Model Selection for Average Reward RL with Application to Utility Maximization in Repeated Games

Alireza Masoumian
University of Alberta / Amii

James R. Wright
University of Alberta / Amii

Abstract

In standard RL, the structure of the Markov Decision Process (e.g. state space) is known. In online model selection, a learner attempts to learn an optimal policy for an MDP knowing only that it belongs to one of $M > 1$ model classes of varying complexity. Recent results have shown that this can be feasibly accomplished in episodic online RL. In this work, we propose MRBEAR, an online model selection algorithm for the average reward RL setting which is based on the idea of regret balancing and elimination. The regret of the algorithm is in $\tilde{O}(MC_{m*}^2 B_{m*}(T, \delta))$ where C_{m*} represents the complexity of the simplest well-specified model class and $B_{m^*}(T, \delta)$ is its corresponding regret bound. This result shows that in average reward RL, the additional cost of model selection scales only linearly in M, the number of model classes.

As an application, in a simultaneous general-sum repeated game, where the opponent follows a fixed unknown limited memory strategy, the learner can maximize its utility using MRBEAR. By proving a lower bound, we showed the learner's regret is tight in opponent's memory order. In addition, the algorithm's performance is demonstrated through experiments.

1 INTRODUCTION

Most work in average reward reinforcement learning assumes that the underlying model class—i.e., state space and actions—is known. However, in practice the definition of the state space must be specified. This leads to a tradeoff between the richness of the model and the hardness of learning within it. Encoding a large part of observations in the states will blow up the set of states, making it very challenging to learn an optimal policy; conversely, a model class that encodes too little information into its states may not be rich enough to describe the underlying interaction.

Suppose the environment is an Atari game for which the algorithm tries to find the optimal strategy. The standard DQN algorithm (Mnih, 2013) uses the previous 4 frames of the game to define the current state (Figure 2). But an important question can be "What is the optimal number of previous frames that the current state should capture?" This is the question that we answer in online model selection. One can consider 10 different model classes vary on the number of previous frames they capture, i.e. in model class i ($1 \leq i \leq 10$), the current state is defined based on the last i frames. As the size of the state space grows exponentially in the number of frames, over-estimating the number of frames needed to find the optimal policy might cause a notable cost for the algorithm.

One way to resolve this tradeoff is by selecting the model class during learning based on interactions with the environment. In the episodic regret setting, it has been shown that online model selection using M different model classes (at least one of which is assumed to be well-specified) can attain a regret bound that is only a factor of M worse than that of learning using the simplest well-specified model class from the collection. Since the most general model class (the model class with 10 frames in the example) can have regret bounds that are superlinearly (often exponentially) greater than those of the optimal model class, this results in an improvement over the naive approach of learning in the largest model class that is guaranteed to be well-specified.
This asymptotic improvement is attained by using a meta-algorithm which selects one model class per time step, and runs one episode of a learning algorithm on the selected model class.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Main Question. The main question that we ask is whether we can achieve a similar regret bound for online model selection in the average reward setting; namely, one which is worse than the regret bound of the optimal model class only up to linear factor of M. We answer this question affirmatively, by designing an algorithm called MRBEAR. Using the regret balancing and elimination technique (Abbasi-Yadkori et al., 2020; Pacchiano et al., 2020), we keep track of the model classes that may be well specified in the set of active model classes, and update this set by checking a misspecification test over the model classes. We theoretically show that by using MRBEAR the cost of considering M different model classes, due to the lack of apriori knowledge of optimal complexity, only linearly grows in M.

The average reward setting presents challenges that are not present in the episodic setting. In online episodic RL, the regret can be decomposed into the sum of instantaneous regrets, each corresponding to an episode. However, there is no reset or episode in the average reward setting, which results in a stricter notion of regret (2), that can not be expressed as a sum of temporally decomposed instantaneous regrets. This means that the model selection algorithms that exploit this temporal decomposition property (Pacchiano et al., 2020; Ghosh et al., 2021) are not easily modifiable to be used in the average reward setting (refer to Appendix B.4 to compare different notions of regret).

In addition, the meta-algorithm can not change its chosen base algorithm, time-step by time-step, or very often. This is because in an RL setting, the actions have long-term (planning) effects. Thus, the meta algorithm needs to commit to a base algorithm for a successive bunch of iterations to capture the planning effect of the actions chosen by the base algorithms. Due to this commitment, we have to check the misspecification test less frequently. We will show that even under this less-frequent checking of our proposed misspecification test for the average reward setting, the set of active classes will contain all of the well-specified model classes with high probability, and more importantly, the average reward model selection regret can be controlled. In online episodic RL, the commitment to the base algorithms is not an issue, since we can change our base algorithm after each episode, and the objective is to have sublinear regret in the number of episodes (usually denoted by K) and not the horizon of each episode (H).

Application. We apply our model selection result to the repeated game setting against an opponent with limited memory, with the goal of maximizing the learner's cumulative utility. This goal, together with having no discounting or reset during the T interactions, leads us to use average reward regret to evaluate the learner's performance. The opponent can condition their choice in each game on the past m^* ($0 \leq m^* < M$) plays, where the upper limit M is known to the learner but the true limit (or memory) m^* is not. A naive approach would be to learn on the MDPs where each state contains the last M choices. The size of the state space in such MDPs is exponentially larger than the true induced MDP that captures only m^* past actions. Model selection allows us to learn the opponent's memory limit simultaneously with learning an optimal policy, by treating the state space induced by each potential memory limit as a separate model class. In average-reward regret, all of the rewards during the course of learning contribute without discounting and the planning effect of actions is captured. This makes this notion a more natural performance measure in this setting than episodic regret or PAC guarantees since the learner's goal is to maximize its utility and not minimizing regret. When both players are assumed to have unbounded memory and common knowledge of rationality (i.e., both are utility-maximizers, and know that the other is a utility-maximizer, and know that the other knows, etc.), an optimal strategy for each agent can be found using backward induction (Shoham and Leyton-Brown, 2008). However, in this work we relax both assumptions. Exploiting common knowledge of rationality requires both agents to know the others' utility; in practice, this will rarely be true. We do not assume the knowledge of the opponent's or even the learner's own utility functions. We further assume that the opponent's policy depends only on a fixed number of past actions taken in the game, but that this memory limit is not known to the learner. Thus, our problem can be understood as finding the best response, in the repeated game, to an opponent's fixed, finite-memory strategy. Our contribution in this work is as follows:

1. We propose MRBEAR, an online model selection meta-algorithm for the average reward setting (section 3) which is compatible with a wide range of base algorithms (definition 3.1). We propose a multiplicative misspecification test which is suitable for average reward RL, and using it we prove a regret guarantee for MRBEAR which is only linearly dependent on the number of model classes (section 4)

2. By making a connection between average reward reinforcement learning and utility maximization in repeated games, we apply our algorithm to obtain an instance-dependent regret bound against an opponent with an unknown limited memory of

m^* (section 5). We prove a minimax lower bound on the learner's regret, showing that the dependency on the opponent's memory in our regret bound is optimal (section 5.1).

3. Confirming the theoretical guarantees, we also empirically demonstrate the performance of MRBEAR (section 6).

1.1 Related Work

There is a rich literature on model selection for different settings (Vapnik, 2006; Lugosi and Nobel, 1999; Massart, 2007; Foster et al., 2017; Abbasi-Yadkori et al., 2020; Marinov and Zimmert, 2021; Krishnamurthy et al., 2024). The cost of model selection under different assumptions can be either multiplicative (Pacchiano et al., 2020) or additive (Ghosh et al., 2021) to the regret of the best well-specified model class. For instance, Ghosh et al. (2021) assume that they can obtain information from the model classes simultaneously, in addition to a separability assumption for the underlying model classes. For an analogy, if we think of the model classes as a set of arms, these assumptions are similar to the full-information setting and knowing a lower bound for the gap between the quality of the arms. Our setting is more similar to a bandit-information without assuming separability, making our result more general.

An extensive literature review is available in Appendix A to also cover other related research threads such as average reward reinforcement learning (Bartlett and Tewari, 2012; Boone and Zhang, 2024), learning in Markov games (Wei et al., 2017; Zhong et al., 2021; Jin et al., 2021), and learning in repeated games (Brown et al., 2024; Braverman et al., 2018; Assos et al., 2024).

2 PROBLEM SETTING

A Markov Decision Process (MDP) $\mathcal{M}$ is a tuple, $(\mathcal{S}, \mathcal{A}, P, r, T, \mu)$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces with the cardinality $|\mathcal{S}| = \boldsymbol{S}$ and $|\mathcal{A}| = \boldsymbol{A}$. The reward $R_t = r(s_t, a_t)$ is obtained by applying the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to the pair of state and action in time-step t. Given this pair, the transition kernel P determines the probability of the next state, with $\mathbb{P}(S_{t+1} = s_{t+1} | S_t = s_t, A_t = a_t) = P(s_{t+1}|s_t, a_t)$. Starting from the initial state $S_1 \sim \mu$, T rounds of interaction occur. A memoryless policy is a mapping $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ from state space to distributions over the action space.

Gain, Bias, and Value Function. For an arbitrary policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ and an initial state s, associated probability and expected operators are denoted by $\mathbf{P}_s^\pi$ and $\mathbf{E}_s^\pi$. Denote $R^\pi(s) := r(s, \pi(s))$ and $P^\pi(s, s') := P(s'|s, \pi(s))$ as the reward and transition kernel induced by policy π. The value function of a policy captures the cumulative reward obtained by following a policy in T rounds, i.e. $V_T^\pi(s) := \mathbf{E}_s^\pi \sum_{t=1}^T R_t$. The gain of the policy $g^\pi(s)$ shows the asymptotic average reward obtained by following policy π and starting from state s, i.e. $g^\pi(s) := \lim_{T\to\infty} \frac{1}{T} \mathbf{E}_s^\pi \sum_{t=1}^T R_t$. The transient part, i.e. $h^\pi(s) := \mathbf{E}_s^\pi[\lim_{T\to\infty} \sum_{t=1}^T (R_t - g^\pi(S_t))]$ measures the bias of starting from state s, thus $h^\pi(s) \in O(1)$ for all states $s \in \mathcal{S}$. The Poisson equation states $h^\pi + g^\pi = r^\pi + P^\pi h^\pi$. The optimal gain is defined as $g^*(s) := \sup_\pi g^\pi(s)$, and the maximizer policy is denoted by π^*, i.e. optimal policy. For a weakly communicating MDP, $g^* \in \{x.\mathbf{1} : x \in \mathbb{R}\}$ where $\mathbf{1}$ is the vector full of ones. It means that the value of the optimal gain does not depend on the initial state which allows us to also use g^* as a scalar when it is clear from context, i.e. $g^*(s) = g^*$ for all $s \in \mathcal{S}$. The Bellman operator L operates on a vector $h \in \mathbb{R}^{\boldsymbol{S}}$ as $Lh(s) := \max_{a\in\mathcal{A}}\{r(s, a) + P(.|s, a)h\}$. It is also known that there exists a solution h^* for the Bellman operator such that $g^* = Lh^* - h^*$, and they satisfy $g^* + h^* \geq r^\pi + P^\pi h^*$ for all $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$. It is not hard to verify that $V_T^\pi = Tg^\pi + h^\pi + \epsilon$ where $V_T^\pi, g^\pi, h^\pi, \epsilon \in \mathbb{R}^{\boldsymbol{S}}$ and $\epsilon \in o(1)$ entry-wise. The optimal value starting from the state s is $V_T^*(s) := \sup_{\pi\in\Pi_{\mathcal{A}}} V_T^\pi(s)$. Note that the solution policy for the previous equation may change for different initial states, unlike the gain-optimal policy which captures the asymptotic behavior. The complexity of an MDP can be measured in multiple ways.

Definition 2.1. *The* diameter *of an MDP $\mathcal{M}$ is* $D(\mathcal{M}) := \sup_{s\neq s'} \inf_\pi \mathbf{E}_s^\pi[\min\{t > 1 : S_t = s'\} - 1$. *The* span *of a vector $h \in \mathbb{R}^{\boldsymbol{S}}$ is* $\mathsf{sp}(h) := \max_s h(s) - \min_{s'} h(s')$.

Note that D is only dependent on the transition kernel of the MDP.

Definition 2.2. *We say that a MDP is* weakly communicating, *if there exists a closed set of states in which each state is accessible from every other using some deterministic policy, plus a possibly empty set of states which is transient under* every *policy.*

Refer to Appendix B.1 for a broader classification of MDPs. The average reward regret of an algorithm Alg is

$$\mathsf{R}(Alg, T) := Tg^* - \sum_{t=1}^T R_t. \tag{1}$$

Recall that $R_t = r(S_t, A_t)$. In regret (1) we compare the performance of the algorithm with the asymptotic optimal average reward. One can instead choose $V_T^*(s_1)$, as the baseline which defines the regret $\mathsf{R}_V(Alg, T) = V_T^*(s_1) - \sum_{t=1}^T R_t$. The difference between these two notions is $Tg^* - V_T^*(s)$ which is less

than $2\mathsf{sp}(h^*)$ (see Lemma C.5). As there is a minimax lower bound for the regret 1 in $\Omega(\sqrt{\mathsf{sp}(h^*)\boldsymbol{SAT}})$, this gap is negligible. The average reward regret captures utility maximization since it compares the learner's performance to the highest possible cumulative utility. This is not the case in external regret, swap regret, and even dynamic regret. In dynamic regret, which is the strongest among them, the baseline captures immediate best actions, without considering any planning effect of the actions. Refer to appendix B.4 for further detailed explanations.

2.1 Model Selection Problem

T rounds of interaction with the environment produce a stream of observations and rewards $(O_1, R_1), (O_2, R_2), \dots (O_T, R_T)$, where $(O_i, R_i) \in \mathcal{O} \times [0, 1]$. The observations can be encoded as states in different ways. For example, there might be two encoding functions $E_1, E_2 : O^* \to \mathcal{S}$ that map observations differently, although their range, i.e. state space, is identical. This different encoding results in different transitions from one state to another when a new observation comes. In addition, when the observations and rewards are stochastic, for a given state space, there might be no encoding function from observations to the state space that induces Markov transitions for states and rewards. This might happen when the state space is too restrictive.

Definition 2.3. $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, T, \mu)$ represents *an interaction* $(O_1, R_1), (O_2, R_2), \dots, (O_T, R_T)$, *if there exists an encoding function* $E : \mathcal{O}^* \to \mathcal{S}$ *such that* $E(O_1) \sim \mu$ *and for all* $t \in \{2, 3, \dots, T\}$,

$$\mathbb{P}[E(O_{1:t})|O_{1:t-1}, A_{t-1}] = P(E(O_{1:t})|E(O_{1:t-1}), A_{t-1}),$$

and $r(E(O_{1:t}), A_t) = R_t$.

The different ways of encoding observations into states lead us to the model selection problem. Let $\mathcal{C}_0 \dots \mathcal{C}_{M-1}$ be M different model classes. Each model class $\mathcal{C}_i$ is a set of MDPs with a common state space $\mathcal{S}_i$, action space $\mathcal{A}_i$, and horizon T, but with different transition kernels and reward functions. Corresponding to each model class $\mathcal{C}_i$ there is a base algorithm Alg_i with a performance guarantee $\mathsf{B}_i(T, \delta)$, which is a high probability upper bound over the regret of using Alg_i in any MDP $\mathcal{M} \in \mathcal{C}_i$.

To give an intuition, consider the Atari example. One might consider an MDP class in which the states are the last two frames. In this case, there might be no Markov transition of states as the two frames do not capture enough information. Thus this model class can not represent our interaction with the environment (Atari game). If a model class $\mathcal{C}_i$ is not rich enough to contain an MDP that represents the interaction with the environment, then Alg_i might not be able to satisfy its regret bound $\mathsf{B}_i(T)$.

Definition 2.4. *(Misspecified and well-specified base algorithms) A base algorithm* Alg_i *is* well-specified *when the regret of its interaction with the environment for* T *rounds is upper bounded by* $\mathsf{B}_i(T)$*; otherwise, it is* misspecified.

We assume there exists at least one well-specified based algorithm (realizability) and also there is an order over the model classes/base algorithms from simple to complex. This is again a natural assumption when you think of the Atari example. If the state of i frames is enough for our interaction, so is the state of j frames for $j > i$.

Assumption 2.5. *We assume that there exists an order* $m \in [0 : M-1]$ *such that* Alg_m *is well specified. In addition, if* Alg_i *is a well specified base algorithm, then* Alg_j *is also well-specified for all* $j > i$.

The optimal order $m^* \in [0 : M-1]$ denotes the smallest index such that Alg_{m^*} is well specified and is unknown to the learner. Usually, the base algorithms are well specified (i.e. guarantee their regret bounds) if their corresponding model class can represent the interaction. And once a well-specified algorithm is found, there is no need to consider larger model classes. Back to our Atari example, if the state of i frames is enough to play optimally, then adding more frames has no point, and the optimal policy of the larger models with the state of j frames ($j > i$) does not achieve a larger gain. We formalized this in the following assumption.

Assumption 2.6. *We assume* $g^*_{m^*} = g^*_{m^*+1} = \dots = g^*_{M-1}$, *where* g^*_i *for* $i \in [m^* : M-1]$, *is the optimal gain of the representing MDP* $\mathcal{M}_i \in \mathcal{C}_i$.

Expansion of state space and/or action space from one model class to the next one is an example of having nested model classes satisfying the above assumptions.

Regret. We denote the gain of well-specified model classes by $g^\star := g^*_{m^*}$. The notion of regret (1) is designed for a single model class. In model selection for average reward, as m^* is unknown the performance of the algorithm is measured by,

$$\mathsf{Reg}(\mathsf{Alg}, T) := T g^*_{m^*} - \sum_{t=1}^{T} R_t. \tag{2}$$

The goal of an online model selection algorithm is to achieve a sublinear regret (2). This intuitively means learning the optimal model class and an optimal policy within it simultaneously.

3 ALGORITHM

First, we specify the base algorithms and their potential regret bounds $\mathsf{B}_i(T,\delta)$ that are compatible with our proposed meta-algorithm MRBEAR.

Definition 3.1. *The set of base algorithms* $\{\mathsf{Alg}_i\}_{i=0}^{M-1}$ *and their corresponding potential regrets of* $\{\mathsf{B}_i(T,\delta)\}_{i=0}^{M-1}$ *are* compatible *with MRBEAR, if* Alg_i *is based on the doubling trick, and there exists constants* $C_0 \le C_1 \le \cdots \le C_{M-1} \in \mathbb{R}^+$ *and warm up iterations* $\{\omega_i\}_{i=0}^{M-1}$ *such that,*

$$\mathsf{B}_i(T,\delta) = C_i\sqrt{T\log(T/\delta)}$$

is an upper bound for the regret (1) of Alg_i *with probability* $1-\delta$*, for* $T \ge \omega_i$ *.*

Remark 3.2. *The restriction to the doubling trick stems from its interruptibility, which allows MRBEAR to switch between the base algorithms after each epoch. In other words, the algorithm should not count on the last transition of the epochs. This means that the base algorithms are able to resume working, no matter how many interactions MRBEAR has had with the other base algorithms in between, and it does not affect the regret guarantee of the base algorithms.*

The increasing order on C_is is usually satisfied with noting that the larger model classes have greater parameters like the size of state space. One can see its relation to the assumption 2.5. The regret guarantee of a wide range of algorithms in the average reward literature is in the above form, such as PMEVI (Boone and Zhang, 2024), UCRL2 (Auer et al., 2008a), KL-UCRL (Filippi et al., 2010; Talebi and Maillard, 2018), UCRL2B (Fruit et al., 2020), etc.

