On-line Learning in Tree MDPs by Treating Policies as Bandit Arms

Anvay Shah
Indian Institute of Technology Bombay
Mumbai, India
anvay@cse.iitb.ac.in

Ramsundar Anandanarayanan
Indian Institute of Technology Bombay
Mumbai, India
ramsundar@cse.iitb.ac.in

Sharayu Moharir
Indian Institute of Technology Bombay
Mumbai, India
sharayum@ee.iitb.ac.in

Shivaram Kalyanakrishnan
Indian Institute of Technology Bombay
Mumbai, India
shivaram@cse.iitb.ac.in

ABSTRACT

A Tree Markov Decision Problem (T-MDP) is a finite-horizon MDP with a starting state s_1, in which every state is reachable from s_1 through exactly one state-action trajectory. T-MDPs arise naturally as abstractions of decision making in sequential games with perfect recall, against stationary opponents. We consider the problem of on-line learning in T-MDPs, both in the PAC and the regret-minimisation regimes. We show that well-known bandit algorithms—Lucb and Ucb—can be applied on T-MDPs by treating each policy as an arm. The apparent technical challenge in this approach is that the number of policies is exponential in the number of states. Our main innovation is in the design of confidence bounds based on data shared by the policies, so that the bandit algorithms can yet be implemented with polynomial memory and per-step computation. We obtain instance-dependent upper bounds on sample complexity and regret that sum a "gap term" from every terminal state, rather than every policy. Empirically, our algorithms consistently outperform available alternatives on a suite of hidden-information games.

KEYWORDS

MDPs; Imperfect Information Games; PAC; Regret.

ACM Reference Format:
Anvay Shah, Ramsundar Anandanarayanan, Sharayu Moharir, and Shivaram Kalyanakrishnan. 2026. On-line Learning in Tree MDPs by Treating Policies as Bandit Arms. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 16 pages. https://doi.org/10.65109/PYUD1139

1 INTRODUCTION

A chief contributor to the rapid advent of artificial intelligence (AI) in the last couple of decades is the widespread adoption of data-driven algorithms. In the realm of decision making, major practical successes—in areas such as robotics [28], game-playing [45, 56], large language models [14, 40], financial trading [37]—have been achieved through reinforcement learning (RL) [49]. The theoretical study of RL [3, 17, 27] has regularly benefitted its empirical progress.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/PYUD1139

In this paper, we consider on-line learning on sequential decision making problems that can be formalised as Tree Markov Decision Problems (T-MDPs). A T-MDP is a finite-horizon MDP in which the state-transitions form a *tree*, rooted at a starting state s_1. Thus, every state in a T-MDP is reachable from s_1 through exactly one state-action trajectory. T-MDPs arise commonly as abstractions of extensive-form games with perfect recall [44], against fixed opponents. The key challenge in these games is their large scale (amplified by hidden information) and the uncertainty regarding the opponent's strategy. Our work is motivated by such games, where the goal is to learn a *best response* against an *unknown* but stationary opponent, through repeated game interactions. This setup gives rise to an on-line learning problem in a T-MDP, in which the agent's states are action-observation histories.

A natural approach for designing learning algorithms for MDPs is to view them as generalisations of multi-armed bandits, their well-understood stateless counterparts [15]. Can on-line learning algorithms for bandits be appropriately generalised for MDPs? Simply put, our paper is the substantiation of an affirmative answer for the special case of T-MDPs. We take up two well-known bandit algorithms: (1) Lucb [24], which achieves order-optimal sample complexity in the PAC setting, and (2) Ucb [2], whose regret is within a constant factor of optimal. We adapt these algorithms to T-MDPs, and denote the generalisations Lucb-T and Ucb-T.

The generalisations share a common core, in which each policy for the T-MDP is treated as a bandit arm. The main technical challenge is that the number of policies is *exponential* in the size of the T-MDP. We propose a framework for policies to share data, tying it to a concentration inequality that we specifically establish for certain families of dependent random variables. We obtain instance-specific upper bounds on the sample complexity (for Lucb-T) and regret (for Ucb-T), both of which include a "gap term" for each terminal state in the T-MDP (whereas a naïve implementation would yield such a term for every policy). The tree structure also facilitates efficient computation: these algorithms need only a polynomial number of operations to make each decision.

