\mathbb{I}(\mathcal{A}_{H+1}^{(t)})\mathrm{Ber}\left(\frac{3}{4}\right).\tag{5}$$

Further, we have the following identity under $\mathcal{E}'$, since $\pi_h^{(t)}$ is chosen uniformly at random from $K-1$ policies at every layer, independently of previous rounds

$$\mathbb{P}_{\mathcal{E}'}(\mathcal{A}_h^{(t)}|r_{H,t-1},\pi_{H,t-1},x_H^{(t-1)},\ldots,\pi_{1,1},x_1^{(1)})\le\left(\frac{1}{K-1}\right)^h.\tag{6}$$

Equation 5 and Equation 6 allow us to bound the KL-divergence between $\mathbb{P}_{\mathcal{E}}$ and $\mathbb{P}_{\mathcal{E}'}$ as

$$\begin{aligned}\mathrm{KL}(\mathbb{P}_{\mathcal{E}'}\|\mathbb{P}_{\mathcal{E}})\le&\sum_{t=1}^T\mathbb{E}_{\mathcal{E}'}\Bigg[\sum_{h=0}^H\mathbb{I}(\mathcal{A}_h^{(t)})\mathrm{KL}\left(\mathrm{Ber}\left(\frac{1}{2}\right)\|\mathrm{Ber}\left(\frac{1}{2}+h\beta\right)\right)\\&+\mathbb{I}(\mathcal{A}_{H+1}^{(t)})\mathrm{KL}\left(\mathrm{Ber}\left(\frac{1}{2}\right)\|\mathrm{Ber}\left(\frac{3}{4}\right)\right)\Bigg]\\\le&\sum_{t=1}^T\sum_{h=1}^H\frac{3h^2\beta^2}{(K-1)^h}+\frac{1}{(K-1)^H}\le\sum_{t=1}^T\frac{10\beta^2}{(K-1)\log(K-1)}.\end{aligned}$$

□

B.3 β Lower Bound

Theorem B.1. *Any algorithm competing against any policy $\Pi_{\mathsf{Bench}}(\beta)$ has expected minimum regret of at least $\Omega(\frac{K\log(K)}{\beta^2})$ for $\beta \le \frac{1}{8H} \wedge \frac{1}{10}$.*

Proof. Under the event $(\mathcal{A}_{H+1}^{(t)})^{\mathsf{C}}$ the agent incurs regret at least $\frac{1}{4} - H\beta$. We consider the following modified game in which the agent incurs 0 regret for any round $T' \ge \frac{(K-1)\log(K-1)}{20\beta^2}$. The regret of the agent in this game is no larger compared to the original game for any strategy of the player. Pinsker's inequality implies that

$$\mathbb{P}_{\mathcal{E}}(\mathcal{A}_{H+1}^{(t)}) \le \mathbb{P}_{\mathcal{E}'}(\mathcal{A}_{H+1}^{(t)}) + \sqrt{\frac{5T'\beta^2}{(K-1)\log(K-1)}} \le \frac{1}{(K-1)^H} + \frac{1}{2} \le \frac{3}{4}.$$

And so for the first T' rounds the expected regret incurred by the agent strategy is at least

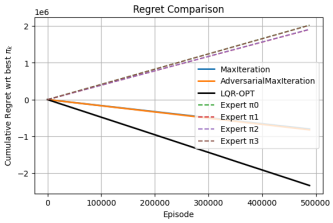
$$\sum_{t=1}^{T'} \frac{\mathbb{E}_{\mathcal{E}'}[\mathbb{I}\left((\mathcal{A}_{H+1}^{(t)})^{\mathsf{C}}\right)]}{8} \ge \frac{T'}{32} \ge \Omega\left(\frac{K\log(K)}{\beta^2}\right).$$