Note that in converting the potential regret bounds of the base algorithms into the form of Definition 3.1, the choice of C_i and ω_i should also cover the lower-order terms. All characteristics of the base model classes that appear in their regret bound, such as the size of state and action spaces, etc, are encapsulated in C_i, and through that, they will contribute to the meta-algorithm performance guarantee.

Meta Algorithm. We propose an online model selection algorithm called **M**ultiplicative **R**egret **B**alancing and **E**limination in **A**verage **R**eward (MRBEAR). The meta algorithm MRBEAR is designed based on regret balancing and elimination technique and a novel misspecification test proper for average reward setting. Figure 1 shows the simplified pseudo-code of MRBEAR (refer to figure 2 for the detailed pseudo-code). Considering all M model classes, the algorithm starts with the initial set of active model classes $\mathcal{I}_0 = \{0,1,\dots,M-1\}$. After warming up all base algorithms, MRBEAR proceeds in epochs. In each epoch k, first, it updates the set of active model classes by checking the misspecification test (4) over all active classes. Then it takes the model class with the smallest B_i, i.e. $\mathcal{C}_{i_k}$, and runs the base algorithm Alg_{i_k} until it terminates due to doubling trick after n_k iterations (Auer et al., 2008b). Then the algorithm updates the set that keeps track of iterations consumed on each model class i until epoch k, i.e. $\mathcal{N}_{i,k}$. The cardinality of $|\mathcal{N}_{i,k}| = N_{i,k}$ shows the number of iterations that MRBEAR has listened to Alg_i until epoch k. MRBEAR terminates by spending T iterations. Intuitively, the meta algorithm rules out a base algorithm when it deviates from its potential regret bound, showing that it is misspecified. As it will be clear from the misspecification test, the meta-algorithm MRBEAR does not need to know how the base algorithms define their states; all that the meta-algorithm needs to know is the potential regret guarantees in the notion of Definition 3.1, and an upper bound on the $\mathsf{sp}(h^*)$.

We analyze the performance of MRBEAR in section 4.

Algorithm 1 Multiplicative Regret Balancing and Elimination in Average Reward (MRBEAR).

Require: $\delta, M, T, c_{h^*}, \mathsf{Alg}_{0:M-1}, \mathsf{B}_{0:M-1}, \omega_{0:M-1}$
  initialize the active class: $\mathcal{I}_0 = \{0,1,\dots,M-1\}$
  Warm up each Alg_i for ω_i iterations
  for $k \in \{1,2,\dots,K_T\}$ **do** ▷ Loop over epochs
    for $i \in \mathcal{I}_{k-1}$ **do**
      Check the misspecification test 4 for i
      Eliminate i if the test is violated (i.e. $\mathcal{I}_k = \mathcal{I}_{k-1}\backslash\{i\}$) ▷ Updating active classes
    end for
    Pick $i_k = \arg\min_{i\in\mathcal{I}_k} \mathsf{B}_i(N_{i,k-1},\delta)$ and run an inner episode of Alg_{i_k} on $\mathcal{C}_{i_k}$
  end for

4 REGRET ANALYSIS

In each time step, the regret balancing and elimination algorithm maintains a set of active model classes that contains the optimal model class with high probability (Abbasi-Yadkori et al., 2020; Agarwal et al., 2017). At each time step, the algorithm picks the active model class with the smallest regret guarantee; thus, all active model classes will have regrets in the same order.

In the average reward setting, we must commit to a model class for multiple iterations to capture its planning effect; thus, we check for model misspecification less often than every time step. We will show that even under this less-frequent checking, the active set will contain all of the well-specified model classes, and more importantly, their regret can be maintained in the same order.

Theorem 4.1 (Main theorem)**.** *By running the algorithm MRBEAR over M compatible base algorithms on model classes $\mathcal{C}_0$ to $\mathcal{C}_{M-1}$, and the unknown optimal model class $\mathcal{C}_{m^*}$, and known upper bound of $c_{h^*} \geq \max_i \mathsf{sp}(h_i^*)$, for $T \geq \sum_i \omega_i$ and all $0 < \delta < 1$, with probability of at least $1 - MT\delta$, the regret (2) is upper bounded by*

$$\mathsf{Reg}(\mathit{MRBEAR}, T) \leq \big(\frac{16m^* C_{m^*}^2 \log^{\frac{3}{2}}(T/\delta)}{C_0^2} + 4M\big)\mathsf{B}_{m^*}(T,\delta) + O(\log^{\frac{3}{2}}(T/\delta)).$$

Before explaining the proof, let us compare the regret bound of Theorem 4.1 with a trivial approach. Due to the realizability assumption 2.6, one can run Alg_{M-1} directly on the model class $M-1$, and it would give the regret bound of $\mathsf{B}_{M-1}(T,\delta) = C_{M-1}\sqrt{T\log(T/\delta)}$ (Refer to 3.1). In many cases, the growth of C_i is exponential in i (corollary 5.3). Therefore, the result in the theorem 4.1, i.e., $\tilde{O}(MC_{m^*}^2\mathsf{B}_{m^*}(T,\delta))$ can be notably better than $\mathsf{B}_{M-1}(T,\delta)$. For instance, in our Atari example, the number of states grows exponentially in the number of frames they are encoding. Thus the constants of the two bounds we are comparing here are in the form of MS^{3m^*} and S^M, which means using MRBEAR has an exponential advantage over conservatively running the biggest model class in terms of constants.

Remark 4.2. *It is important to note that Theorem 4.1 states that the dominant term of the regret bound which grows with T only depends on B_{m^*}. However, the warm-up phase of MRBEAR is still affected by the over-specified base algorithm through $\omega_{m^*+1}, \ldots, \omega_M$. This is aligned with the average reward setting (especially gain-optimal average reward) in which we implicitly consider large horizons, T. Although the warm-up length might depend on the characteristics of the misspecified model classes, by definition of* warm-up*, it is independent of T. In addition, for a variety of base algorithms, the warm-up length is constant, not growing with the size of the model. For example, the regret bound of UCRL2 (Auer et al., 2008b) and SCAL (Fruit et al., 2018b) holds for all $T \geq 1$ (meaning no warm-up), KL-UCRL (Filippi et al., 2010) for $T \geq 6$, which means a universal constant warm-up phase.*

Regret Decomposition. Now we present the ideas in the proof of the main theorem 4.1. We decompose the regret into the regret of each model class as follows,

$$\begin{aligned}\mathsf{Reg}(\mathsf{MRBEAR}, T) &:= Tg_{m^*}^* - \sum_{t=1}^{T} R_t \\ &= \sum_{i=0}^{M-1} \underbrace{\big[N_{i,\mathsf{K}(T)}\ g^\star - \sum_{t\in\mathcal{N}_{i,\mathsf{K}(T)}} R_t\big]}_{\mathsf{Reg}_i(\mathsf{Alg}_i, N_{i,\mathsf{K}(T)})}.\end{aligned} \tag{3}$$

where $\mathsf{K}(t)$ is the number of epochs until iteration t. The important note is that most of the average reward algorithms do not count on the transitions between epochs. In other words, after ending an epoch due to the doubling trick, they do not pay attention to or exploit that the first state of the next epoch is a successor of the last state, action pair of the previous epoch. This allows us to pause the base algorithm Alg_i on model class $\mathcal{C}_i$ after it terminates its epoch, then spending a couple of epochs in the other model classes, and then get back to $\mathcal{C}_i$ and resume the base algorithm Alg_i. Therefore the use of $t \in \mathcal{N}_{i,K(T)}$ in above expression is valid. Also, this changing of the model classes does not harm the regret guarantee of the well-specified model classes.

Misspecification test and Regret Balancing. Consider the following event,

$$\mathcal{G} = \{\forall i \in [m^* : M-1], k \in [1 : \mathsf{K}(T)+1] : \mathsf{R}_i(\mathsf{Alg}_i, N_{i,k}) \leq C_i\sqrt{N_{i,k}\log(\frac{N_{i,k}}{\delta})}\}$$

which captures the event that all the well-specified model classes, at the beginning of all epochs, satisfy their regret guarantee. $\mathcal{G}$ happens with a probability of at least $1-\delta(M-m^*)(\mathsf{K}(T)+1)$. Refer to appendix C.7 for the proof. Under the event $\mathcal{G}$ we know that if $i \in [m^* : M-1]$ is well specified, then for all $j \geq i$, $g_i^* = g_j^* = g^\star$. Also for all $k \in [1 : \mathsf{K}(T)+1]$ we have $-2\mathsf{sp}(h_i^*) \leq \mathsf{R}_i(\mathsf{Alg}_i, N_{i,k}) \leq \mathsf{B}_i(N_{i,k},\delta)$, where the lower bound is a well-known result (Refer to C.5). Thus by noting that $c_{h^*} \geq \mathsf{sp}(h_i^*)$, we can write,

$$\begin{aligned}\frac{\mathsf{B}_i(N_{i,k},\delta) + \sum_{t\in\mathcal{N}_{i,k}} R_t}{N_{i,k}} &\geq g_i^* = g_j^* \\ &\geq \frac{\sum_{t\in\mathcal{N}_{j,k}} R_t - 2c_{h^*}}{N_{j,k}}.\end{aligned}$$

This expression can be used as a misspecification test. Although g_i^* and g_j^* are unknown, the reward that the algorithm gathers in each model class and the regret bound $\mathsf{B}_i(N_{i,k},\delta)$ are observable and computable.

Therefore, violation of the inequality

$$\frac{\mathsf{B}_i(N_{i,k},\delta)+\sum_{t\in\mathcal{N}_{i,k}} r_t}{N_{i,k}} \geq \max_{j\geq i} \frac{\sum_{t\in\mathcal{N}_{j,k}} r_t - 2c_{h^*}}{N_{j,k}} \tag{4}$$

by an $i \in [0:M-1]$, indicates that *at least* Alg_i is not well specified and i should be eliminated from the set of active classes (or $\mathcal{G}$ has not occurred which is low probable). This also means that with high probability all the well-specified model classes remain active in all epochs. Thus we use the above inequality as the misspecification test in MRBEAR. Regarding the regret balancing idea, the following is the key lemma.

Lemma 4.3. *By running algorithm MRBEAR with compatible base algorithms, for all $0<\delta<1$, $k\in\mathbb{N}$ and any pair of $i,j\in\mathcal{I}_k$ where $i\neq j$, the followings hold,*

1. $\mathsf{B}_i(N_{i,k},\delta) \leq \mathsf{B}_j(N_{j,k},\delta) + \alpha_i \mathsf{B}_i(N_{i,k-1},\delta) + \beta$
2. $\frac{N_{i,k}}{N_{j,k}} \leq \big(\frac{C_j}{(1-\alpha_i)C_i} + \frac{\beta}{(1-\alpha_i)C_i\sqrt{N_{j,k}\log(N_{j,k}/\delta)}}\big)^2 \log(\frac{N_{j,k}}{\delta})$,

where $1/2 \leq \alpha_i = \frac{\log(\omega_i\vee 9)+1}{2\log(\omega_i\vee 9)} \leq \frac{3}{4}$ and $\beta = \mathsf{B}_{M-1}(\omega_{M-1},\delta) - \mathsf{B}_0(\omega_0,\delta)$.

The proof is in appendix C.8 by induction on k. It exploits the doubling trick that base algorithm uses, and the definition of B_i. From this lemma, we reach the following bound on the regret of misspecified model classes, based on the number of iterations consumed on them and on $\mathcal{C}_{m^*}$.

Lemma 4.4. *For any active model class $i\in\mathcal{I}_k$ such that $i<m^*$, under the event $\mathcal{G}$, the regret of model class i is bounded as follows,*

$$\mathsf{Reg}_i(N_{i,k-1}) \leq \big(\frac{N_{i,k-1}}{N_{m^*,k-1}} + \frac{1}{1-\alpha_i}\big)\mathsf{B}_{m^*}(N_{m^*,k-1}) + \frac{2N_{i,k-1}}{N_{m^*,k-1}}c_{h^*} + \frac{\beta}{1-\alpha_i}.$$

The proof can be found in appendix C.9. Again from lemma 4.3 we know that the fraction $\frac{N_{i,k-1}}{N_{m^*,k-1}}$ is controlled which gives us the following result.

Lemma 4.5. *For any active model class $i\in\mathcal{I}_{\mathsf{K}(T)+1}$ such that $i<m^*$, under the event $\mathcal{G}$, the regret of model class i is bounded as follows.*

$$\mathsf{Reg}_i(N_{i,\mathsf{K}(T)}) \leq \frac{C_{m^*}^3\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2} + \frac{1}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + O(\log^{\frac{3}{2}}(T/\delta))$$

Recall the regret decomposition 3. The factor $\frac{1}{2}\leq\alpha_i\leq\frac{3}{4}$ in lemma 4.3 makes a multiplicative balance between the guarantees, which roughly means that for all $j\geq m^*$ the guarantee is $\mathsf{B}_j \leq \frac{1}{1-\alpha_j}\mathsf{B}_{m^*}$. Note these are well-specified regret guarantees so that they can truly be an upperbound on the performance of Alg_j. Thus, we have $\sum_{i=m^*}^{M-1}\mathsf{Reg}_i \leq \frac{M-m^*}{1-\alpha_i}\mathsf{B}_{m^*}$. For the other term corresponded to misspecified model classes, i.e., $\sum_{i=0}^{m^*-1}\mathsf{Reg}_i$, we use lemma 4.5. Adding these two parts implies theorem 4.1 (see appendix C.9).

5 APPLICATION TO REPEATED GAMES

In this section, we come back to our motivating question, and apply the model selection result to repeated games. The goal of the learner is to maximize its utility without knowledge of the opponent's memory order. The two players play a stage game G for T simultaneous rounds. Learner's utility function is denoted by $\boldsymbol{U}:\mathcal{A}\times\mathcal{B}\to[0,1]$, and cardinality of action sets are $\boldsymbol{A}=|\mathcal{A}|$ and $\boldsymbol{B}=|\mathcal{B}|$. The learner's action, $A_t\in\mathcal{A}$, and opponent's action, $B_t\in\mathcal{B}$, at round t make the interaction of $O_t=(A_t,B_t)$. We save the history of the game in a tuple $((A_1,B_1),(A_2,B_2),\ldots,(A_t,B_t)) = (O_1,O_2,\ldots,O_t)$. An m-th order policy π is a mapping from the the last m interactions in history to the distributions over the actions. The opponent's fixed, unknown policy is denoted by ψ and the best response (in the repeated game) against it is computed as follows, $\mathsf{BR}(\psi,s_1) = \arg\max_{\pi\in\Pi_{\mathcal{A}}}\mathbb{E}\sum_{t=1}^{T}\boldsymbol{U}(A_t,B_t) = \arg\max_{\pi\in\Pi_{\mathcal{A}}} V_T^{\psi,\pi}(s_1)$. The Learner considers a model class $\mathcal{C}_m$ for each memory order of $0\leq m\leq M-1$.

Assumption 5.1. *We assume ψ is an m^*-th order policy where m^* is unknown to the learner, while the learner knows of an upper bound M such that, $M>m^*$. The opponent's policy ψ induces a weakly communicating MDP on well-specified classes.*

The weakly communicating assumption is satisfied when ψ takes no action with zero probability. The opponent by taking a policy ψ in the order of m^* makes all $\mathsf{Alg}_{m^*}\ldots\mathsf{Alg}_{M-1}$ well specified. For a formal definition of policies and repeated game setting refer to appendix B.7. The following proposition is crucial to show that the repeated game setting can be modeled as an average reward RL, and the assumptions needed for applying MRBEAR are met.

Proposition 5.2. *Against any m-th order policy ψ and for any policy $\pi\in\Pi_{\mathcal{A}}$, there exists a policy $\pi'\in\bigcup_{i=0}^{m}\Pi_{\mathcal{A}}^i$ in the order of at most m, where π and π' have the same value, i.e., $V_T^{\psi,\pi}(s_1)=V_T^{\psi,\pi'}(s_1)$ for all $T\in\mathbb{N}$.*

This proposition intuitively states that the set of policies which are depending on whatever ψ depends on, contains the optimal policy (proof in appendix C.1). It implies that the domain of the maximum in the definition of best response can be $\Pi_{\mathcal{A}}^{m^*}$ instead of $\Pi_{\mathcal{A}}$.

Corollary 5.3. *For T rounds of playing a repeated game, against an opponent with an unknown memory $m^* \leq M$, using MRBEAR with instances of PMEVI-DT as the base algorithms, gives us the following bound on the regret with a probability of at least $1 - 26MT\delta$,*

$$\mathsf{Reg}(T) \in O\big(M(\boldsymbol{A}^{m^*+1}\boldsymbol{B}^{m^*}\mathsf{sp}(h^*))^{\frac{3}{2}}\ T^{\frac{1}{2}}\log^{\frac{3}{2}}(T/\delta)\big).$$

This corollary is the result o applying MRBEAR to the repeated game and the main theorem 4.1.

Remark 5.4. *For the repeated game application, we give an upper bound for the $\mathsf{sp}(h^*)$ based on the Kemeny's index of the Markov chain induced by ψ in Appendix B.5.*

In this setting, the size of the state space (i.e., $(\mathcal{A}\times\mathcal{B})^m$) increases exponentially with respect to the memory m. Therefore, instead of having a regret bound in $\tilde{O}(\sqrt{\boldsymbol{A}^M\boldsymbol{B}^{M-1}T})$ by directly learning in $\mathcal{C}_M$, MRBEAR enjoys from a regret in $\tilde{O}(M\sqrt{\boldsymbol{A}^{3(m^*+1)}\boldsymbol{B}^{3m^*}T})$ which is an exponential improvement with respect to the memory. Furthermore, our proposed lower bound in the next section 5.1 , shows that the exponent of m^* in regret bound is inevitable. Refer to Appendix B.8 to see how this bound changes for different types of opponents, and to Appendix B.6 for further explanation on using PMEVI-DT as a base algorithm.

5.1 Lower Bound

The MDPs constructed in the repeated game setting have a special structure that prevents us from directly using previous lower bounds designed for generic MDPs (Auer et al., 2008b; Osband and Van Roy, 2016). This is because they propose hard-to-learn MDPs that do not obey the structure of our setting. In this section, we present a minimax lower bound on the regret of any algorithm, using Le Cam method. Towards that, we give a divergence decomposition lemma (appendix C.10), which is similar to the the key lemma in Le Cam-based lower bounds in bandit literature (refer to C.12). Then, using de Bruijn sequences (de Bruijn, 1975; Crochemore et al., 2021), we design two complementing opponent policies ψ and ψ' that force the learner to have high regret in the interaction with at least one of them, even with the advantage of knowing the opponent's memory order.