Our algorithms assume that the reward function of the T-MDP is *known*, and that the only parameters to estimate are transition probabilities. While restrictive in some cases, this assumption is reasonable for our motivating domain of imperfect information games. We implement Lucb-T and Ucb-T on three games with varying state-space sizes: Kuhn Poker [31] (10's of states), Leduc Poker [47]

(100's of states), and Reconnaissance Blind Tic-Tac-Toe (RBT) [48] (millions of states). RBT is inspired by Reconnaissance Blind Chess [20, 42], and is significantly larger than test problems considered in the literature. We observe that our algorithms consistently outperform alternatives, especially as the problem size increases. Our experiments also throw light on gaps between theory and practice, especially in the PAC setting. In summary, our contributions include

(1) a new analytical result (Section 3) with potentially broader applicability to the analysis of learning in MDPs;
(2) conceptually-simple algorithms for T-MDPs (Section 4);
(3) strong theoretical guarantees in the form of instance-specific upper bounds on sample complexity (Theorem 9) and regret (Theorem 10); and
(4) empirical validation on large T-MDPs, arising from well-known imperfect-information games (Section 5).

We also publish our code[1], so our work can be reproduced and built upon. We begin by formalising our problem statements.

2 PROBLEM DESCRIPTION

In this section, we define Tree MDPs, and thereafter specify the PAC and regret-minimisation problems.

2.1 Tree Markov Decision Problems

A Tree MDP (T-MDP) $\mathcal{M}$ is specified by a 7-tuple $\langle \mathcal{S}, \Sigma, \mathcal{A}, p, r, \gamma, \mathcal{H} \rangle$. Here $\mathcal{S}$ and Σ are sets of non-terminal and terminal states, respectively. We assume that $\mathcal{S}$ contains a starting state s_1. $\mathcal{A}$ is a set of actions, and $\mathcal{H} \geq 1$ is the horizon. Transition probabilities are given by the function p, while the function r specifies rewards. $\gamma \in [0, 1]$ is used to discount future rewards in the definition of values.

A T-MDP induces a tree rooted at s_1, which is the only state at "level" 1. As illustrated in Figure 1, suppose $s \in \mathcal{S}$ is a state at level $1 \leq h \leq \mathcal{H}$. When the agent takes action a from s, it goes to state s', which is either (1) terminal, or (2) a non-terminal state at level $h+1$. The probability of this transition is $p(s, a, s')$. It is a property of T-MDPs that each "next state" s' has exactly one parent s from one lower level, from which s' is reachable through exactly one action a.

The transition from s to s' by taking action a merits a numeric reward $r(s, a, s')$. We assume that this reward is known—an assumption that is reasonable for many practical applications [39, 56], particularly those from our motivating domain of games [36, 43, 51]. This requirement is not uncommon in the on-line learning literature [33], although many approaches can also accommodate stochastic and unknown rewards [53, 55]. We assume that the discounted cumulative reward (that is, the *return*) along each trajectory is bounded, and for convenient exposition taken to lie in $[0, 1]$.

[1] Codebase: https://github.com/anvay09/On-line-Learning-in-Tree-MDPs.

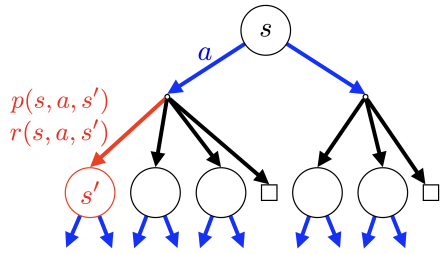


Figure 1: Transition from non-terminal state s to state s' upon taking action a. State s' could be non-terminal or terminal.

A deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$ specifies an action $\pi(s)$ for every non-terminal state s. Let Π be the set of all deterministic policies for MDP $\mathcal{M}$. For every non-terminal state $s \in \mathcal{S}$, the *value* under π is given by the Bellman equation

$$V^{\pi}(s) = \sum_{s' \in \mathcal{S} \cup \Sigma} p(s, \pi(s), s') \left(r(s, \pi(s), s') + \gamma V^{\pi}(s') \right), \quad (1)$$

with the convention that for every terminal state $\sigma \in \Sigma$, $V^{\pi}(\sigma) = 0$. The tree structure of $\mathcal{M}$ enables π to be conveniently evaluated bottom-up. A policy π that maximises the RHS in (1) for each state $s \in \mathcal{S}$ is an *optimal* policy. Let $\Pi_{\text{opt}} \subseteq \Pi$ to be the set of optimal policies. We arbitrarily pick one optimal policy and denote it $\pi^{\star}$. Of particular interest to us is the value of policies at the start state s_1. For $\pi \in \Pi$, we define $V(\pi) \stackrel{\text{def}}{=} V^{\pi}(s_1)$.

2.2 On-line Learning Problems

Our agent faces the challenge of not knowing the transition function p. However, the agent can interact with the MDP by playing complete episodes starting at s_1. Suppose that on episode $t \geq 1$, the agent plays a policy π^t. From each state s that is reached, an action is taken according to π^t. The environment generates next state s' by sampling $p(s, \pi^t(s))$. If s' is terminal, we proceed to the next episode $t+1$, wherein the agent can select a fresh policy π^{t+1}. Note that the agent does *not* have arbitrary access to sample any state-action pair in the tree, as is assumed in some other work [3, 58]. Thus, if the agent wishes to explore some particular state, it may have to try for multiple episodes until randomness takes it down that state's path.

For each episode $t \geq 1$, a learning algorithm $\mathcal{L}$ must pick a policy π^t, based on the trajectories observed in the preceding $t-1$ episodes. Recall that each trajectory is a state-action sequence beginning with s_1, and ending in a terminal state. Additionally, in the PAC setting, the learning algorithm may stop after any number of episodes t and return π^t as answer.

2.2.1 PAC. In the PAC setting, a tolerance $\epsilon \in (0, 1)$ and a mistake probability $\delta \in (0, 1)$ are given as input to learning algorithm $\mathcal{L}$. The algorithm is required to stop with probability 1 on every input T-MDP. Also, the policy it returns must satisfy $V(\pi) \geq V(\pi^{\star}) - \epsilon$ with probability at least $1-\delta$. The main question is how many samples (that is, episodes) are required to provide such a guarantee. The PAC problem described above is one of "pure exploration", in the sense that the rewards accrued while learning are not of consequence.

2.2.2 Regret. The regret of an algorithm after T episodes is

$$R_T \stackrel{\text{def}}{=} \sum_{t=1}^{T} \left(V(\pi^{\star}) - \mathbb{E}[V(\pi^t)] \right).$$

We desire an algorithm that minimises regret on every T-MDP instance. Unlike the PAC setting, it is well-known that regret-minimisation needs to balance exploring (playing less-sampled policies) and exploiting (playing empirically-dominant policies).

From the specifications above, it is apparent that bandit algorithms such as Lucb and Ucb can be run "as is" on T-MDPs, by treating each

policy as an arm. However, since the number of policies is $|\mathcal{A}|^{|\mathcal{S}|}$, the computational and memory requirements of naïve implementations would be prohibitive on all but toy tasks. Our main contribution is an efficient, scalable implementation that exploits the structure of T-MDPs—and performs well both in theory and in practice. These algorithms scale polynomially in the number of possible trajectories from the starting state, which, in T-MDPs, is the same as the number of terminal states. In regular MDPs, the number of trajectories could, in general, be exponential in the number of states.

3 CONFIDENCE BOUNDS ON $V(\pi)$

The main tool we devise is a confidence bound on $V(\pi)$ for arbitrary policy $\pi \in \Pi$. Facilitating this bound is a representation of $V(\pi)$ as a convex combination of terminal returns.

3.1 Bottom-up View of $V(\pi)$

The Bellman equations in (1) recursively define a parent node's value in terms of its children's. We introduce notation to unroll this recursion, up to the base case involving only terminal states.