□

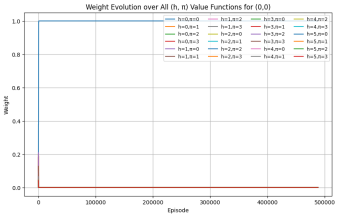
We note that the above lower bound can be tightened to $\Omega(HK\log(K)/\beta^2)$ by slightly extending the construction to replace each leaf of the MDP by length H paths with a single state and transition with a Bernoulli reward corresponding to the reward of the leaf in the current construction. This will increase the variance of the value functions from $O(1)$ to $\Omega(H)$ and decrease the KL-divergence in Lemma B.1 by H. Further, we note that the above construction does not really use the adversarial nature of the setting and applies to the stochastic setting as long as we allow MDPs with $\Omega(K^{H+1})$ states and in particular will apply to the setting of Hussing et al. [2024]. There is, however, still a gap between the sample complexity implied by the lower bound, i.e., $\tilde{\Omega}(K/\beta^2)$ and that achieved by the max-following algorithm in Hussing et al. [2024] which (ignoring all other terms) is $\tilde{O}(K/\beta^3)$.

C Experiments

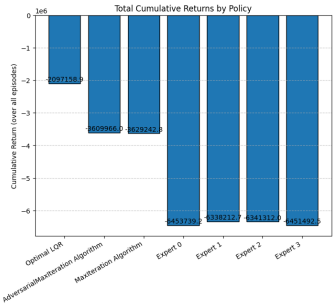
Notice that the MaxIteration algorithm learns an approximate-max-following policy using $O(HK)$ oracle calls, so during the evaluation over T episodes, we are utilizing the learned approximate max-following policy. In contrast, for AMI, the learner is updating its policy over episodes, which is tracked with index t. We track the weight evolution of the time step $h = 0$ and π_0 policy because we want to ensure over time that it learns $V_0^{\pi_0}$. Finally, we provide a chart plotting the cumulative returns across all episodes for all of the policies. This plot provides similar information to that of the cumulative regret plots, but we provide it for additional visual clarity on the relative performance of the policies. We will denote the error parameter ϵ in Hussing et al. [2024] as ξ to avoid confusion with our exploration parameter. ξ is the parameter governing the competitiveness of the policy outputted from MaxIteration to the approximate max-following policy class and base class of policies. That is, the regret is shown to be within ξ of the worst approximate max-following policy and within ξ of the best policy from the base class. α is the regression oracle accuracy governing how the tolerance of squared error we have from the oracle outputting estimated value-functions of the base class. Since MaxIteration is an oracle-efficient algorithm making $O(HK)$ oracle queries, their sample complexity translates into the number of samples needed per oracle call (which depends on the α chosen) multiplied by HK for each oracle call. Then, if we were to select all the optimized quantities for (β and α are set from Hussing et al. [2024]) our parameters we would have $\beta \in \Theta(\frac{\xi}{H})$ and $\alpha \in \Theta(\frac{\xi^3}{KH^4})$ and $N_{traj} \propto \frac{1}{\alpha}$ to simulate the regression oracle assumption in the algorithm MaxIteration. As one form of comparison, we might evaluate AMI for $T \propto HKN_{traj}$ based upon Hussing et al. [2024] and then set our $\epsilon_{exp} = \frac{H^2K\sqrt{\log(|\mathcal{V}|)}}{\beta\sqrt{\eta T}}$. Note that if we were to run our algorithm for significantly longer, i.e. if we choose to set our target average regret to be say $O(\xi)$, our algorithm may outperform MaxIteration (compare Figure 2 and Figure 3 or Figure 4 and Figure 5) , however our number of episodes T also would become large very quickly making it more time-consuming in practice. Therefore, we choose to compare performance over a range of values of β between AdversarialMaxIteration and MaxIteration and try to compare with a slightly longer number of episodes for that same β.



(a) Cumulative Regret of Algorithm vs. Base Experts



(b) Weight Evolution of Value Function for $h = 0, \pi_0$



(c) Bar Chart of Cumulative Rewards

Figure 5: Linear Quadratic Regulator Experiments for $\beta = \frac{4}{25}$ with 100 times longer T than Figure 4