Theorem 5.5. *Suppose the number of opponent's actions $|\mathcal{B}| \geq 3$ and number of learners actions $|\mathcal{A}| \geq 2$. For any fixed memory $m \in \mathbb{N}$ known for the learner, and any algorithm Alg, there exist a stage game with utility $U : \mathcal{A}\times\mathcal{B} \to [0,1]$, and an opponent's policies ψ such that*

$$\mathsf{R}(Alg, T) \in \Omega(\frac{1}{m}\sqrt{\boldsymbol{A}^{m-1}\boldsymbol{B}^{m-1}T}).$$

The important takeaway is that the exponential dependency to the memory m^* in theorem 4.1 is inevitable. The proof of the previous lemma and theorem is in appendix C.10.

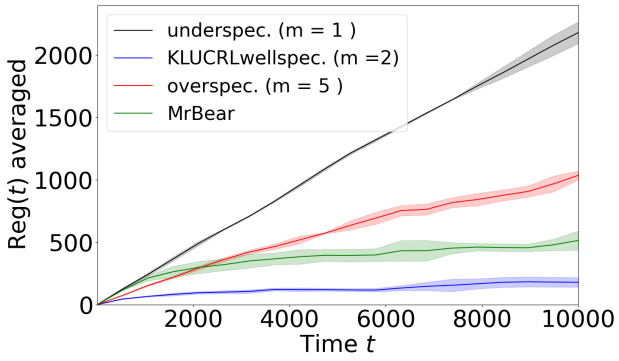
6 EXPERIMENTS

We demonstrate the empirical performance of the meta algorithm MRBEAR, confirming the previous theoretical results. We choose a stage game (refer to 3), and a second-order policy for the opponent, such that the optimal order for the repeated interaction on the game is $m^* = 2$. This means that the optimal strategy for the learner needs to consider the last two previous interactions to choose the optimal action. Refer to appendix D for a more detailed explanation. We apply MRBEAR to three model classes with the orders of 1 (under-specified), 2 (well-specified), and 5 (over-specified).

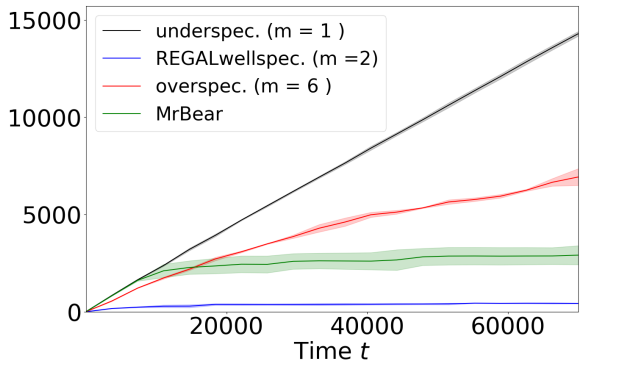
We run MRBEAR over Regal, UCRL2, and KLUCRL as base algorithms. The performance of algorithms in terms of their cumulative regret is depicted. Since there is no optimal first-order policy for the learner, the regret of base algorithms on $m = 1$ is linear (Figure 1). On the other hand, knowing the optimal model class $m = 2$ in hindsight, one could directly choose the well-specified base algorithm and achieve the best performance in the plot. Finally, due to the realizability assumption, one could conservatively run the algorithm of order $m = 5$. In this overspecified case, the regret is sublinear, but due to the large size of MDPs its curve is too slow. The performance of MRBEAR is between this conservative case and the best case. More experiments are in Appendix D.

7 DISCUSSION AND FUTURE WORK

We propose MRBEAR as an online model selection algorithm for the average reward RL. We show that it enjoys a regret of $\tilde{O}(MC_{m^*}^2\mathsf{B}_{m^*}(T,\delta))$, where M is the number of model classes and $\mathsf{B}_{m^*}(T,\delta)$ is the regret guarantee for the optimal well-specified model class. We also construct a framework for solving utility maximization in repeated games using average reward reinforcement learning, when the learner is facing an



(a)



(b)

Figure 1: The cumulative regret of three instances of (a) KLUCRL and (b) Regal in repeated games setting and the meta-algorithm MRBEAR over these base algorithms.

opponent with unknown limited memory m^*. We obtain a regret bound in $\tilde{O}(M\sqrt{\boldsymbol{A}^{3(m^*+1)}\boldsymbol{B}^{3m^*}T})$ for the learner and show that the exponential dependency in m^* is unavoidable.

As discussed in Section 3, our proposed misspecification test (4) and the way MRBEAR works is independent of how base algorithms work. This allows MRBEAR to be applied on either model-based or model-free base algorithms (Abbasi-Yadkori et al., 2019; Zhang and Xie, 2023; Wei et al., 2020). More importantly, since its performance is independent of the state space of underlying model classes, MRBEAR is also compatible with base algorithms that use function approximation to deal with large state spaces, as long as there is a theoretical regret bound for them.

Future Work. One interesting direction is *online model creation*, i.e. gradually enrich or simplify the model class based on previous interactions. Another extension of this work and can come from using the shared information among model classes for structured settings to transfer learning between them and gaining better bounds. On the game theoretic side, studying a non-stationary opponent's strategy ψ, particularly when it is the result of an algorithm that the opponent deploys, is a promising avenue.

8 ACKNOWLEDGMENT

The authors thank Ronan Fruit and Csaba Szepesvari for helpful comments. This work was funded by an NSERC Discovery Grant, a CIFAR Canada AI Research Chair (Alberta Machine Intelligence Institute).

References

Abbasi-Yadkori, Y., Lazic, N., Szepesvari, C., and Weisz, G. (2019). Exploration-enhanced politex. *arXiv preprint arXiv:1908.10479*.

Abbasi-Yadkori, Y., Pacchiano, A., and Phan, M. (2020). Regret balancing for bandit and rl model selection. *arXiv preprint arXiv:2006.05491*.

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

Agrawal, S. and Jia, R. (2017). Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. *Advances in neural information processing systems*, 30.

Arora, R., Dinitz, M., Marinov, T. V., and Mohri, M. (2018). Policy regret in repeated games. *Advances in Neural Information Processing Systems*, 31.

Arunachaleswaran, E. R., Collina, N., and Schneider, J. (2024). Pareto-optimal algorithms for learning in games. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 490–510.

Assos, A., Dagan, Y., and Daskalakis, C. (2024). Maximizing utility in multi-agent environments by anticipating the behavior of other learners. *arXiv preprint arXiv:2407.04889*.

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

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

Bartlett, P. L., Boucheron, S., and Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, 48:85–113.

Bartlett, P. L. and Tewari, A. (2012). Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. *arXiv preprint arXiv:1205.2661*.

Boone, V. and Zhang, Z. (2024). Achieving tractable minimax optimal regret in average reward mdps. *arXiv preprint arXiv:2406.01234*.

Bourel, H., Maillard, O., and Talebi, M. S. (2020). Tightening exploration in upper confidence reinforcement learning. In *International Conference on Machine Learning*, pages 1056–1066. PMLR.

Braverman, M., Mao, J., Schneider, J., and Weinberg, M. (2018). Selling to a no-regret buyer. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 523–538.

Brown, W., Schneider, J., and Vodrahalli, K. (2024). Is learning in games good for the learners? *Advances in Neural Information Processing Systems*, 36.

Crochemore, M., Lecroq, T., and Rytter, W. (2021). *125 Problems in Text Algorithms: With Solutions*. Cambridge University Press.

Daskalakis, C., Golowich, N., and Zhang, K. (2023). The complexity of markov equilibrium in stochastic games. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 4180–4234. PMLR.

de Bruijn, N. G. (1975). Acknowledgement of priority to c. flye sainte-marie on the counting of circular arrangements of 2^n zeros and ones that show each n-letter word exactly once.

Deng, Y., Schneider, J., and Sivan, B. (2019). Strategizing against no-regret learners. *Advances in neural information processing systems*, 32.

Filippi, S., Cappé, O., and Garivier, A. (2010). Optimism in reinforcement learning and kullback-leibler divergence. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 115–122. IEEE.

Foster, D. J., Kale, S., Mohri, M., and Sridharan, K. (2017). Parameter-free online learning via model selection. *Advances in Neural Information Processing Systems*, 30.

Foster, D. J., Krishnamurthy, A., and Luo, H. (2019). Model selection for contextual bandits. *Advances in Neural Information Processing Systems*, 32.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Fruit, R., Pirotta, M., and Lazaric, A. (2018a). Near optimal exploration-exploitation in non-communicating markov decision processes. *Advances in Neural Information Processing Systems*, 31.

Fruit, R., Pirotta, M., and Lazaric, A. (2020). Improved analysis of ucrl2 with empirical bernstein inequality. *arXiv preprint arXiv:2007.05456*.

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

Ghosh, A., Chowdhury, S. R., and Ramchandran, K. (2021). Model selection for generic reinforcement learning. *arXiv preprint arXiv:2107.05849*.

Jin, C., Liu, Q., Wang, Y., and Yu, T. (2021). V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*.

Krishnamurthy, S. K., Propp, A. M., and Athey, S. (2024). Towards costless model selection in contextual bandits: A bias-variance perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 2476–2484. PMLR.

Lugosi, G. and Nobel, A. B. (1999). Adaptive model selection using empirical complexities. *The Annals of Statistics*, 27(6):1830–1864.

Maillard, O.-A., Nguyen, P., Ortner, R., and Ryabko, D. (2013). Optimal regret bounds for selecting the state representation in reinforcement learning. In *International Conference on Machine Learning*, pages 543–551. PMLR.

Marinov, T. V. and Zimmert, J. (2021). The pareto frontier of model selection for general contextual bandits. *Advances in Neural Information Processing Systems*, 34:17956–17967.

Massart, P. (2007). *Concentration inequalities and model selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII-2003*. Springer.

Mnih, V. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Nguyen-Tang, T. and Arora, R. (2024). Learning in markov games with adaptive adversaries: Policy regret, fundamental barriers, and efficient algorithms. *arXiv preprint arXiv:2411.00707*.

Ortner, R. (2007). Pseudometrics for state aggregation in average reward markov decision processes. In *International Conference on Algorithmic Learning Theory*, pages 373–387. Springer.

Ortner, R. (2020). Regret bounds for reinforcement learning via markov chain concentration. *Journal of Artificial Intelligence Research*, 67:115–128.

Ortner, R. (2024). Markov chain estimation, approximation, and aggregation for average reward markov decision processes and reinforcement learning. *submitted to Handbook of Statistics*.

Ortner, R., Maillard, O.-A., and Ryabko, D. (2014). Selecting near-optimal approximate state representations in reinforcement learning. In *International Conference on Algorithmic Learning Theory*, pages 140–154. Springer.

Osband, I. and Van Roy, B. (2016). On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*.

Pacchiano, A., Dann, C., Gentile, C., and Bartlett, P. (2020). Regret bound balancing and elimination for model selection in bandits and rl. *arXiv preprint arXiv:2012.13045*.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qian, J., Fruit, R., Pirotta, M., and Lazaric, A. (2019). Exploration bonus for regret minimization in discrete and continuous average reward mdps. *Advances in Neural Information Processing Systems*, 32.

Seneta, E. (1984). Explicit forms for ergodicity coefficients and spectrum localization. *Linear Algebra and its Applications*, 60:187–197.

Seneta, E. (1993). Sensitivity of finite markov chains under perturbation. *Statistics & probability letters*, 17(2):163–168.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Talebi, M. S. and Maillard, O.-A. (2018). Variance-aware regret bounds for undiscounted reinforcement learning in mdps. In *Algorithmic Learning Theory*, pages 770–805. PMLR.

Vapnik, V. (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media.

Wei, C.-Y., Hong, Y.-T., and Lu, C.-J. (2017). Online reinforcement learning in stochastic games. *Advances in Neural Information Processing Systems*, 30.

Wei, C.-Y., Jahromi, M. J., Luo, H., Sharma, H., and Jain, R. (2020). Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International conference on machine learning*, pages 10170–10180. PMLR.

Xie, Q., Chen, Y., Wang, Z., and Yang, Z. (2020). Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on learning theory*, pages 3674–3682. PMLR.

Zhang, Z. and Ji, X. (2019). Regret minimization for reinforcement learning by evaluating the optimal bias function. *Advances in Neural Information Processing Systems*, 32.

Zhang, Z. and Xie, Q. (2023). Sharper model-free reinforcement learning for average-reward markov decision processes. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5476–5477. PMLR.

Zhong, H., Yang, Z., Wang, Z., and Jordan, M. I. (2021). Can reinforcement learning find stackelberg-nash equilibria in general-sum markov games with myopic followers? *arXiv preprint arXiv:2112.13521*.

Zurek, M. and Chen, Y. (2024). The plug-in approach for average-reward and discounted mdps: Optimal sample complexity analysis. *arXiv preprint arXiv:2410.07616*.

Supplementary Materials

A Extensive Related Work

Average Reward. Auer et al. (2008a) achieve a regret bound of $O(DS\sqrt{AT})$ with their seminal algorithm UCRL2. Further work improves the dependency on the diameter D by considering the number of next probable states or local diameters (Fruit et al., 2020; Bourel et al., 2020). Other work uses mixing time or hitting time of the Markov chain induced by optimal policy (Wei et al., 2020; Ortner, 2020; Zurek and Chen, 2024), or by using the variance of the next state to have problem-dependent regret bounds (Talebi and Maillard, 2018). Diameter based algorithms usually assume communicating MDPs. However, Fruit et al. (2018a) propose a UCRL-based algorithm for weakly communicating MDPs, by concatenating UCRL2 with an estimator of communicating part of the MDP. When rewards are bounded, it is not hard to show that $\mathsf{sp}(h^*) \leq D$. There even exist weakly communicating MDPs with infinite diameter but with finite $\mathsf{sp}(h^*)$. These show that results that capture $\mathsf{sp}(h^*)$ tend to be stronger (Fruit et al., 2018b; Zhang and Ji, 2019; Boone and Zhang, 2024). The leading algorithm is the recently introduced PMEVI, which efficiently achieves the optimal regret bound of $\tilde{O}(\sqrt{\mathsf{sp}(h^*)SAT})$ (Boone and Zhang, 2024). Its power in comparison to previous work, in addition to its tractability, is that it does not need the prior knowledge of $\mathsf{sp}(h^*)$. In Table 1, we gather the algorithms developed in the average reward literature with their regret bound. $K_{s,a}$ is the number of possible next states from s and taking action a, i.e. $K_{s,a} := |\{x \in \mathcal{S} : P(x|s,a) > 0\}|$. D_s is the local diameter (refer to Bourel et al. (2020) for definition). $L_{s,a} := (\sum_{x\in\mathcal{S}} \sqrt{P(x|s,a)(1-P(x|s,a))})^2$. $V^*_{s,a}$ is the variance of h^* when the distribution over the states come from $P(.|s,a)$. In the environments where the next state is deterministically implied from the previous state and action, $K_{s,a} = 1$ and $L_{s,a} = V^*_{s,a} = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Playing against self-oblivious opponents with limited memory is an example of such environments (section 5).

Model Selection. There is a rich literature on model selection for statistical learning (Vapnik, 2006; Lugosi and Nobel, 1999; Massart, 2007; Bartlett et al., 2002) and online learning (Freund and Schapire, 1997; Foster et al., 2017). For the case of contextual bandit and reinforcement learning, many approaches rely on the same underlying idea: using a meta-algorithm above all model classes, to determine which of the model classes are well specified based on interaction with the environment. This can be accomplished by online misspecification tests, like checking whether the empirical regret satisfies known bounds (Abbasi-Yadkori et al., 2020; Pacchiano et al., 2020). Some work assumes nested structure on the model classes (Ghosh et al., 2021; Foster et al., 2019), but some others do not (Agarwal et al., 2017). The cost of model selection under different assumptions can be either multiplicative (Pacchiano et al., 2020) or additive (Ghosh et al., 2021) to the regret of the best well-specified model class (Marinov and Zimmert, 2021; Krishnamurthy et al., 2024). For instance, Ghosh et al. (2021) assume that they can obtain information from the model classes simultaneously, in addition to a separability assumption for the underlying model classes. For an analogy, if we think of the model classes as a set of arms, these assumptions are similar to the full-information setting and knowing a lower bound for the gap between the quality of the arms. Our setting is more similar to a bandit-information without assuming separability, making our result more general. In state aggregation (Ortner, 2007, 2024) and state representation (Maillard et al., 2013; Ortner et al., 2014), the same concern is taken into account, i.e. finding the proper size of MDP, however their approach is slightly different.

Learning in Markov Games. Another related direction to this work is learning in Markov games (also known as stochastic games) (Wei et al., 2017; Zhong et al., 2021; Xie et al., 2020). One challenge for learning in Markov games lies in the trade-off between learning a *Markov* (C)CE (corresponding to time-independent no-regret policies) and how the regret depends on the number of agents, i.e. one can not have both Markov policies and non-exponential dependency on the number of players (Daskalakis et al., 2023; Jin et al., 2021). Xie et al. (2020) by assuming linearly representable transition kernel and reward functions, apply function approximation to learn course correlated equilibria in Markov game. The setting in this paper differs from the previous works either in the notion of regret, assuming different levels of memory for the opponent, or the measure of complexity. An

Algorithm	**Bound $\tilde{O}(.)$**	**Comment(Reference)**
Regal	$O\big(\mathsf{sp}(h^*)S\sqrt{AT\log(AT/\delta)}\big)$	Known $\mathsf{sp}(h^*)$ (Bartlett and Tewari, 2012)
UCRL2	$DS\sqrt{AT}$	(Auer et al., 2008b)
PSRL	$DS\sqrt{AT}$	Bayesian regret (Agrawal and Jia, 2017)
SCAL	$\mathsf{sp}(h^*)S\sqrt{AT}$	Known $\mathsf{sp}(h^*)$ (Fruit et al., 2018b)
SCAL+	$D\sqrt{\sum_{s,a}K_{s,a}T}$	(Qian et al., 2019)
UCRL2B	$O\big(\sqrt{D\sum_{s,a}K_{s,a}T\log(T)\log(T/\delta)}\big)$	(Fruit et al., 2020)
UCRL3	$(D\sqrt{T}+\sqrt{T\sum_{s,a}(D_s^2L_{s,a}\vee 1)})$	(Bourel et al., 2020)
KL-UCRL	$D\sqrt{T}+\sqrt{S\sum_{s,a}V^*_{s,a}\ T}$	Ergodic MDPs (Talebi and Maillard, 2018)
Politex	$(t_{mix})^3t_{hit}\sqrt{SA}T^{\frac{3}{4}}$	Ergodic MDPs, Model-free (Abbasi-Yadkori et al., 2019)
MDP-OOMD	$t_{mix}\sqrt{t_{hit}AT}$	Ergodic MDPs, Model-free (Wei et al., 2020)
EBF	$\sqrt{\mathsf{sp}(h^*)SAT}$	Optimal, Known $\mathsf{sp}(h^*)$ (Zhang and Ji, 2019)
Optimistic-Q	$\mathsf{sp}(h^*)(SA)^{\frac{1}{3}}T^{\frac{2}{3}}$	Model-free (Wei et al., 2020)
UCB-AVG	$S^5A^2\mathsf{sp}(h^*)\sqrt{T}$	Model-free, Known $\mathsf{sp}(h^*)$ (Zhang and Xie, 2023)
PMEVI-DT	$\sqrt{\mathsf{sp}(h^*)SAT}$	Optimal, Unknown $\mathsf{sp}(h^*)$ (Boone and Zhang, 2024)
Lower Bound	$\Omega(\sqrt{\mathsf{sp}(h^*)SAT})$	(Fruit et al., 2018b) (Zhang and Ji, 2019)

Table 1: Developed algorithms in tabular average reward setting.