Due to the tree property of T-MDPs, every state $s \in \mathcal{S} \cup \Sigma$ has a unique path from the root s_1; suppose for s the path is $s_1, a_1, s_2, a_2, \ldots, s_m, a_m, s$ for some $m \geq 1$. We denote by $\rho(s)$ the discounted cumulative reward (that is, the return) along this path, and by $q(s)$ the probability of reaching s from s_1 if taking action a_i from s_i for $1 \leq i \leq m$. That is:

$$\rho(s) \stackrel{\text{def}}{=} \sum_{i=1}^{m} \gamma^{i-1} r(s_i, a_i, s_{i+1}); \qquad q(s) \stackrel{\text{def}}{=} \prod_{i=1}^{m} p(s_i, a_i, s_{i+1}), \tag{2}$$

where s_{m+1} is taken to be s.

A policy $\pi \in \Pi$ is defined to be *consistent* with terminal state $\sigma \in \Sigma$ if π takes the action following each state in the path to σ. Concretely, suppose σ has path $(s_1, a_1, s_2, a_2, \ldots, s_m, a_m, \sigma)$. Then $\pi \in \Pi$ is consistent with σ if and only if for $1 \leq i \leq m$, $\pi(s_i) = a_i$. Observe that in general, a policy π can be consistent with multiple terminal states; on any episode one of these states will be reached. On the other hand, also note that multiple policies can be consistent with a given terminal state σ—such policies would differ on states that are not on the path to σ. For $\pi \in \Pi$, let $X(\pi)$ denote the subset of terminal states with which π is consistent; similarly, for $\sigma \in \Sigma$, let $Y(\sigma)$ denote the set of policies that are consistent with σ:

$$X(\pi) = \{\sigma \in \Sigma : \pi \text{ is consistent with } \sigma\}, \tag{3}$$

$$Y(\sigma) = \{\pi \in \Pi : \pi \text{ is consistent with } \sigma\}. \tag{4}$$

Repeated expansion of the RHS of (1), while invoking definitions from (2), yields the following decomposition of $V(\pi)$; a detailed working is provided in Appendix A.[2]

Proposition 1 (Value as a convex combination of terminal returns). *For $\pi \in \Pi$,*

$$V^{\pi}(s_1) = \sum_{\sigma \in X(\pi)} q(\sigma) \cdot \rho(\sigma). \tag{5}$$

The proposition tells us that in order to learn $V(\pi)$, it suffices to estimate $q(\sigma)$ for all $\sigma \in X(\pi)$. This reduces the problem of estimating values for all policies in Π (an exponentially-sized set) to estimating the "q"-s for all the elements of Σ. Moreover, we can get data for estimating $q(\sigma)$ merely by playing any policy that is consistent with σ—which may or may not reach σ on any given episode. On the other hand, if $\rho(\sigma)$ was also unknown and had to be estimated, note that we would get sufficient information for it only from episodes that do reach σ. Thus, our assumption of known rewards carries a non-trivial advantage.

3.2 Upper and Lower Confidence Bounds

Consider the run of any arbitrary algorithm. On the t-th episode, $t \geq 1$, for each terminal state $\sigma \in \Sigma$, let $n^t(\sigma)$ denote the number of episodes so far on which the policy played was consistent with σ, and let $n^t_+(\sigma)$ denote the number of times σ was reached. These counts can be maintained using $\Theta(|\Sigma|)$ integer operations per episode. If policy π is played on the t-th episode, and reaches σ, the updates are

$$n^{t+1}(\sigma') \leftarrow n^t(\sigma') + 1, \text{ for all } \sigma' \in X(\pi); \tag{6}$$

$$n^{t+1}_+(\sigma) \leftarrow n^t_+(\sigma) + 1, \tag{7}$$

while for other terminal states the counts do not change from episode t to episode $t+1$. With a slight abuse of notation, we define

$$n^t(\pi) \stackrel{\text{def}}{=} \min_{\sigma \in X(\pi)} n^t(\sigma)$$

for $\pi \in \Pi$ as a "play count" signifying the amount of usable data we have for evaluating π after $t-1$ episodes. Note that π need not be played at all for $n^t(\pi)$ to be positive—we only need policies consistent with states in $X(\pi)$ to have been played.