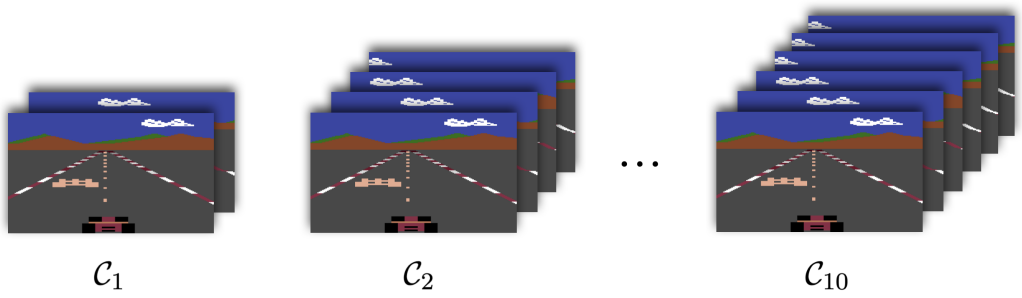


Figure 2: The Atari Example: The underlying model classes upon which MRBEAR acts, can be different on the number of previous frames the state encodes (image reference: Atari 2600 - Pole Position- 1983).

attractive recent line of research studies regret minimization when the opponent runs online learning algorithms (Brown et al., 2024; Deng et al., 2019; Braverman et al., 2018; Arunachaleswaran et al., 2024). In our work, we bound average reward regret, a more demanding benchmark which better captures utility maximization in repeated play than external regret. Assos et al. (2024) show that utility maximization against an opponent deploying multiplicative weights updates (MWU) algorithms is tractable in zero-sum repeated games, while there is no Fully Polynomial Time Approximation Scheme (FPTAS) for utility maximization against a best-responding opponent in general sum games. This is aligned with our result in Theorem 5.5, where we see that exponential dependency on the opponent's memory m^* is inevitable. Another recent work is (Nguyen-Tang and Arora, 2024) in which a bounded memory for the opponent is assumed and the performance of the learner is measured by policy regret (Arora et al., 2018). However Nguyen-Tang and Arora (2024) model the interaction as a Stackelberg game, in episodic Markov Games. We study average reward in simultaneous repeated games which are arguably more complicated settings. Zhong et al. (2021) use reinforcement learning to solve the Stackelberg equilibrium for myopic followers. Wei et al. (2017) study stochastic games under average reward using the diameter as the complexity measure. Our own work proves bounds based on $\mathsf{sp}(h^*)$ which are tighter than those based on diameter.

B Additional Explanations

B.1 MDP Classes

Here is the definition for different classes of MDPs (refer to (Puterman, 2014)):

Definition B.1. *We say that an MDP is* recurrent *(or* ergodic*), if* every *deterministic memoryless policy induces a Markov chain with a single recurrent class;* Communicating*, if it has a finite diameter (i.e., for every pair of states s and s' there exists a deterministic memoryless policy under which s' is accessible from s with positive probability);* Weakly communicating*, if there exists a closed set of states in which each state is accessible from every other using some deterministic policy, plus a possibly empty set of states which is transient under* every *policy.*

B.2 Figure

Figure 2 shows different model classes one can consider in the Atari example based on the number of frames, and Figure 3 is the stage game used in the experiments.

B.3 Algorithms Pseudo Code

Here we present a more detailed version of MRBEAR pseudo code (Algorithm 2).

B.4 Connection to Other Notions of Regret

The average reward regret captures utility maximization since it compares the learner's performance to the highest possible cumulative utilities. This is not the case in external regret, swap regret, and even dynamic

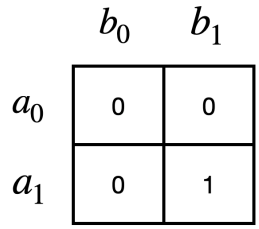


Figure 3: The stage game G used in the experiment. The numbers represent the learner's utility at each action profile.

Algorithm 2 Multiplicative Regret Balancing and Elimination in Average Reward (MRBEAR).

Require: $\delta, M, T, c_{h^*}, \mathsf{Alg}_{0:M-1}, \mathsf{B}_{0:M-1}, \omega_{0:M-1}$
  $\mathcal{I}_0 = \{0, 1, \dots, M-1\}$, $t_0 = 0$
  for $i \in \{0, \dots, M-1\}$ **do** ▷ Warm up phase
    Run Alg_i for ω_i iterations
    $N_{i,0} = \tau_i$, $\mathcal{N}_{i,0} = \{t_0 + 1, \dots, t_0 + \omega_i\}$
    $t_0 = t_0 + \omega_i$
  end for
  for $k \in \{1, 2, \dots\}$ **do** ▷ Loop over epochs
    if $t_{k-1} \geq T$ **then** break
    end if
    for $i \in \mathcal{I}_{k-1}$ **do**
      if ▷ Checking misspecification test

$$\frac{\mathsf{B}_i(N_{i,k-1}, \delta) + \sum_{t \in \mathcal{N}_{i,k-1}} r_t}{N_{i,k-1}} < \max_{j \geq i} \frac{\sum_{t \in \mathcal{N}_{j,k-1}} r_t - 2c_{h^*}}{N_{j,k-1}}$$

    then
        $\mathcal{I}_k = \mathcal{I}_{k-1} \backslash \{i\}$ ▷ Updating active classes
      end if
    end for
    Pick $i_k = \arg\min_{i \in \mathcal{I}_k} \mathsf{B}_i(N_{i,k-1}, \delta)$
    Run an inner episode of Alg_{i_k} on $\mathcal{C}_{i_k}$ ▷ (n_{i_k} iterations)
    Update parameters:
    $t_k = t_{k-1} + n_{i_k}$, $\mathcal{N}_{i_k,k} = \mathcal{N}_{i_k,k-1} \cup \{t_{k-1} + 1, \dots t_k\}$
    $N_{i_k,k} = N_{i_k,k-1} + n_{i_k}$, $\forall j \neq i_k : N_{j,k} = N_{j,k-1}$
  end for

regret. In dynamic regret which is the strongest among them, the baseline captures immediate best actions, without considering any planning effect of the actions. In this section, we compare average reward regret to dynamic regret and episodic regret in online RL (which is an external notion of regret) to illustrate the previous point in detail.

Episodic Regret. Episodic regret is a common notion in online reinforcement learning. Suppose there are K episodes, each of H iterations' length. The episodic regret is defined as

$$\mathsf{R}_E(Alg, \psi) = \sum_{k=1}^{K} V_H^{\pi^*}(s_1) - V_H^{\pi_k}(s_1),$$

where π_k is the fixed policy employed by the algorithm in episode $k \in [K]$. This can be a natural evaluation when there is an episodic structure in the environment. The point that makes this regret easier to deal with, in comparison to average reward regret, is that we usually want to be sublinear in K, the number of episodes. Because of that, one can rewrite episodic regret as the sum of K practically (and not statistically) independent immediate regrets $\Delta_k = V_H^{\pi^*}(s_1) - V_H^{\pi_k}(s_1)$, making it similar to external regret. However, this is not the case in average reward regret, as it can be interpreted as a single episode of running in which we want to find the highest rewarding sequence of actions by considering the planning effects of actions we take.

Dynamic Regret. Suppose an episodic regret with a planning horizon of $H = 1$ (i.e., $K = T$), where the initial state of the episode $k > 1$ is the state that the transition dynamic suggests based on s_{k-1} and a_{k-1}. In this case, you can change the policy at every iteration. Furthermore, $V_H^{\pi^*}(s_k)$ coincides with the highest possible immediate reward. This gives us the *adaptive* dynamic regret,

$$\begin{aligned}\mathsf{R}_{AD}(Alg, \psi) &= \mathbb{E}\big[\sum_{k=1}^{\mathsf{K}(T)} V_1^{\pi^*}(S_k) - V_1^{\pi_k}(S_k)\big] \\ &= \mathbb{E}\big[\sum_{t=1}^{T} \max_{a \in \mathcal{A}} r(a, S_t) - r(A_t, S_t)\big]\end{aligned}$$

We emphasize *adaptive* dynamic regret because the current action a_t not only determines the current reward but also affects the future rewards through the transition kernel. Even this regret is weaker than R_V, exactly because of this adaptivity point. Formally,

$$\mathbb{E}\big[\sum_{t=1}^{T} \max_{a \in \mathcal{A}} r(a, S_t)\big] \leq \max_{a_{1:T} \in \mathcal{A}^T} \mathbb{E}\big[\sum_{t=1}^{T} r(a_t, S_t)\big] = V_T^*(s_1),$$

which shows that R_V considers a stronger competitor. The equality holds when the adaptivity collapses, and there is no planning effect in taking actions, as in the contextual bandit setting. In that case, the dynamic regret coincides with R_V.

B.5 From ψ to $\mathsf{sp}(h^*)$

In this section, we study how different choices of ψ with the same memory of m affect the span of optimal bias and give an upper bound c_{h^*} on $\mathsf{sp}(h^*)$ based on the Kemeny's index induced by ψ.

Span of h^* is a parameter that appears in many problem-dependent regret bounds in the average reward RL literature (Boone and Zhang, 2024; Zhang and Ji, 2019; Fruit et al., 2018b). From the Poisson equation, we have $(I - P^\pi)h^\pi = (I - P_\infty^\pi)r^\pi$, where $P_\infty^\pi = lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{\mathbb{N}} (P^\pi)^n$. In the aperiodic Markov chains, the limit is well-defined. Note that $g^\pi = P_\infty^\pi r^\pi$. The fact that the rows of P_∞^π are the stationary distribution of the Markov chain generated by P^π , denoted by μ_∞^π, helps in giving the intuition and why $g^\pi \in \mathbb{R}\mathbf{1}$ (in ergodic MDPs). The optimal bias of a Markov chain can be computed as $h^* = (I - P^*)^\# r^* = (I - P^* + P_\infty^*)^{-1}(I - P_\infty^*)r^*$, where $A^\#$ is the group inverse of a singular matrix A, and superscript $*$ is to note the use of optimal policy. $Z^\pi := (I - P^\pi + P_\infty^\pi)^{-1}$ is called the fundamental matrix, also $H^\pi := (I - P^\pi)^\#$ is called the deviation matrix.

To upper bound the span of optimal bias, we use the ergodicity coefficient (Seneta, 1984, 1993) which can be interpreted as a matrix norm that captures the maximum total variation between a pair of its rows. Note Z^π and H^π have the rows that sum up to one and zero respectively.

Definition B.2. *(Seneta, 1993) For any $n \times n$ matrix $A = \{a_{ij}\}$ with all its rows sum up to a (i.e., $A\mathbf{1} = a\mathbf{1}$), the ergodicity coefficient of A is,*

$$\tau_1(A) = \sup_{\{\delta : \|\delta\|_1 = 1, \delta\mathbf{1}=0\}} \|\delta A\|_1 = \max_{i,j} \frac{1}{2} \sum_{k=1}^{n} |A_{ik} - A_{jk}|$$

The following lemma leads us to the upper bound B.4.

Lemma B.3. *(Seneta, 1993) For an ergodic transition matrix P and its corresponding fundamental matrix Z and deviation matrix H,*

$$\frac{1}{\min_{2 \le k \le n} |1 - \lambda_k|} \le \tau_1(H) = \tau(Z) \le \sum_{k=2}^{n} \frac{1}{|1 - \lambda_k|}$$

where $\lambda_i s$ are the eigenvalues of P, enumerated such as $1 = \lambda_1 > |\lambda_2| \ge \cdots \ge |\lambda_n|$.

Note that the set of eigenvalues of H is $\{0, (1-\lambda_2)^{-1}, \ldots, (1-\lambda_n)^{-1}\}$, so the upper bound in B.3 is $tr(H)$ which is also called the Kemeny's index of the associated Markov chain.

Proposition B.4. *For any policy π in an ergodic or unichain MDP we have $\mathsf{sp}(h^\pi) \le 2\mathsf{sp}(r^\pi)\kappa^\pi$, where h^π and r^π are respectively the bias and reward vectors induced by π and κ^π is the Kemeny's constant of the Markov chain with transition kernel P^π. The same inequality holds in weakly communicating MDPs for the optimal policy π^*, $\mathsf{sp}(h^*) \le 2\mathsf{sp}(r^*)\kappa^*$.*

In the repeated game setting, as the range of the utility function is assumed to be $[0, 1]$, $\mathsf{sp}(r^\pi) \le 1$ for all π.

Proof. (Sketch) We expand the span of bias based on $h^\pi = Z^\pi(r^\pi - g^\pi)$, and using triangle inequality and lemma B.3 conclude. The distinguishing point is that in weakly communicating MDPs the optimal gain is a constant vector which is not necessarily the case for every policy. Refer to appendix C.2 for details. □

In the self-oblivious case, by leveraging on the structure of the MDP, we can have the following upper bound on the $\mathsf{sp}(h^*)$,

Proposition B.5. *In a self-oblivious model $\mathcal{M}$ with the order of m (described in appendix (7)), for every policy π the upper bound $D(\mathcal{M}) \le m$ on the diameter holds. This implies $\mathsf{sp}(h^*) \le m\mathsf{sp}(r^*)$.*

Proof. As the transition kernel is deterministic and the state space $\mathcal{S} = \mathcal{A}^m$, the learner has full control of the stream of states he wants to visit. Thus the learner can reach any state s' from any start state s after at most m iteration by taking the actions of s'. This shows $D(\mathcal{M}) \le m$ (refer to the definition of Diameter 2.1). It is also a well-known result that $\mathsf{sp}(h^*) \le \mathsf{sp}(r^*)D(\mathcal{M})$. One can consider the above travel from s to s' as a prefix of the optimal policy starting from s' which shows that $h^*(s)$ can be at most $\mathsf{sp}(r^*)D(\mathcal{M})$ less than $h^*(s')$. □

These propositions show how the choice of ψ by the opponent affect the optimal span which contributes in the regret bounds in corollary 5.3.

B.6 PMEVI as the Base Algorithm

Since we use PMEVI as the base algorithm to have corollary 5.3, here we explain why the assumptions needed for using this algorithm are met in our game-theoretic setting. Obviously, the assumptions required for an algorithm are inherited when it is going to be used as a base algorithm of MRBEAR. For example, we could not use KLUCRL as the base algorithm if the opponent's policy ψ did not induce an ergodic environment.

PMEVI runs over the epochs and uses a modified version of Extended Value Iteration (EVI) to compute the policy that the algorithm deploys on each epoch which terminates when the number of visitation of a (s, a) pair doubles (doubling trick (Auer et al., 2008b)). PMEVI-DT requires some mild assumption on the input confidence regions $\boldsymbol{\mathcal{M}}_t$ over the transition kernel and reward function. They can be satisfied by empirical estimations of reward and transition kernel, such as the way Auer et al. (2008b) generate them. See appendix B.6.1 or (Boone and Zhang, 2024) for further explanation. In addition, it needs to know an upper bound c_{h^*} on the $\mathsf{sp}(h^*)$.

The following high probability regret bound shows that the instances of PMEVI-DTon different model classes are compatible with MRBEAR.

Proposition B.6. *(Boone and Zhang (2024), Theorem 5) Let $c_{h^*} > 0$. Assume that PMEVI-DT runs with proper confidence regions $\boldsymbol{\mathcal{M}}_t$, If $T \geq c_{h^*}^5$, then for every weakly communicating model with $\mathsf{sp}(h^*) \leq c_{h^*}$, PMEVI-DT achieves the following regret with a probability of at least $1 - 26\delta$,*

$$\mathsf{R}(\textit{PMEVI-DT}, T) \in \tilde{O}(\sqrt{(1+\mathsf{sp}(h^*))\mathsf{sp}(r)\boldsymbol{SAT}\log(T/\delta)}) + \tilde{O}(\mathsf{sp}(h^*)\mathsf{sp}(r)\boldsymbol{S}^{\frac{5}{2}}\boldsymbol{A}^{\frac{3}{2}}(1+c_0)T^{\frac{1}{4}})),$$

where c_0 is a universal constant.

The algorithm has a warm-up of $\omega = c_{h^*}^5$ iterations, where c_{h^*} is an upper bound on the $\mathsf{sp}(h^*)$. For proof refer to (Boone and Zhang, 2024) (Theorem 5 and equation 13). From Proposition B.6, we know that there exists a constant $\boldsymbol{C}$ satisfying the following inequality,

$$\mathsf{R}(\mathsf{PMEVI\text{-}DT}, T) \leq \boldsymbol{C}\sqrt{(1+\mathsf{sp}(h^*))\mathsf{sp}(r)SAT\log(T/\delta)}. \tag{5}$$

This shows that the potential regret guarantees for the model class of order i is $\mathsf{B}_i(T,\delta) = C_i\sqrt{T\log(T/\delta)}$, where $C_i = \boldsymbol{C}\sqrt{(1+\mathsf{sp}(h_i^*))\mathsf{sp}(r_i)\boldsymbol{S}_i\boldsymbol{A}_i}$.

B.6.1 Confidence Regions $\boldsymbol{\mathcal{M}}_t$ by Estimations on ψ

The algorithm PMEVI-DT, gets a system of confidence regions $\boldsymbol{\mathcal{M}}_t$ as the input, and its guarantee on the regret holds if this system satisfies some assumptions. Fix a model class,

Assumption B.7. *With probability $1-\delta$, we have $M \in \cap_{k=1}^{\mathsf{K}(T)} \boldsymbol{\mathcal{M}}_{N_{\mathsf{K}(T)}}$, where M is the tuple of the true unknown transition kernel and reward function.*

This is a common achievable assumption that is usually captured by empirical estimation of transition kernel and reward function. For further explanation refer to appendix A.2 of (Boone and Zhang, 2024). In our setting, we can empirically estimate the utility function and the distributions induced by ψ over the opponent's actions, and then by transferring from $\hat{\psi}$ to transition kernel and reward function, get an estimation that satisfies the above assumption (section 5).

Assumption B.8. *There exists a constant $C > 0$ such that for all (s,a), for all $t \leq T$, we have,*

$$\mathcal{R}_t(s,a) \subseteq \{\tilde{r}(s,a) : N_t(s,a)\|\hat{r}(s,a) - \tilde{r}(s,a)\|_1^2 \leq C\log(2SA(1+N_t(s,a))/\delta)\}$$

This assumption also emphasizes that the reward function satisfies a concentration on the difference between the optimistic bounesed estimation of reward and the empirical estimation of it. As we have a bounded utility function and discrete distribution over the opponent's action ψ this assumption can be easily satisfied in the game theoretic setting. Refer to appendix A.2.3 Boone and Zhang (2024) for an explanation of this assumption.

Assumption B.9. *For $t \geq 0$, $\boldsymbol{\mathcal{M}}_t$ is a (s,a)-rectangular convex region and $L_t^n(u)$ converges a fixed point.*

based on the estimation of ψ, the reward function and the transition kernel will be estimated and the confidence region will be the rectangular product of their confidence regions. Also the Bellman operator upon which PMEVI adds mitigation and projection, converges in typical cases including us (Boone and Zhang, 2024). In (Boone and Zhang, 2024), there is another assumption to make sure the initial set of candidate optimal bias functions includes the true optimal bias, which is not used in this work.

B.7 Modeling Untility Maximiztion in Repeated Games as Average Reward RL

In this section, we explain how utility maximization in a repeated game can be modeled as an average reward RL learning problem. Consider a two-player game G between a learner $\mathbb{A}$ and an opponent $\mathbb{B}$. The action spaces of the two players are denoted by $\mathcal{A}$ and $\mathcal{B}$ respectively, and we denote their cardinality by $\boldsymbol{A} = |\mathcal{A}|$ and $\boldsymbol{B} = |\mathcal{B}|$. Both player's are going to simultaneously play the stage game G, repeatedly for T rounds. Learner's utility function denoted by $\boldsymbol{U} : \mathcal{A} \times \mathcal{B} \to [0,1]$, given an action profile outputs a positive real value less than or equal to 1. We denote the two players' actions for time step t by A_t and B_t. The pair of A_t and B_t makes the interaction of time step t denoted by $O_t = (A_t, B_t)$. We save the history of the game in a tuple $\mathcal{O}_t = ((A_1,B_1),(A_2,B_2),\ldots,(A_t,B_t)) = (O_1,O_2,\ldots,O_t)$. A not necessarily Markov (a.k.a memory-less) policy $\boldsymbol{\pi}$ is a tuple of $(\pi_1,\ldots,\pi_T)$, where each $\pi_t : (\mathcal{A}\times\mathcal{B})^{t-1} \to \Delta_{\mathcal{A}}$ gets the history of the game and outputs a distribution over the learner's action space. The resulting distribution gives the mixed strategy played by the learner in the stage game at time step t.

Definition B.10. *A policy $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ is m-th order if $m \in \mathbb{N}$ is the* smallest *natural number such that there exists a $\pi : (\mathcal{A} \times \mathcal{B})^m \to \Delta_{\mathcal{A}}$ where, $\pi_t(o_{1:t-1}) = \pi(o_{t-m:t-1})$ for all $t \in [T]$ and $o_{1:t-1} \in (\mathcal{A} \times \mathcal{B})^{t-1}$. In other words, the output distributions of $\boldsymbol{\pi}$ only depend on the last m interactions in history, and are independent from the previous ones, and t. In this case $\boldsymbol{\pi}$ is fully representable by π.*

With slight abuse of notation, we denote $\mathbb{P}_\pi[A_t = a_t | O_{t-1:t-m} = o_{t-1:t-m}] = \pi(a_t | o_{t-1:t-m})$. The set of all m-th order policies over action space $\mathcal{A}$ is denoted by $\Pi^m_{\mathcal{A}}$. The set of all policies over $\mathcal{A}$ is given by $\Pi_{\mathcal{A}} = \bigcup_{m=0}^{\infty} \Pi^m_{\mathcal{A}}$.
Best Response. The best response (in the repeated game) against policy $\boldsymbol{\psi} \in \Pi_{\mathcal{B}}$ is computed as follows,

$$\mathsf{BR}(\boldsymbol{\psi}, s_1) = \arg\max_{\boldsymbol{\pi} \in \Pi_{\mathcal{A}}} \mathbb{E} \sum_{t=1}^{T} \boldsymbol{U}(A_t, B_t) = \arg\max_{\boldsymbol{\pi} \in \Pi_{\mathcal{A}}} V_T^{\boldsymbol{\psi}, \boldsymbol{\pi}}(s_1), \tag{6}$$

where $A_t \sim \pi_t(.|o_{1:t-1})$and $B_t \sim \psi_t(.|o_{1:t-1})$.

The Learner considers a model class $\mathcal{C}_m$ for each possible memory order of $0 \le m \le M-1$. Suppose $m \ge m^*$. All of the MDPs in $\mathcal{C}_m$ share the state space $\mathcal{S}_m = (\mathcal{A} \times \mathcal{B})^m$ with the cardinality $|\mathcal{S}_m| = \boldsymbol{S}_m$, and the action space $\mathcal{A}$, which is the set of learner's pure strategies in G.

Given a state $s_t = (o_{t-1}, \dots, o_{t-m})$ and action a_t, the reward $R_t^\psi = r(s_t, a_t) = \boldsymbol{U}(a_t, B_t)$ is obtained by applying learner's utility function to the action profile played in time-step t. As ψ is unknown, the action B_t is random, and the distribution over the rewards is also unknown. Here this point appears that not only the opponents' utility function, but even learner's utility function can be unknown to the learner, as he is going to learn the unknown reward function too. The transition dynamic induced by ψ, $P^\psi : \mathcal{S}_m \times \mathcal{A} \to \mathcal{S}_m$, is defined as follows,

$$P^\psi(s'|a, s) = \mathbb{P}[o'_{m:1} | a, o_{m:1}] = \begin{cases} \psi(b'_m | o_{m:m-m^*+1}) & \text{if } a'_m = a \ , \ o'_{m-i} = o_{m-i+1} \\ 0 & \text{otherwise} \end{cases}$$

where $o_i = (a_i, b_i)$ and $o'_i = (a'_i, b'_i)$ for all $i \in [m]$.

For m-th order policies, the very first m actions can not be taken as the history is not long enough to make the first state. We assume the first m actions, (for simplicity indexed by $a_{1-m:0}$ in negative time steps $t \in [1-m:0]$) are chosen such that the initial state s_1 is a sample from distribution μ. As we assume ψ induces a weakly communicating MDP, the initial state does not affect the performance of the algorithm. These elements collectively build the MDP of $\mathcal{M} = (\mathcal{S}_m, \mathcal{A}, P^\psi, R^\psi, T, \mu)$. We omit superscript ψ on P^ψ and R^ψ when having no change in ψ is clear from the context.

Remark B.11. *It is important to note that only for $m \ge m^*$ the opponent's policy ψ induces a well-defined* Markov *transition probability and reward function. The observed actions of player $\mathbb{B}$ coming from an m^*-th order ψ, and their resulting rewards, are not representable with any lower order model classes $m < m^*$, thus the regret guarantees of those under specified based algorithms may not hold. The corresponding base algorithms for $m \ge m^*$ are all well-specified.*

As it was clear from the best response definition (6), the value of a policy π is $V_T^{\psi,\pi}(s_1) = \mathbb{E}_{\pi,\psi} \sum_{t=1}^{T} \boldsymbol{U}(A_t, B_t)$, and is maximized by the best response policy $\mathsf{BR}(\psi, s_1)$. However, the *optimal policy* π^*_ψ is the one that maximizes $g^\pi_\psi = \lim_{T\to\infty} \frac{1}{T} V_T^{\psi,\pi}(s_1)$, and is independent of the initial state s_1 in weakly communicating MDPs (Refer to assumption 5.1). In the appendix C.5 we show the difference between the rewards obtained by $\mathsf{BR}(\psi, s_1)$ and π^*_ψ is less than $2\mathsf{sp}(h^*)$, which is negligible.

The opponent by taking a policy ψ in the order of m^* makes all $\mathsf{Alg}_{m^*} \dots \mathsf{Alg}_{M-1}$ well specified. Therefore the assumption 2.5 is satisfied. Also, proposition 5.2 implies that the optimal gain g^*_ψ is invariant in the induced MDPs of ψ in $\mathcal{C}_{m^*}$ to $\mathcal{C}_{M-1}$. Therefore the assumption 2.6 holds as well. Another important result from proposition 5.2 is that in all well-specified model classes, $i \in [m^*, M-1]$ the spans of h^*_i are the same. Thus we can use $\mathsf{sp}(h^*)$ without specifying the model class in which optimal bias is defined. Refer to appendix C.1 for the proof of previous claims.

B.8 Different Types of Opponents

Depending on ψ's input, we focus on two types of opponents:

General Opponent. When we only assume an unknown limited memory of m^* for the opponent, and ψ can be any mapping from both the learner's and opponent's actions, i.e., $\psi : (\mathcal{A} \times \mathcal{B})^{m^*} \to \Delta_{\mathcal{B}}$.

Self-Oblivious Opponent. When $\psi : \mathcal{A}^{m^*} \to \Delta_{\mathcal{B}}$ only depends on the learner's actions, it makes some specific structure on the underlying MDPs. In more detail, from proposition 5.2 we know that we can define the state space only over the learner's actions. Furthermore, unlike the general opponent, the transition probability will be known and deterministic. This is because the current state and action fully determine the next state s_{t+1} (See equation (7)). The opponent's unknown policy ψ only contributes to the reward function. Based on this crucial assumption, (Arora et al., 2018) have defined generalized equilibrium.

In this case, the state space is $\mathcal{S} = \mathcal{A}^m$, and $s_t = (a_{t-1}, \dots, a_{t-m})$. The reward $R_t = \boldsymbol{U}(A_t, B_t)$ where $B_t \sim \psi(.|S_t)$. This means that the distribution of the reward function is also unknown. The transition dynamic, is a deterministic dynamic, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, where,

$$P(s'|a, s) = \mathbb{P}[a'_{m:1}|a, a_{m:1}] = \begin{cases} 1 & \text{if } a'_m = a \ , \ a'_{m-i} = a_{m-i+1} \quad \forall i \in [m-1] \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Note that the recent action, together with the current state, deterministically implies the next state. These two different types of opponents imply different regret bounds.

Corollary B.12. *For T rounds of playing a repeated game, against an opponent with an unknown memory $m^* \leq M$, using MRBEAR with instances of PMEVI-DT as the base algorithms, gives us the following bound on the regret with a probability of at least $1 - 26MT\delta$,*

$$\mathsf{Reg}(T) \in O\big(M(\boldsymbol{A}^{m^*+1}\boldsymbol{B}^{m^*}\mathsf{sp}(h^*))^{\frac{3}{2}} \ T^{\frac{1}{2}} \log^{\frac{3}{2}}(T/\delta)\big)$$

And against self-oblivious opponents we have,

$$\mathsf{Reg}(T) \in O\big(M(\boldsymbol{A}^{m^*+1}\mathsf{sp}(h^*))^{\frac{3}{2}} \ T^{\frac{1}{2}} \log^{\frac{3}{2}}(T/\delta)\big)$$

One might also consider *Learner-Oblivious Opponent* for which all contributes in ψ is the opponent's actions, i.e., $\psi : \mathcal{B}^{m^*} \to \Delta_{\mathcal{B}}$. This turns the setting into a contextual online learning problem, because the actions taken by the learner do not affect the distribution of the opponent's actions (the context). We do not consider this case because it is not natural to assume the opponent takes action only with respect to its own previous ones, with no care about how the learner responds.

B.9 Discussion on Applications and Future Work

In online model selection, our target is the optimal design of MDPs, since either too simple or too complex models cause poor performance. In addition to the previously mentioned application and future work, this concern is shared in different techniques like state abstraction, tile coding, etc. The ideas of this work might be useful in the *online* versions of those techniques. There are a lot of applications for repeated games in the real world, such as auctions and stock markets. The result of this work can also have applications in those real-world repeated interactions.

C Missing Proofs

C.1 Proof of Proposition 5.2

Lemma C.1. *Against any m-th order policy ψ and for any policy $\pi \in \Pi_{\mathcal{A}}$, the policy $\pi' \in \bigcup_{i=0}^{m} \Pi_{\mathcal{A}}^i$ in the order of at most m, with the description of*

$$\pi'(a|o_{m:1}) = \frac{\sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(A_t = a, O_{t-1:t-m} = o_{m:1})}{\sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(O_{t-1:t-m} = o_{m:1})} \qquad \forall a \in \mathcal{A} \text{ and } o_{m:1} \in \mathcal{O}^m$$

and the same initial distribution $\mathbb{P}_{\pi,\psi}(O_{0:1-m} = o_{0:1-m}) = \mathbb{P}_{\pi',\psi}(O_{0:1-m} = o_{0:1-m})$ over the actions, satisfies,

$$\sum_{t=1}^{T} \mathbb{P}_{\pi',\psi}(O_{t-1:t-m} = o_{m:1}) = \sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(O_{t-1:t-m} = o_{m:1}) \qquad \forall o_{m:1} \in \mathcal{O}^m$$

Proof. we denote $\lambda_{\pi,\psi}(o_{m:1}) = \sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(O_{t-1:t-m} = o_{m:1})$ and $\lambda_{\pi,\psi}(a; h_{m:1}) = \sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(A_t = a, O_{t-1:t-m} = o_{m:1})$. Using this notation

$$\pi'(a|o_{m:1}) = \frac{\lambda_{\pi,\psi}(a; o_{m:1})}{\lambda_{\pi,\psi}(o_{m:1})} \qquad \forall a \in \mathcal{A} \text{ and } o_{m:1} \in \mathcal{O}^m$$

Now we write

$$\begin{aligned}
\lambda_{\pi,\psi}(o_{m:1}) &= \mathbb{P}_{\pi,\psi}(O_{0:1-m} = o_{m:1}) + \sum_{t=1}^{T-1} \mathbb{P}_{\pi,\psi}(O_{t:t-m+1} = o_{m:1}) \\
&= \mathbb{P}_{\pi,\psi}(O_{0:1-m} = o_{m:1}) + \sum_{t=m+1}^{T-1} \sum_{o_{prev} \in \mathcal{O}} \mathbb{P}_{\pi,\psi}(A_t = a_m, O_{t-1:t-m} = o_{m-1:1}, o_{prev}) \psi(b_m | o_{m-1:1}, o_{prev}) \\
&= \mathbb{P}_{\pi,\psi}(O_{0:1-m} = o_{m:1}) + \sum_{o_{prev} \in \mathcal{O}} \psi(b_m | o_{m-1:1}, o_{prev}) \lambda_{\pi,\psi}(a_m; o_{m-1:1}, o_{prev}) \\
&= \mathbb{P}_{\pi,\psi}(O_{0:1-m} = o_{m:1}) + \sum_{o_{prev} \in \mathcal{O}} \psi(b_m | o_{m-1:1}, o_{prev}) \pi'(a_m | o_{m-1:1}, o_{prev}) \lambda_{\pi,\psi}(o_{m-1:1}, o_{prev})
\end{aligned}$$

where that last equality comes from the definition of π'. We can start from $\lambda_{\pi',\psi}(h_{m:1})$, and reach the same equality, i.e.,

$$\begin{aligned}
\lambda_{\pi',\psi}(h_{m:1}) =& \mathbb{P}_{\pi',\psi}(O_{0:1-m} = o_{m:1}) \\
&+ \sum_{o_{prev} \in \mathcal{O}} \psi(b_m | o_{m-1:1}, o_{prev}) \pi'(a_m | o_{m-1:1}, o_{prev}) \lambda_{\pi',\psi}(o_{m-1:1}, o_{prev})
\end{aligned}$$

By subtracting the two results and noting that the two policies have the same initial distributions $\mathbb{P}_{\pi,\psi}(O_{0:1-m} = o_{0:1-m}) = \mathbb{P}_{\pi',\psi}(O_{0:1-m} = o_{0:1-m})$, we will have,

$$\epsilon(o_{m:1}) = \sum_{o_{prev} \in \mathcal{O}} \psi(b_m | o_{m-1:1}, o_{prev}) \pi'(a_m | o_{m-1:1}, o_{prev}) \epsilon(o_{m-1:1}, o_{prev}) \qquad \forall o_{m:1} \in \mathcal{O}^m$$

where $\epsilon(h_{m:1}) = \lambda_{\pi',\psi}(h_{m:1}) - \lambda_{\pi,\psi}(h_{m:1})$. This gives us a system of $|\mathcal{H}|^m$ linear equations which has a unique solution of $\epsilon = 0$. This implies $\lambda_{\pi',\psi}(h_{m:1}) = \lambda_{\pi,\psi}(h_{m:1})$. □

Proposition C.2. *Against any m-th order policy ψ and for any policy $\boldsymbol{\pi} \in \Pi_{\mathcal{A}}$, there exists a policy $\pi' \in \bigcup_{i=0}^{m} \Pi_{\mathcal{A}}^{i}$ in the order of at most m, where π and π' have the same value, i.e., $V_T^{\psi,\boldsymbol{\pi}}(s_1) = V_T^{\psi,\pi'}(s_1)$ for all $s_1 \in \mathcal{S}_m$ and $T \in \mathbb{N}$.*

Proof. Let $\boldsymbol{\pi} = (\pi_1, \dots, \pi_T)$ be an arbitrary policy, where $\pi_t : \mathcal{O}^{t-1} \to \Delta_{\mathcal{A}}$. We show that the policy introduced in lemma C.1 is the solution. We start by writing the value of $\boldsymbol{\pi}$,

$$\begin{aligned}
V_T^{\psi,\boldsymbol{\pi}}(s_1) &= \sum_{t=1}^{T} \sum_{a,b \in \mathcal{A} \times \mathcal{B}} \boldsymbol{U}(a,b) \mathbb{P}_{\boldsymbol{\pi},\psi}(A_t = a, B_t = b) \\
&= \sum_{a,b \in \mathcal{A} \times \mathcal{B}} \boldsymbol{U}(a,b) \sum_{t=1}^{T} \sum_{o_{t-1:1} \in \mathcal{O}^{t-1}} \mathbb{P}_{\boldsymbol{\pi},\psi}(A_t = a, B_t = b | O_{t-1:1} = o_{t-1:1}) \mathbb{P}_{\boldsymbol{\pi},\psi}(O_{t-1:1} = o_{t-1:1}) \\
&= \sum_{a,b \in \mathcal{A} \times \mathcal{B}} \boldsymbol{U}(a,b) \sum_{t=1}^{T} \sum_{o_{t-1:1} \in \mathcal{O}^{t-1}} \psi(b | o_{t-1:1}) \pi_t(a | o_{t-1:1}) \mathbb{P}_{\pi,\psi}(O_{t-1:1} = o_{t-1:1})
\end{aligned}$$