Notice that the ratio $\widehat{q}^t(\sigma) \stackrel{\text{def}}{=} \frac{n^t_+(\sigma)}{n^t(\sigma)}$ is an unbiased estimator of $q(\sigma)$; consequently the empirical value estimate

$$\widehat{V}^t(\pi) \stackrel{\text{def}}{=} \sum_{\sigma \in X(\pi)} \widehat{q}^t(\sigma) \cdot \rho(\sigma)$$

is an unbiased estimator of $V(\pi)$. Due to our convention that $\rho(\cdot)$ lies in $[0,1]$, and since $\widehat{q}^t(\cdot)$ must also lie in $[0,1]$, it follows that $\widehat{V}^t(\pi)$ must lie in $[0, |X(\pi)|]$. As seen shortly, we clip this estimate to $[0,1]$ for "one direction" of our algorithm and analysis.

At the core of our technical contribution are the following confidence bounds that we propose on $V(\pi)$. For $\delta \in (0,1)$,

$$\text{ucb}^t(\pi, \delta) \stackrel{\text{def}}{=} \widehat{V}^t(\pi) + \beta(n^t(\pi), \delta), \tag{8}$$

$$\text{lcb}^t(\pi, \delta) \stackrel{\text{def}}{=} \min\left\{\widehat{V}^t(\pi), 1\right\} - \beta(n^t(\pi), \delta), \tag{9}$$

$$\text{where } \beta(m, \delta) \stackrel{\text{def}}{=} \sqrt{\frac{8}{3m} \ln \frac{1}{\delta}} \text{ for } m \geq 1.$$

In (8) and (9), observe that the "confidence width" $\beta(\cdot,\cdot)$ for π depends on $n^t(\pi)$, which, in turn is determined by the *least-played* terminal state of π. When π is played, every terminal state in $X(\pi)$ is played; hence the confidence width of π necessarily decreases.

Theorem 2 (Confidence bounds on policy's value). *Consider any policy $\pi \in \Pi$. Consider any given $t \geq 1$ and sequence $\left(n^t(\sigma)\right)_{\sigma \in X(\pi)}$ with $1 \leq n^t(\sigma) \leq t$ for all $\sigma \in X(\pi)$. During a run of any algorithm, suppose t equals the number of episodes and $n^t(\sigma)$ equals the number of plays of σ for each $\sigma \in X(\pi)$, the terminal states, respectively. Then, for $\delta \in (0,1)$:*

$$\mathbb{P}\{V(\pi) \geq ucb^t(\pi, \delta)\} \leq \delta; \; \mathbb{P}\{V(\pi) \leq lcb^t(\pi, \delta)\} \leq \delta.$$

[2] Appendices are included in a longer version of the paper linked from SK's home page: https://www.cse.iitb.ac.in/~shivaram/.

The non-trivial proof of this theorem is a central contribution of our paper, and takes up the remainder of this section. The main technical challenge is the *dependence* among the data used to compute $\widehat{V}^t(\pi)$. Applying results from the literature on concentration inequalities [19, 41], we establish that the particular nature of this dependence yet permits the use of Chernoff bounds, which we then apply. Our exposition below assumes that π, t, and $\left(n^t(\sigma)\right)_{\sigma \in X(\pi)}$ are given.