As ψ is m-th order,

$$V_T^{\psi,\boldsymbol{\pi}}(s_1) = \sum_{a,b\in\mathcal{A}\times\mathcal{B}} \boldsymbol{U}(a,b) \sum_{t=1}^{T} \sum_{o_{t-1:1}\in\mathcal{O}^{t-1}} \psi(b|o_{t-1:t-m})\pi_t(a|o_{t-1:1})\mathbb{P}_{\pi,\psi}(O_{t-1:1} = o_{t-1:1})$$

$$= \sum_{a,b\in\mathcal{A}\times\mathcal{B}} \boldsymbol{U}(a,b) \sum_{o_{m:1}\in\mathcal{O}^m} \psi(b|o_{m:1}) \sum_{t=1}^{T} \chi(a,b,o_{m:1},t)$$

where $\chi(a,b,o_{m:1},t) = \sum_{o_{t-m-1:1}\in\mathcal{O}^{t-m-1}} \pi_t(a|o_{m:1},o_{t-m-1:1})\mathbb{P}_{\pi,\psi}(O_{t-1:1} = o_{m:1},o_{t-m-1:1})$. Then we continue by writing,

$$V_T^{\psi,\boldsymbol{\pi}}(s_1) = \sum_{a,b\in\mathcal{A}\times\mathcal{B}} \boldsymbol{U}(a,b) \sum_{o_{m:1}\in\mathcal{O}^m} \psi(b|o_{m:1}) \sum_{t=1}^{T} \chi(a,b,o_{m:1},t)$$

$$= \sum_{a,b\in\mathcal{A}\times\mathcal{B}} \boldsymbol{U}(a,b) \sum_{o_{m:1}\in\mathcal{O}^m} \psi(b|o_{m:1}) \sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(A_t = a, O_{t-1:t_m} = o_{m:1})$$

$$= \sum_{a,b\in\mathcal{A}\times\mathcal{B}} \boldsymbol{U}(a,b) \sum_{o_{m:1}\in\mathcal{O}^m} \psi(b|o_{m:1})\pi'(a|o_{m:1}) \sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(O_{t-1:t-m} = o_{m:1}) = V_T^{\psi,\pi'}(s_1)$$

Note that the last equation comes from the previous lemma which states

$$\sum_{t=1}^{T} \mathbb{P}_{\pi',\psi}(O_{t-1:t-m} = o_{m:1}) = \sum_{t=1}^{T} \mathbb{P}_{\pi,\psi}(O_{t-1:t-m} = o_{m:1}) \qquad \forall o_{m:1} \in \mathcal{O}^m,$$

and we assume $\boldsymbol{\pi}$ and π' are initialized with a state coming from the initial distribution μ of MDP. □

Proposition C.3. *Against any m^*-th order policy ψ in the game setting the following property holds,*

$$g^*_{m^*} = g^*_{m^*+1} = \cdots = g^*_{M-1} \tag{8}$$

Proof. We show $g^*_{M-1} = g^*_i$ for all $i \in [m^* : M-2]$. Suppose, π^*_{M-1} is the optimal policy achieving the optimal gain of g^*_{M-1}. We know that $g^*_i \leq g^*_{M-1}$, since it is the maximum value over a smaller set. Thus we only need to show $g^*_i \geq g^*_{M-1}$. Since ψ is an m^*-th order policy, from proposition 5.2 we know that for all $i \in [m^* : M-2]$ there exists a policy π_i in the order of at most $i$corresponded to g^*_{M-1}, such that for all $T \in \mathbb{N}$, $V_T^{\psi,\pi^*_{M-1}}(s_1) = V_T^{\psi,\pi_i}(s_1)$. Therefore by taking limits we also have,

$$g^*_{M-1} = \lim_{T\to\infty} \frac{1}{T} V_T^{\psi,\pi^*_{M-1}}(s_1) = \lim_{T\to\infty} \frac{1}{T} V_T^{\psi,\pi_i}(s_1) = g_i \leq g^*_i.$$

This concludes the proof.

As we are in the game setting, we know that $\pi^*_{m^*}$ is also a valid policy in the model classes of higher orders $i \in [m^*, M-1]$, since the action space remains the same for all model classes and $\pi^*_{m^*}$ can only consider m^* many previous interactions between the learner and opponent, even if more interactions are provided in the larger model classes. This means that $\pi^*_{m^*}$ is an optimal policy for all MDPs that ψ induces in all $\mathcal{C}_{m^*}, \ldots, \mathcal{C}_{M-1}$. Therefore the rewards it gathers are the same. Thus the optimal bias of an state $h^*_i(s) := \mathbf{E}_s^{\pi^*_{m^*}}[\lim_{T\to\infty} \sum_{t=1}^{T}(R_t - g(S_t))]$, only depends on the first m^* many interactions, and therefore, $\mathsf{sp}(h^*_i)$ is invariant for $i \in [m^*, M-1]$. This allows us to denote it by $\mathsf{sp}(h^*)$. □

C.2 Proof of Upper Bound on $\mathsf{sp}(h^*)$

Proposition C.4. *For any policy π in an ergodic or unichain MDP we have,*

$$\mathsf{sp}(h^\pi) \leq 2\mathsf{sp}(r^\pi)\kappa^\pi,$$

where h^π and r^π are respectively the bias and reward vectors induced by π and κ^π is the Kemeny's constant of the Markov chain with transition kernel P^π. The same inequality holds in weakly communicating MDPs for the optimal policy π^*,

$$\mathsf{sp}(h^*) \leq 2\mathsf{sp}(r^*)\kappa^*. \tag{9}$$

Proof. Define $\bar{r}^\pi = r^\pi - g^\pi$, as $P^\pi_\infty \bar{r}^\pi = \mathbf{0}$, we have $\mathsf{sp}(r^\pi) = \mathsf{sp}(\bar{r}^\pi) \geq \max_i \bar{r}^\pi_i$. Note that in ergodic and/or unichain MDPs the gain of every policy is a constant vector in $\mathbb{R}\mathbf{1}$. Also consider $i = \arg\max_k |Z^\pi_k \bar{r}^\pi|$ and $j = \arg\max_k |Z^\pi_k \bar{r}^\pi|$ where Z^π_k is the k-th row of Z^π. Now we can write,

$$\begin{aligned}
\mathsf{sp}(h^\pi) = \mathsf{sp}(Z^\pi \bar{r}\pi) &= |\sum_{k=1}^n Z^\pi_{ik}\bar{r}^\pi_k| + |\sum_{k=1}^n Z^\pi_{jk}\bar{r}^\pi_k| \\
&\leq |\sum_{k=1}^n \bar{r}^\pi_k (Z^\pi_{ik} - Z^\pi_{jk})| \\
&\leq \mathsf{sp}(\bar{r}^\pi) \sum_{k=1}^n |Z^\pi_{ik} - Z^\pi_{jk}| \\
&\leq 2\mathsf{sp}(r^\pi)\tau_1(Z^\pi) \leq 2\mathsf{sp}(r^\pi)\kappa^\pi
\end{aligned}$$

The same can be done for the optimal policy in a weakly communicating MDP, where the optimal gain is constant. □

In the repeated game setting, since the range of the utility function is assumed to be $[0, 1]$, $\mathsf{sp}(r^\pi) \leq 1$ for all π.

C.3 Gap between V^*_T and Tg^*

Lemma C.5. *For a weakly communicating MDP, we have,*

$$\|V^*_T - Tg^*\|_\infty \leq 2\|h^*\|_\infty$$

Proof. We know $V^*_T = L^T\mathbf{0}$, where L is the optimal bellman operator and $\mathbf{0}$ is the vector full of zeros. Also we know $g^* + h^* = Lh^*$ and, because the MDP is weakly communicating, the g^* is a constant vector, which means $L^2h^* = L(g^* + h^*) = g^* + L(h^*) = 2g^* + h^*$. By induction you will get $L^Th^* = Tg^* + h^*$ (Puterman, 2014). Therefore we can write,

$$\begin{aligned}
\|V^*_T - Tg^*\|_\infty &= \|L^T\mathbf{0} - L^Th^* + h^*\|_\infty \\
&\leq \|L^T\mathbf{0} - L^Th^*\|_\infty + \|h^*\|_\infty \\
&\leq 2\|h^*\|_\infty
\end{aligned}$$

where the last inequality holds because L is a non-expansive operator. □

C.4 Upper bound on $\mathsf{K}(T)$ Under Doubling Trick

Lemma C.6. *Auer et al. (2008b) The number of epochs up to time $T \geq SA$ under doubling trick, is upper bounded by,*

$$\mathsf{K}(T) \leq SA\log_2(\frac{8T}{SA})$$

C.5 $\mathcal{G}$ is Highly Probable

Lemma C.7. $\mathbb{P}(\mathcal{G}) \geq 1 - (M - m^*)(\mathsf{K}(T) + 1)\delta$, *where*

$$\mathcal{G} = \{\forall i \in [m^* : M - 1], k \in [1 : \mathsf{K}(T) + 1] : \mathsf{Reg}_i(\mathsf{Alg}_i, N_{i,k}) \leq C_i\sqrt{N_{i,k}\log(\frac{N_{i,k}}{\delta})}\}$$

Proof. Define the following events for all $i \in [m^* : M-1]$ and $k \in [1 : \mathsf{K}(T)+1]$,

$$\mathcal{G}_{i,k} = \{\mathsf{R}_i(\mathsf{Alg}_i, N_{i,k}) \leq \ C_i\sqrt{N_{i,k}\log(\frac{N_{i,k}}{\delta})}\}$$

As $N_{i,k} \geq \omega_i$ for all $1 \leq k \leq \mathsf{K}(T)$, and i lies in the well-specified model classes, from 3.1, we know

$$\mathbb{P}(\mathcal{G}_{i,k}) \geq 1-\delta.$$

Applying union bound,

$$\mathbb{P}(\mathcal{G}^c) = \mathbb{P}(\cup_{i,k}\mathcal{G}^c_{i,k}) \leq \sum_{i,k}\mathbb{P}(\mathcal{G}^c_{i,k}) \leq (M-m^*)(\mathsf{K}(T)+1)\delta,$$

concludes the proof. □

C.6 Regret is Multiplicatively Balanced

Lemma C.8. *By running algorithm MRBEAR with compatible base algorithms for all $0<\delta<1$, $k \in \mathbb{N}$ and any pair of $i, j \in \mathcal{I}_k$ where $i \neq j$, the followings hold,*

1. $$\mathsf{B}_i(N_{i,k},\delta) \leq \mathsf{B}_j(N_{j,k},\delta) + \alpha_i\mathsf{B}_i(N_{i,k-1},\delta) + \beta$$

2. $$\frac{N_{i,k}}{N_{j,k}} \leq \big(\frac{C_j}{(1-\alpha_i)C_i} + \frac{\beta}{(1-\alpha_i)C_i\sqrt{N_{j,k}\log(N_{j,k}/\delta)}}\big)^2\log(\frac{N_{j,k}}{\delta}) \tag{10}$$

where $1/2 \leq \alpha_i = \frac{\log(\omega_i \vee 9)+1}{2\log(\omega_i\vee 9)} \leq \frac{3}{4}$ and $\beta = \mathsf{B}_{M-1}(\omega_{M-1},\delta) - \mathsf{B}_0(\omega_0,\delta)$.

Proof. Based on 3.1 we have defined $\mathsf{B}_i(N_{i,k},\delta) = C_i\sqrt{N_{i,k}\log(\frac{N_{i,k}}{\delta})}$, therefore we have,

$$\begin{aligned}\mathsf{B}_i(N_{i,k},\delta) - \mathsf{B}_i(N_{i,k-1},\delta) &\leq \mathsf{B}_i(2N_{i,k-1},\delta) - \mathsf{B}_i(N_{i,k-1},\delta)\\ &\leq \frac{1}{2}C_i\big(\sqrt{N_{i,k-1}\log(\frac{N_{i,k-1}}{\delta})} + \sqrt{\frac{N_{i,k-1}\delta^2}{\log(N_{i,k-1}/\delta)}}\big)\end{aligned}$$

where the second inequality holds because $\mathsf{B}_i(t,\delta)$ is increasing and concave in t, and the first one is due to the doubling trick ($N_{i,k} \leq 2N_{i,k-1}$). Then since $\delta \leq 1$ and $N_{i,k} \geq \omega_i \vee 9$ we have (recall ω_i is the warm up iterations of Alg_i),

$$\begin{aligned}\mathsf{B}_i(N_{i,k},\delta) - \mathsf{B}_i(N_{i,k-1},\delta) &\leq \frac{1}{2}C_i\big(\sqrt{N_{i,k-1}\log(\frac{N_{i,k-1}}{\delta})} + \sqrt{\frac{N_{i,k-1}\delta^2}{\log(N_{i,k-1}/\delta)}}\big) &(11)\\ &\leq (\frac{1}{2} + \frac{1}{2\log(\omega_i\vee 9)})\mathsf{B}_i(N_{i,k-1},\delta) &(12)\\ &= \alpha_i\mathsf{B}_i(N_{i,k-1},\delta) &(13)\end{aligned}$$

Before showing the main statement, note $\mathsf{B}_i(t,\delta)$ is increasing in $i \in \mathbb{N}\cup\{0\}$ and $t \in \mathbb{N}$. Also, for reducing redundancy, we temporarily drop δ and just write $\mathsf{B}_i(t)$. We show the lemma statement by induction on k: For $k=1$, note that $N_{m,0} = \omega_m$ for all $m \in [0 : M-1]$. Suppose there exists a pair of $i, j \in \mathcal{I}_1$ such that $\mathsf{B}_i(N_{i,1}) > \mathsf{B}_j(N_{j,1}) + \alpha_i\mathsf{B}_i(\omega_i) + \beta$, and we want to reach a contradiction. This means that the algorithm has picked model class i in epoch k, i.e. $i_k = i$, since if this would not be the case, $N_{i,1} = N_{i,0} = \omega_i$ and the inequality

of the lemma should have been satisfied (Note the definition of β). Therefore, $\mathsf{B}_i(N_{i,1}) > \mathsf{B}_j(N_{j,1}) + \alpha_i \mathsf{B}_i(\omega_i) + \beta$ implies $i_1 = i$. Now we can write,

$$\begin{aligned}\mathsf{B}_i(\omega_i) &\geq \mathsf{B}_i(N_{i,1}) - \alpha_i \mathsf{B}_i(\omega_i) \\ &> \mathsf{B}_j(N_{j,1}) + \beta \\ &= \mathsf{B}_j(\omega_j) + \beta \\ &\geq \mathsf{B}_j(\omega_j)\end{aligned}$$

which contradicts with $i = \arg\min_{m \in \mathcal{I}_1} \mathsf{B}_m(\omega_m)$. The first inequality is from 11, the strict inequality is from the contradiction assumption, and the equality is for $j \neq i = i_1$.

Induction step: Again for the sake of contradiction, suppose k is the first epoch in which the lemma's inequality violates, i.e.

$$\mathsf{B}_i(N_{i,k}) > \mathsf{B}_j(N_{j,k}) + \alpha_i \mathsf{B}_i(N_{i,k-1}) + \beta.$$

For a similar reason as before, this implies that $i_k = i$. And by a similar chain of inequalities we get to a contradiction with $i = \arg\min_{m \in \mathcal{I}_k} \mathsf{B}_m(N_{m,k-1})$,

$$\begin{aligned}\mathsf{B}_i(N_{i,k-1}) &\geq \mathsf{B}_i(N_{i,k}) - \alpha_i \mathsf{B}_i(N_{i,k-1}) \\ &> \mathsf{B}_j(N_{j,k}) + \beta \\ &= \mathsf{B}_j(N_{j,k-1}) + \beta \\ &\geq \mathsf{B}_j(N_{j,k-1}).\end{aligned}$$

This concludes the proof for the first part. Now we move on to the second statement. From the first part, we have,

$$\mathsf{B}_i(N_{i,k}) \leq \mathsf{B}_j(N_{j,k}) + \alpha_i \mathsf{B}_i(N_{i,k-1}) + \beta,$$

and by replacing $\mathsf{B}_i(N_{i,k-1})$ with $\mathsf{B}_i(N_{i,k})$, the following holds,

$$\mathsf{B}_i(N_{i,k}) \leq \frac{1}{1-\alpha_i} \mathsf{B}_j(N_{j,k}) + \frac{\beta}{1-\alpha_i}. \tag{14}$$

Note that $\frac{1}{2} \leq \alpha_i \leq \frac{3}{4}$ for all $i \in [0, ..., M-1]$. Replacing $\mathsf{B}_i(N_{i,k}, \delta) = C_i \sqrt{N_{i,k} \log(\frac{N_{i,k}}{\delta})}$ gives us,

$$C_i \sqrt{N_{i,k} \log(N_{i,k}/\delta)} \leq \frac{1}{1-\alpha_i} C_j \sqrt{N_{j,k} \log(N_{j,k}/\delta)} + \frac{\beta}{1-\alpha_i}$$

Now by rearranging the terms we have,

$$\frac{\sqrt{N_{i,k} \log(N_{i,k}/\delta)}}{\sqrt{N_{j,k} \log(N_{j,k}/\delta)}} \leq \frac{C_j}{(1-\alpha_i)C_i} + \frac{\beta}{(1-\alpha_i)C_i \sqrt{N_{j,k} \log(N_{j,k}/\delta)}}$$

By squaring both sides and noting that $\log(N_{i,k}/\delta) \geq 1$, we reach the second statement. □

C.7 Reg_i Upper Bound Based on $\frac{N_{i,k-1}}{N_{m^*,k-1}}$ for Misspecified $i < m^*$

Lemma C.9. *For any active model class $i \in \mathcal{I}_k$ such that $i < m^*$, under the event $\mathcal{G}$, the regret of model class i is bounded as follows,*

$$\mathsf{Reg}_i(N_{i,k-1}) \leq \left(\frac{N_{i,k-1}}{N_{m^*,k-1}} + \frac{1}{1-\alpha_i}\right) \mathsf{B}_{m^*}(N_{m^*,k-1}) + \frac{2N_{i,k-1}}{N_{m^*,k-1}} c_{h^*} + \frac{\beta}{1-\alpha_i}. \tag{15}$$

Proof. When $i \in \mathcal{I}_k$, it should have passed the miss specification test. Thus as $m^* > i$, the following holds

$$\frac{\mathsf{B}_i(N_{i,k-1}, \delta) + \sum_{t \in \mathcal{N}_{i,k-1}} r_t}{N_{i,k-1}} \geq \frac{\sum_{t \in \mathcal{N}_{m^*,k-1}} r_t - 2c_{h^*}}{N_{m^*,k-1}}.$$

We temporarily omit δ as this input does not change. Subtracting $g^\star$ from both sides gives us,

$$\frac{\mathsf{B}_i(N_{i,k-1})}{N_{i,k-1}} - \frac{\mathsf{Reg}_i(N_{i,k-1})}{N_{i,k-1}} \geq -\left(\frac{\mathsf{Reg}_{m^*}(N_{m^*,k-1}) + 2c_{h^*}}{N_{m^*,k-1}}\right)$$

by rearranging the terms we have,

$$\mathsf{Reg}_i(N_{i,k-1}) \leq \mathsf{B}_i(N_{i,k-1}) + \frac{N_{i,k-1}}{N_{m^*,k-1}}(\mathsf{Reg}_{m^*}(N_{m^*,k-1}) + 2c_{h^*}) \tag{16}$$

$$\leq \mathsf{B}_i(N_{i,k-1}) + \frac{N_{i,k-1}}{N_{m^*,k-1}}(\mathsf{B}_{m^*}(N_{m^*,k-1}) + 2c_{h^*}) \tag{17}$$

where the second inequality is implied as m^* is well specified 5.2. From lemma C.8, for $i \neq j \in \mathcal{I}_{k-1}$, we have,

$$\mathsf{B}_i(N_{i,k-1}) \leq \frac{1}{1-\alpha_i}\mathsf{B}_j(N_{j,k-1}) + \frac{\beta}{1-\alpha_i}. \tag{18}$$

Note that $\frac{1}{2} \leq \alpha_i \leq \frac{3}{4}$. Obviously, when i is in $\mathcal{I}_k$ it had been also in $\mathcal{I}_{k-1}$. Also under $\mathcal{G}$ we know all the wellspecified model classes, including m^* remain active. So we can apply the above bound with $j = m^*$ on 16 and get the lemmas claim,

$$\mathsf{Reg}_i(N_{i,k-1}) \leq \left(\frac{N_{i,k-1}}{N_{m^*,k-1}} + \frac{1}{1-\alpha_i}\right)\mathsf{B}_{m^*}(N_{m^*,k-1}) + \frac{2N_{i,k-1}}{N_{m^*,k-1}}c_{h^*} + \frac{\beta}{1-\alpha_i}.$$

□

C.8 Final Bound Reg_i for Misspecified $i < m^*$

Lemma C.10. *For any active model class $i \in \mathcal{I}_{\mathsf{K}(T)+1}$ such that $i < m^*$, under the event $\mathcal{G}$, the regret of model class i is bounded as follows,*

$$\mathsf{Reg}_i(N_{i,\mathsf{K}(T)}) \leq \frac{C_{m^*}^3\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2} + \frac{1}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + O(\log^{\frac{3}{2}}(T/\delta)) \tag{19}$$

Proof. We continue on the result of C.9 using 10. Consider

$$\mathsf{Reg}_i(N_{i,\mathsf{K}(T)}) \leq \left(\frac{N_{i,\mathsf{K}(T)}}{N_{m^*,\mathsf{K}(T)}} + \frac{1}{1-\alpha_i}\right)\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \frac{2N_{i,\mathsf{K}(T)}}{N_{m^*,\mathsf{K}(T)}}c_{h^*} + \frac{\beta}{1-\alpha_i}, \tag{20}$$

and we will bind it term by term. Denote $\gamma_i = \left(\frac{C_{m^*}}{(1-\alpha_i)C_i} + \frac{\beta}{(1-\alpha_i)C_i\sqrt{N_{m^*,\mathsf{K}(T)}\log(N_{m^*,\mathsf{K}(T)}/\delta)}}\right) = \frac{\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})+\beta}{(1-\alpha_i)C_i\sqrt{N_{m^*,\mathsf{K}(T)}\log(N_{m^*,\mathsf{K}(T)}/\delta)}}$. We write,

$$\frac{N_{i,\mathsf{K}(T)}}{N_{m^*,\mathsf{K}(T)}}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) \leq \gamma_i^2\log(N_{m^*,\mathsf{K}(T)}/\delta)\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) \tag{21}$$

$$= \frac{(\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \beta)^2\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})}{(1-\alpha_i)^2C_i^2N_{m^*,\mathsf{K}(T)}} \tag{22}$$

$$= \frac{C_{m^*}(\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \beta)^2\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}}, \tag{23}$$

where the first inequality is due to 10, and the further equalities are from the definition of B_{m^*}. Now we inspect the last term,

$$\begin{aligned}\frac{C_{m^*}(\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \beta)^2\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} &= \frac{C_{m^*}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})^2\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} \quad (:= \chi_1)\\ &+ \frac{2C_{m^*}\beta\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} \quad (:= \chi_2)\\ &+ \frac{C_{m^*}\beta^2\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} \quad (:= \chi_3)\end{aligned}$$

For the first term we have,

$$\chi_1 = \frac{C_{m^*}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})^2\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} = \frac{C_{m^*}^3\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2}.$$

For the second term, note that $N_{i,\mathsf{K}(T)} \le T$ for all i, so

$$\chi_2 = \frac{2C_{m^*}\beta\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} = \frac{2C_{m^*}^2\beta\log^{3/2}(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2} \in O(\log^{\frac{3}{2}}(T/\delta))$$

and for the third term,

$$\chi_3 = \frac{C_{m^*}\beta^2\log(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2\sqrt{N_{m^*,\mathsf{K}(T)}}} \in o(1)$$

Therefore, χ_1 is the dominant term and we can wrap up this part by saying,

$$\frac{N_{i,\mathsf{K}(T)}}{N_{m^*,\mathsf{K}(T)}}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) \le \frac{C_{m^*}^3\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2} + O(\log^{\frac{3}{2}}(T/\delta)). \tag{24}$$

Now we move on to the other terms of 20. We keep the second term as it is $\frac{\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)})}{1-\alpha_i}$. With expanding $\gamma_i^2 = \big(\frac{C_{m^*}}{(1-\alpha_i)C_i} + \frac{\beta}{(1-\alpha_i)C_i\sqrt{N_{m^*,\mathsf{K}(T)}\log(N_{m^*,\mathsf{K}(T)}/\delta)}}\big)^2$, and 10, it is easy to show that,

$$\frac{2N_{i,\mathsf{K}(T)}}{N_{m^*,\mathsf{K}(T)}}c_{h^*} \le 2\gamma_i^2\log(N_{m^*,\mathsf{K}(T)}/\delta)c_{h^*} \in O(\log(T/\delta)) \tag{25}$$

And finally we know,

$$\frac{\beta}{1-\alpha_i} \in O(\log(1/\delta)), \tag{26}$$

even if we put $\delta = T^\eta$ for some η it will be in $O(\log(T))$. By gathering all parts 24,25, and 26 together, we have,

$$\mathsf{Reg}_i(N_{i,\mathsf{K}(T)}) \le \frac{C_{m^*}^3\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2C_i^2} + \frac{1}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + O(\log^{\frac{3}{2}}(T/\delta)),$$

which concludes the proof. □

C.9 Proof of Theorem 4.1 (Main Theorem)

Theorem C.11 (Main theorem)**.** *By running the algorithm MRBEAR over M compatible base algorithms on model classes $\mathcal{C}_0$ to $\mathcal{C}_{M-1}$, and the unknown optimal model class $\mathcal{C}_{m^*}$, and known upper bound of $c_{h^*} \ge \max_i \mathsf{sp}(h_i^*)$, for $T \ge \sum_i \omega_i$ and all $0 < \delta < 1$, with probability of at least $1 - MT\delta$, the regret (2) is upper bounded by*

$$\mathsf{Reg}(\mathit{MRBEAR}, T) \le \left(\frac{16m^*C_{m^*}^2\log^{\frac{3}{2}}(T/\delta)}{C_0^2} + 4M\right)\mathsf{B}_{m^*}(T,\delta) + O(\log^{\frac{3}{2}}(T/\delta)).$$

Proof. After proving the lemmas, we are now ready to prove the main theorem 4.1. We start with decomposing the regret into the regrets of each model class (refer to 3),

$$\begin{aligned}
\mathsf{Reg}(\mathsf{MRBEAR}, T) &:= Tg_{m^*}^* - \sum_{t=1}^{T} R_t \\
&= \sum_{i=0}^{M-1}\Big[N_{i,\mathsf{K}(T)}\ g^\star - \sum_{t\in\mathcal{N}_{i,\mathsf{K}(T)}} R_t\Big] \\
&= \sum_{i=0}^{M-1}\mathsf{Reg}_i(\mathsf{Alg}_i, N_{i,\mathsf{K}(T)}).
\end{aligned}$$

Suppose we are under event $\mathcal{G}$ which happens with the probability of at least $1-(M-m^*)\mathsf{K}(T)\delta$. For all $i \in \mathcal{I}_{K(T)+1}$ such that $i < m^*$ and , we can use C.10 and write,

$$\begin{aligned}
&\sum_{i=0}^{m^*-1} \mathsf{Reg}_i(\mathsf{Alg}_i, N_{i,\mathsf{K}(T)}) \\
&\leq \sum_{i=0}^{m^*-1} \Big(\frac{C^3_{m^*}\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2 C_i^2} + \frac{1}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + O(\log^{\frac{3}{2}}(T/\delta)) \Big) \\
&\leq \frac{m^* C^3_{m^*}\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2 C_0^2} + \frac{m^*}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + O(\log^{\frac{3}{2}}(T/\delta)).
\end{aligned}$$

Note that $C_0 \leq C_1 \leq \dots C_{M-1}$ as the models get richer by increasing i, i.e. their regret guarantees increase. For the well-specified model classes $i \geq m^*$ we use their regret guarantees, and write,

$$\begin{aligned}
\sum_{i=m^*}^{M-1} \mathsf{Reg}_i(\mathsf{Alg}_i, N_{i,\mathsf{K}(T)}) &\leq \sum_{i=m^*}^{M-1} \mathsf{B}_i(N_{i,\mathsf{K}(T)}, \delta) \\
&\leq \mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \sum_{i=m^*+1}^{M-1} \big[\frac{1}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \frac{\beta}{1-\alpha_i}\big] \\
&\leq \frac{M-m^*}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}) + \frac{(M-m^*-1)\beta}{1-\alpha_i},
\end{aligned}$$

where the first and second inequalities are implied from lemma C.8 (specifically 14), and the last inequality is from $\frac{1}{(1-\alpha_i)} \geq 1$ as $1/2 \leq \alpha_i \leq 3/4$ for all $i \in [0, \dots, M-1]$. Note that we are exploiting on the fact that visiting other model classes between the epochs of Alg_i on $\mathcal{C}_i$ does not affect on the regret bound guarantees (refer to 3.2). This is clear from the proof and procedure of most of the algorithms, including the state of the art PMEVI-DT Boone and Zhang (2024). Also, note that $\frac{(M-m^*-1)\beta}{1-\alpha_i} \in O(\log(1/\delta))$ which by choice of $\delta = T^\eta$ for some η it will be in $O(\log(T))$. So we can put the parts together and have,

$$\begin{aligned}
\mathsf{Reg}(\mathsf{MRBEAR}, T) \leq &\frac{m^* C^3_{m^*}\sqrt{N_{m^*,\mathsf{K}(T)}}\log^2(N_{m^*,\mathsf{K}(T)}/\delta)}{(1-\alpha_i)^2 C_0^2} + \frac{M}{1-\alpha_i}\mathsf{B}_{m^*}(N_{m^*,\mathsf{K}(T)}, \delta) \\
&+ O(\log^{\frac{3}{2}}(T/\delta))
\end{aligned}$$

which by applying $N_{m^*,\mathsf{K}(T)} \leq T$ and changing $\mathsf{K}(T) \leq T$ concludes the proof. □

C.10 Proof of Lower Bound

The MDPs constructed in the repeated game setting have a special structure that prevents us from directly using previous lower bounds designed for generic MDPs (Auer et al., 2008b; Osband and Van Roy, 2016). This is because they propose hard-to-learn MDPs that do not obey the structure of our setting. In this section, we present a minimax lower bound on the regret of any algorithm, using Le Cam method. First, we give a divergence decomposition lemma, which is similar to the the key lemma in Le Cam-based lower bounds in bandit literature.

Lemma C.12. *(Divergence Decomposition) Fix an algorithm Alg. Let ψ and ψ' be two opponent's policies and $\mathbf{P}_\psi$ (resp. $\mathbf{P}_{\psi'}$) be the probability measures on the trajectories of learner and opponent actions, induced by the interaction of Alg with ψ (resp. ψ') for T rounds. Then,*

$$D_{KL}(\mathbf{P}_\psi \| \mathbf{P}_{\psi'}) = \sum_{s\in\mathcal{S}} \lambda_\psi(s) D_{KL}(\psi(s) \| \psi'(s)) \tag{27}$$

where $\psi(s) \in \Delta_{\mathcal{B}}$ is the distribution from which ψ takes action in state s, and$\lambda_\psi(s) = \sum_{t=1}^T \mathbb{P}_\psi[S_t = s]$ is the occupancy measure induced over the states by ψ and Alg after T rounds.

Proof. Assume that for $s \in \mathcal{S}$ we have $D_{KL}(\psi(s)\|\psi'(s)) < \infty$. The algorithm *Alg* implements a memory-full

policy π_t at round t. We can write

$$\begin{aligned}\mathbf{P}_\psi(a_1, b_1, \ldots, a_T, b_T) &= \Pi_{t=1}^T \pi_t(a_t|a_1, b_1, \ldots, a_{t-1}, b_{t-1})\mathbb{P}_\psi(b_t|a_1, b_1, \ldots, a_{t-1}, b_{t-1}) \\ &= \Pi_{t=1}^T \pi_t(a_t|a_1, b_1, \ldots, a_{t-1}, b_{t-1})\psi(b_t|s_t)\end{aligned}$$

By chain rull for Radon-Nikodym derivatives we have,

$$\log \frac{d\mathbf{P}_\psi}{d\mathbf{P}_{\psi'}}(a_1, b_1, \ldots, a_T, b_T) = \sum_{t=1}^T \log \frac{\psi(b_t|s_t)}{\psi'(b_t|s_t)}$$

And by taking expectations,

$$\mathbb{E}_\psi[\log \frac{d\mathbf{P}_\psi}{d\mathbf{P}_{\psi'}}(A_1, B_1, \ldots, A_T, B_T)] = \sum_{t=1}^T \mathbb{E}_\psi[\log \frac{\psi(B_t|S_t)}{\psi'(B_t|S_t)}],$$

because b_t is only be determined by ψ, the distribution of B_t under $\mathbb{P}_\psi(.|S_t)$ is the same as $\psi(.|S_t)$. This together with the tower rule gives us,

$$\mathbb{E}_\psi[\log \frac{\psi(B_t|S_t)}{\psi'(B_t|S_t)}] = \mathbb{E}_\psi\big[\mathbb{E}_\psi[\log \frac{\psi(B_t|s_t)}{\psi'(B_t|s_t)}|S_t = s_t]\big] = \mathbb{E}_\psi[D_{KL}(\psi(S_t)\|\psi'(S_t))].$$

Now we can write,

$$\begin{aligned}D_{KL}(\mathbf{P}_\psi\|\mathbf{P}_{\psi'}) &= \mathbb{E}_\psi[\log \frac{d\mathbf{P}_\psi}{d\mathbf{P}_{\psi'}}(A_1, B_1, \ldots, A_T, B_T)] \\ &= \sum_{t=1}^T \mathbb{E}_\psi[D_{KL}(\psi(S_t)\|\psi'(S_t))] \\ &= \sum_{t=1}^T \mathbb{E}_\psi[\sum_{s\in\mathcal{S}} \mathbf{1}\{S_t = s\}D_{KL}(\psi(s)\|\psi'(s))] \\ &= \sum_{s\in\mathcal{S}} \lambda_\psi(s)D_{KL}(\psi(s)\|\psi'(s)).\end{aligned}$$

When $s \in \mathcal{S}$ is occupied by ψ with a positive measure and $\log \frac{\psi(b|s)}{\psi'(b|s)}$ is infinite for some $b \in \mathcal{B}$, then there a trajectory with a positive probability that makes $\log \frac{d\mathbf{P}_\psi}{d\mathbf{P}_{\psi'}}(a_1, b_1, \ldots, a_T, b_T)$ also infinite, meaning the lemma holds for the case of infinity as well.

□

We also need the following lemma before designing the opponent's policies.

Lemma C.13. *For two distribution vectors P and Q in $\Delta_{\boldsymbol{B}}$, such that*

$$\begin{cases} P_1 = P_2 = 1/2 \\ P_i = 0 \quad \forall i \in [3 : \boldsymbol{B}] \end{cases}$$

and

$$\begin{cases} Q_1 = 1/2 - 2\epsilon \\ Q_2 = 1/2 + 2\epsilon \\ Q_i = 0 \quad \forall i \in [3 : \boldsymbol{B}] \end{cases}$$

$D_{KL}(P\|Q) = \frac{1}{2}(\log(\frac{1}{1-4\epsilon}) + \log(\frac{1}{1+4\epsilon})) = 8\epsilon^2 + c\epsilon^4$ *for* $\epsilon \in (0, 1/4)$ *and some constant* c.

Proof. From the definition of Kullback-Leibler divergence we have,

$$D_{KL}(P\|Q) = \sum_i P_i \log(\frac{P_i}{Q_i}) = \frac{1}{2}(\log(\frac{1}{1-4\epsilon}) + \log(\frac{1}{1+4\epsilon})),$$

since the only non-zero terms are the first and the second terms. Now write the Taylor expansion for the two terms of $\log(\frac{1}{1-4\epsilon})$ and $\log(\frac{1}{1+4\epsilon})$ at point $\epsilon = 0$,

$$\log(\frac{1}{1-4\epsilon}) = \epsilon[\frac{4}{1-4\epsilon}]_{\epsilon=0} + \frac{\epsilon^2}{2!}[\frac{16}{(1-4\epsilon)^2}]_{\epsilon=0} + \frac{\epsilon^3}{3!}[\frac{128}{(1-4\epsilon)^3}]_{\epsilon=0} + c\epsilon^4,$$

and

$$\log(\frac{1}{1+4\epsilon}) = \epsilon[\frac{-4}{1+4\epsilon}]_{\epsilon=0} + \frac{\epsilon^2}{2!}[\frac{16}{(1+4\epsilon)^2}]_{\epsilon=0} + \frac{\epsilon^3}{3!}[\frac{-128}{(1+4\epsilon)^3}]_{\epsilon=0} + c\epsilon^4.$$

Therefore by summing the two parts, we conclude the statement in the lemma. □

Now we need to define special sequences that will be of use in designing communicating opponent's policies. These special series are called de Bruijn sequences (de Bruijn, 1975; Crochemore et al., 2021).

Definition C.14. *The sequence $b_{\boldsymbol{B}^m} \dots b_2 b_1$ is called an m-th order* de Bruijn *sequence on alphabet $\mathcal{B}$ with the cardinality of $\boldsymbol{B} = |\mathcal{B}|$, if it contains every ordered tuple of actions in length m. In other words,*

$$\forall\ (b^m, b^{m-1}, \dots, b^1) \in \mathcal{B}^m \quad \exists t \in [\boldsymbol{B}^m] : \quad (b^m, b^{m-1}, \dots, b^1) = (b_t, b_{t-1}, \dots, b_{t-m+1})$$

where the indexes are cyclic.

This is a sequence that visits all the permutations of length m from alphabet $\mathcal{B}$ without any repetition. As an example, for $\mathcal{B} = \{0, 1\}$ and $m = 2$, the sequence 0110 is a 2nd-order de Bruijn sequence. Note that the sequence is cyclic. These sequences exist for all order m and alphabet size $\boldsymbol{B}$ de Bruijn (1975); Crochemore et al. (2021).

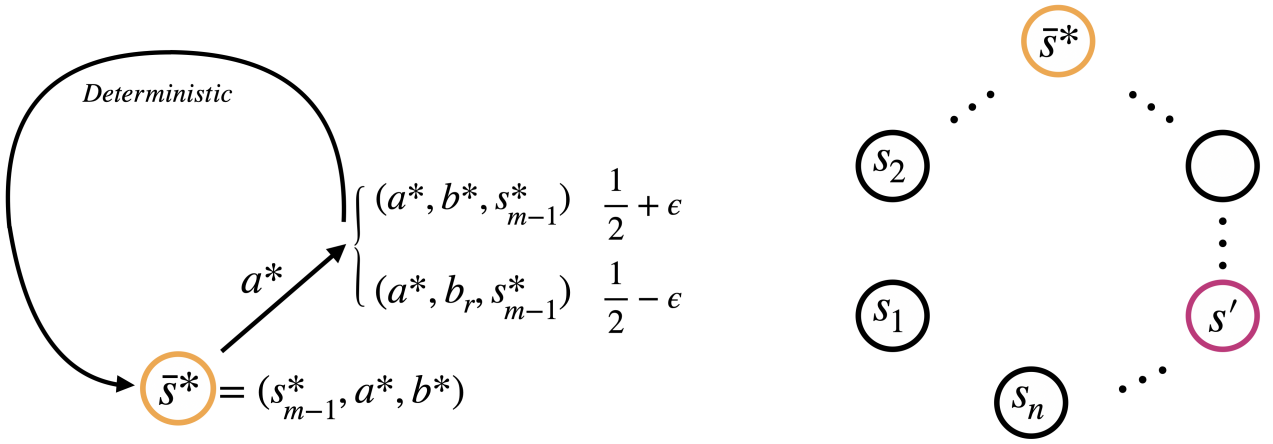
C.10.1 Designing the Game and ψ

We design two complementary opponent policies ψ and ψ' that force the algorithm to have high regret in at least one of them. Note that this is for a fixed memory order m known to the learner, which even gives an advantage to him.

Theorem C.15. *Suppose the number of opponent's actions $|\mathcal{B}| \geq 3$ and number of learner's actions $|\mathcal{A}| \geq 2$. For any fixed memory $m \geq 2$ known for the learner, and any algorithm Alg, there exists a stage game with utility $U : \mathcal{A} \times \mathcal{B} \to [0, 1]$, and a general opponent's policies ψ_{Gen} such that*

$$\mathsf{R}_{\psi_{Gen}}(Alg, T) \in \Omega(\frac{1}{m}\sqrt{\boldsymbol{A}^{m-1}\boldsymbol{B}^{m-1}T}).$$

Proof. Consider the set of learner's actions $\mathcal{A} = \{a_1, \dots, a_{\boldsymbol{A}-1}\} \cup \{a^*\}$ and the set of opponent's actions $\mathcal{B} = \{b_1, \dots, b_{\boldsymbol{B}-2}\} \cup \{b^*, b_r\}$. Also consider a utility function for the learner such that $\boldsymbol{U}(a^*, b^*) = 1$ for an special learner's action a^* and an special opponent's action b^*, and $\boldsymbol{U}(a, b) = 0$ for all other entries, i.e. $a \neq a^*$ or $b \neq b^*$. Therefore the utility matrix has only one entry with the reward of 1 and 0 every where else. Fix a de Bruijn sequence $\mathfrak{b}_{\boldsymbol{B}-2^{(m-1)}} \dots \mathfrak{b}_2 \mathfrak{b}_1$ over the alphabet $\{b_1, \dots, b_{\boldsymbol{B}-2}\}$ in the order of $m-1$. We define a special $m-1$-th order prefix $s^*_{m-1} = (a^*_m, b^*_m, \dots, a^*_2, b^*_2) \in (\mathcal{A}\backslash\{a^*\} \times \mathcal{B}\backslash\{b^*, b_r\})^{m-1}$. This prefix, together with the last actions of (a^*, b^*) make a special rewarding state of $\bar{s}^* = (s^*_{m-1}, a^*, b^*)$. Define $\mathsf{dB}^{+1} : \mathcal{S}_{m-1} \to \mathcal{B}$ to be a function that gets an $m-1$-th order state $s = (a_m, b_m, \dots, a_2, b_2)$ as an input, then search for the sequence of $(b_m, \dots, b_2)$ in the de Bruijn sequence $\mathfrak{b}_{\boldsymbol{B}-2^{(m-1)}} \dots \mathfrak{b}_2 \mathfrak{b}_1$, and outputs the next character in the de Bruijn sequence. i.e. if $(\mathfrak{b}_{t+m} \dots \mathfrak{b}_{t+2}) = (b_m, \dots, b_2)$, then $\mathsf{dB}^{+1}(s) = \mathfrak{b}_{t+m+1}$. This is an increment operator on the de Bruijn sequence.



(a) (b)

Figure 4: (a) The circular rewarding process for the optimal policy. (b) The set of states making a haystack of size $n = ((A-1)(B-2))^{m-1}$, with two needles of $\bar{s}^*$ and s'.

We are now ready to define the following opponent's policy ψ,

$$\begin{cases}
\psi(b^*|(s^*,a,b)) = 1/2+\epsilon \qquad \forall a,b \in \mathcal{A}\backslash\{a^*\}\times\mathcal{B} \\
\psi(\mathsf{dB}^{+1}(s^*)|(s^*,a,b)) = 1/2-\epsilon \qquad \forall a,b \in \mathcal{A}\backslash\{a^*\}\times\mathcal{B} \\
\psi(b^*|(s,a,b)) = 1/2 \qquad \forall a,b \in \mathcal{A}\backslash\{a^*\}\times\mathcal{B}\ ,\ \forall s\neq s^* \text{ and } s\in(\mathcal{A}\backslash\{a^*\}\times\mathcal{B}\backslash\{b^*,b_r\})^{m-1} \\
\psi(\mathsf{dB}^{+1}(s^*)|(s,a,b)) = 1/2 \qquad \forall a,b \in \mathcal{A}\backslash\{a^*\}\times\mathcal{B}\ ,\ \forall s\neq s^* \text{ and } s\in(\mathcal{A}\backslash\{a^*\}\times\mathcal{B}\backslash\{b^*,b_r\})^{m-1} \\
\psi(b|(s,a,b)) = 1 \qquad \forall a,b \in \mathcal{A}\backslash\{a^*\}\times\mathcal{B}\ ,\ \forall s\notin(\mathcal{A}\backslash\{a^*\}\times\mathcal{B}\backslash\{b^*,b_r\})^{m-1} \\
\psi(b^*|(s^*,a^*,b)) = 1/2+\epsilon \qquad \forall b\in\mathcal{B} \\
\psi(b_r|(s^*,a^*,b)) = 1/2-\epsilon \qquad \forall b\in\mathcal{B} \\
\psi(b^*|(s,a^*,b)) = 1/2 \qquad \forall b\in\mathcal{B}\ ,\ \forall s\neq s^* \text{ and } s\in(\mathcal{A}\backslash\{a^*\}\times\mathcal{B}\backslash\{b^*,b_r\})^{m-1} \\
\psi(b_r|(s,a^*,b)) = 1/2 \qquad \forall b\in\mathcal{B}\ ,\ \forall s\neq s^* \text{ and } s\in(\mathcal{A}\backslash\{a^*\}\times\mathcal{B}\backslash\{b^*,b_r\})^{m-1}
\end{cases}$$

This choice of ψ considers the previous m interactions, and based on the learner's actions in m-th previous interaction, it either goes to the next action suggested by the de Bruijn policy with probability $\frac{1}{2}$, or it takes between two special actions of b^* and b_r. In the case that there are at least one of the actions a^*, b^* or b_r in the previous $m-1$ actions, the opponent takes the last action of the state deterministically, so that it makes a circular return. In special states with the prefix of s^*_{m-1} the probability of taking b^* is $\frac{1}{2}+\epsilon$, which implies the optimal strategy to be reaching to $\bar{s}^*$ and then taking action a^* followed the action in s^*_{m-1} to get back to $\bar{s}_*$. When the learner's action reaches the m-th previous action, then ψ probabilistically chooses b^* or b^r,i.e. the action that helps in repetition of the state (refer to figure 4a).

The optimal gain for this MDP is $\frac{1}{2m}+\frac{\epsilon}{m}$, as in each m iteraitons it gets $\frac{1}{2}+\epsilon$ reward on expectation. So the optimal policy iterates in the state space of, $\mathcal{G}_\psi = \{s\in\mathcal{S}_m : s = (a^*_m, b^*_m, \dots a^*_m, b^*_m, a^*, b_r), s = (a^*_m, b^*_m, \dots a^*_m, b^*_m, a^*, b^*)$and their rotations$\}$ So the size of $|\mathcal{G}_\psi| \leq 2m$. Now suppose an algorithm Alg playing with ψ, induces an occupancy measure $\lambda_\psi(s) = \sum_{t=1}^{T}\mathbb{P}_\psi[S_t = s]$. We denote an state $s'\in(\mathcal{A}\backslash\{a^*\}\times\mathcal{B}\backslash\{b^*,b_r\})^{m-1}\backslash\mathcal{G}_\psi$ that has the least occupancy measure. That will be the Achilles' heel of the algorithm (refer to figure 4b). According to this choice, $\lambda_\psi(s')\leq\frac{T}{C_L}$ where $C_L = (\boldsymbol{A}-1)^{m-1}(\boldsymbol{B}-2)^{m-1}-2m$. Now suppose another opponent's policy ψ' which is exactly identical to ψ, except it has the probability of taking b^* equal to $\frac{1}{2}+2\epsilon$ in state s'. So the optimal policy in response to ψ' is iterating on s' instead of $\bar{s}^*$, and it achieves optimal gain of $\frac{1}{2m}+\frac{2\epsilon}{m}$. Now we can write,

$$\mathsf{Reg}_\psi(Alg,T) \geq \mathbf{P}_\psi[\mathsf{T}(\mathcal{G}_\psi)\leq\frac{T}{2}]\frac{T\epsilon}{2m},$$

where $\mathsf{T}(\mathcal{G})$ is the number of iterations that a state $s\in\mathcal{G}$ is occupied. The inequality holds because when at least $T/2$ of iterations are not in the states of $\mathcal{G}_\psi$, for each iteration we are suffering ϵ/m regret on average. The

same can be said for ψ',

$$\mathsf{Reg}_{\psi'}(Alg, T) \geq \mathbf{P}_{\psi'}[\mathsf{T}(\mathcal{G}_\psi) \geq \frac{T}{2}]\frac{T\epsilon}{2m},$$

because each iteraion consumed in $\mathcal{G}_\psi$ is accompanied by a regret of ϵ/m on average when the opponent is playing ψ'. So from Bretagnolle–Huber in equality we have,

$$\begin{aligned}\mathsf{Reg}_{\psi}(Alg, T) + \mathsf{Reg}_{\psi'}(Alg, T) &\geq (\mathbf{P}_{\psi}[\mathsf{T}(\mathcal{G}_\psi) \leq \frac{T}{2}] + \mathbf{P}_{\psi'}[\mathsf{T}(\mathcal{G}_\psi) \geq \frac{T}{2}])\ \frac{T\epsilon}{2m} \\ &\geq \frac{1}{2}\exp(-D_{KL}(\mathbf{P}_\psi \| \mathbf{P}_{\psi'}))\frac{T\epsilon}{2m}.\end{aligned}$$

Now from the two previous lemmas, we have,

$$\begin{aligned}\mathsf{Reg}_{\psi}(Alg, T) + \mathsf{Reg}_{\psi'}(Alg, T) &\geq \frac{1}{2}\exp(-D_{KL}(\mathbf{P}_\psi \| \mathbf{P}_{\psi'}))\frac{T\epsilon}{2m} \\ &\geq \frac{1}{2}\exp(-\mathbb{E}_\psi(\mathsf{T}(s'))D_{KL}(\psi(s') \| \psi'(s')))\frac{T\epsilon}{2m} \\ &\geq \frac{1}{2}\exp(-\frac{T}{C_L}(8\epsilon^2 + c\epsilon^4))\frac{T\epsilon}{2m},\end{aligned}$$

which means by choosing $\epsilon = \sqrt{C_L/T} \leq 1/4$ for big enough T, we have $\exp(-\frac{T}{C_L}(8\epsilon^2 + c\epsilon^4))$ less than a constant and,

$$\mathsf{Reg}_{\psi}(Alg, T) + \mathsf{Reg}_{\psi'}(Alg, T) \geq \frac{C}{m}\sqrt{T((\boldsymbol{A}-1)^{m-1}(\boldsymbol{B}-2)^{m-1} - 2m)}$$

so for at least one of the regrets we have,

$$\mathsf{Reg}(Alg, T) \in \Omega(\frac{1}{m}\sqrt{\boldsymbol{A}^{m-1}\boldsymbol{B}^{m-1}T}).$$

□

D Experiments

In this section, we demonstrate the empirical performance of the meta algorithm MRBEAR confirming the previous theoretical results.

Environment and Setting. We use the repeated game environment described in the previous section. The learner and the opponent both have two actions (pure strategies) making the 2×2 stage game of G (refer to 3).

The entries contain the utility of the learner for each action profile. The learner receives a reward of 1 only when the action profile (a_1, b_1) is played, otherwise, he gets 0. The opponent's policy ψ is a second order self oblivious policy with the following descriptions,

$$\psi(.|s) = \begin{cases}(0.1, 0.9) & s = (a_0, a_0) \\ (0.9, 0.1) & o.w.\end{cases}$$

This policy takes action b_1 with a probability of 0.9 only when the previous two actions played by the learner are both a_0. In any other cases, i.e. when the last two actions played by the learner are (a_0, a_1), (a_1, a_0) or (a_1, a_1) it takes b_0 with probability of 0.9. It is not hard to verify that the induced environments by this opponent's policy are ergodic, and the optimal strategy for the learner is to make a sequence of $(a_0, a_0, a_1, a_0, a_0, a_1, \dots)$, i.e.

$$\pi^*(.|s) = \begin{cases}(0, 1) & s = (a_0, a_0) \\ (1, 0) & o.w.\end{cases}$$

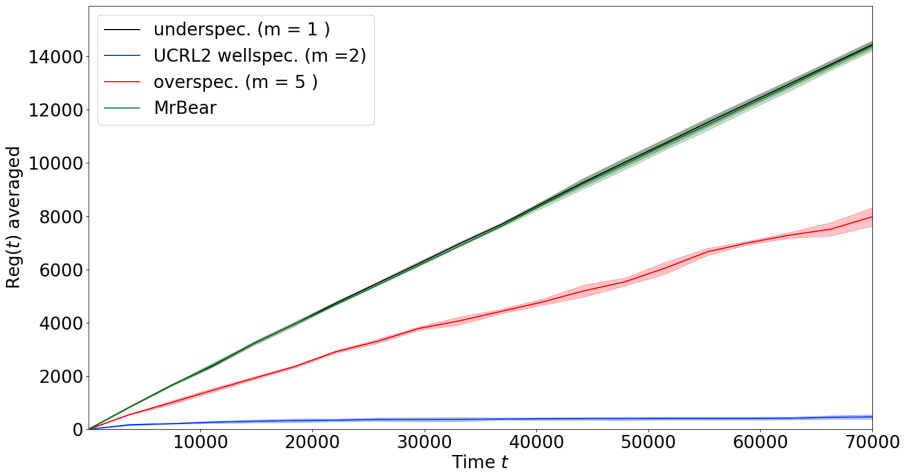
The optimal gain and optimal bias vectors are $[0.3, 0.3, 0.3, 0.3]^\intercal$ and $[0.3, -0.3, 0, -0.3]^\intercal$ respectively. Thus $\mathsf{sp}(h^*) = 0.6$. The above choice of ψ makes the optimal order $m^* = 2$ which means the second order model class

with 4 states is optimal. The model class of order $m = 1$ with 2 states is misspecified (underspecified) and the model classes of order $m \geq 3$ are overspecified.

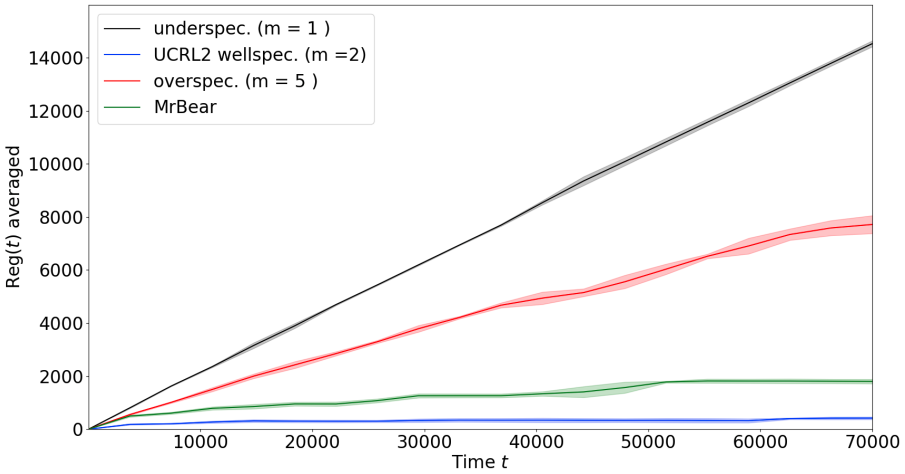
Different Choices for Base Algorithms. We run MRBEAR over Regal, UCRL2, and KLUCRL as base algorithms. The performance of algorithms in terms of their cumulative regret is depicted. As there is no first-order policy for the learner with the optimal gain, the regret of KLUCRL instance on $m = 1$ is linear (Figure 1a). On the other hand, knowing the optimal model class $m = 2$ in hindsight, one could directly go for its base algorithm and achieve the best performance in the plot. Finally, due to the realizability assumption, one could conservatively run the algorithm of order $m = 5$. In this overspecified case, the regret is sublinear but due to the larger size of MDPs (32 states), its curve is too slow. The performance of MRBEAR is between this conservative case and the best case. We show the results of using Regal (Bartlett and Tewari, 2012) as the base algorithms in Figure 1b.

Importance of Having Sharp Potential Bounds. As it is clear from the procedure of MRBEAR and its regret bound (section 3), the sharper the potential regret bounds are, the better the upper bound MRBEAR obtains. We empirically show this effect by using instances of UCRL2 as the base algorithms. In Figure 5a we use the potential bound $\mathsf{B}_i(T, \delta) \in \tilde{\Theta}(D_i \boldsymbol{S}_i \sqrt{AT})$ which is the result of (Auer et al., 2008b). This loose bound causes MRBEAR to bear with the under-specified algorithm of order $m = 1$ for too many iterations, making the performance of MRBEAR almost identical to the first base algorithm for $70k$ iterations. However, if we use the same bound as what we used for KLUCRL meaning $\mathsf{B}_i(T, \delta) \in \tilde{\Theta}(D_i \sqrt{T})$ (which is not theoretically proved for UCRL2), we obtain a much better empirical performance shown in Figure 5b.

Our code, which is available in the supplementary material, is based on the implementations of the PMEVI paper by Boone and Zhang (2024) which is open-source.



(a) Potential bounds in $\tilde{\Theta}(D_i \boldsymbol{S}_i \sqrt{AT})$ (MRBEAR and under-specified UCRL2 overlap each other).



(b) Potential bounds in the order of $\tilde{\Theta}(D_i \sqrt{T})$.

Figure 5: Using UCRL2 as base algorithm with different potential bounds.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**. The code is available in the supplementary material.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**]
   (b) Complete proofs of all theoretical results. [**Yes**]
   (c) Clear explanations of any assumptions. [**Yes**]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**. The code is available in the supplementary material.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**, but partially not applicable.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**. Partially not applicable since the implementations are not too heavy.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**]
   (b) The license information of the assets, if applicable. [**Yes**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
   (d) Information about consent from data providers/curators. [**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]