3.2.1 $\widehat{V}^t(\pi)$ as a weighted sum of Bernoullis. For each $\sigma \in X(\pi)$ and $1 \le i \le n^t(\sigma)$, let the Bernoulli random variable $B(\sigma, i)$ denote the outcome of the i-th time that a policy consistent with σ was played. $B(\sigma, i)$ is 1 if this play reaches σ, otherwise it is 0. We informally refer to these random variables as "B-variables", and denote their collection $\mathcal{B}$. Note that $|\mathcal{B}| = \sum_{\sigma \in X(\pi)} n^t(\sigma)$. We observe that $\widehat{V}^t(\pi)$ is a weighted combination of B-variables, with non-negative weights.

$$\widehat{V}^t(\pi) = \sum_{\sigma \in X(\pi)} \frac{\rho(\sigma)}{n^t(\sigma)} \sum_{i=1}^{n^t(\sigma)} B(\sigma, i). \tag{10}$$

Since $B(\sigma, i)$ has mean $q(\sigma)$; we observe from Proposition 1 that $\mathbb{E}[\widehat{V}^t(\pi)] = \sum_{\sigma \in X(\pi)} \rho(\sigma) \cdot q(\sigma) = V(\pi)$. For proving the theorem, we need to upper-bound the probability that $\widehat{V}^t(\pi)$ deviates from its expectation $V(\pi)$ by more than some amount in each direction.

3.2.2 Negative cylinder dependence of $\mathcal{B}$. Notice that for $\sigma \in X(\pi)$ and $1 \le i < j \le n^t(\sigma)$, $B(\sigma, i)$ and $B(\sigma, j)$ are *independent*, since they must necessarily come from different episodes. However, For $\sigma, \sigma' \in X(\pi)$ and $1 \le i \le j \le n^t(\sigma)$, $B(\sigma, i)$ and $B(\sigma', j)$ *need not* be independent, since the i-th play of σ and the j-th play of σ' may be on the *same* episode. At an intuitive level, it appears that this dependence between $B(\sigma, i)$ and $B(\sigma', j)$ should only help, since when one of them is 1 (or 0), the other has a higher probability of being 0 (respectively 1), thereby keeping the average more concentrated. Formally, these random variables are "negative cylinder dependent" (NCD).

Definition 3 (Negative Cylinder Dependence [19]). *Bernoulli random variables $Z_1, Z_2, \ldots, Z_m$, where $m \ge 2$, are negative cylinder dependent (NCD) if and only if for each $S \in \{1, 2, \ldots, m\}$,*

$$\mathbb{P}\left\{\cap_{i \in S}(Z_i = 1)\right\} \le \prod_{i \in S} \mathbb{P}\{Z_i = 1\}; \text{ and}$$
$$\mathbb{P}\left\{\cap_{i \in S}(Z_i = 0)\right\} \le \prod_{i \in S} \mathbb{P}\{Z_i = 0\}.$$

Lemma 4 (B-variables are NCD). *The set of random variables $\mathcal{B}$ are Negative Cylinder Dependent.*

The intuition behind this claim is as follows. A subset of B-variables cannot all be 1, when conditioned on the event that any two of them occur on the same episode. Nor can they all be 0, when conditioned on the event that even one of the episodes generating them results in 0 as the outcome for every terminal state in the complement of this subset. Our formal proof involves such conditioning of the LHS probabilities, removing terms from 0-probability events, and re-aggregating the surviving terms. This working involves lengthy mathematical expansions, and for reasons of space is deferred to Appendix B.

The standard recipe for analysing NCD variables is to introduce "twins", which are amenable to the application of Chernoff bounds.

3.2.3 Twin variables. For each B-variable in $\mathcal{B}$, we define a corresponding "twin-variable" (called a "B_0-variable") that is also a Bernoulli with the same mean, but which is generated independently (of all the other variables). Thus, for each $\sigma \in X(\pi)$, $1 \le i \le n^t(\sigma)$, the Bernoulli random variable $B_0(\sigma, i)$ is generated independently, and satisfies $\mathbb{E}[B_0(\sigma, i)] = \mathbb{E}[B(\sigma, i)] = q(\sigma)$. We also define $\widehat{V}_0^t$ to be the same linear combination of the B_0-variables as $\widehat{V}^t$ is of the B-variables. Thus, similar to (10), we get

$$\widehat{V}_0^t(\pi) = \sum_{\sigma \in X(\pi)} \frac{\rho(\sigma)}{n^t(\sigma)} \sum_{i=1}^{n^t(\sigma)} B_0(\sigma, i). \tag{11}$$

The twin variables are convenient since they are all independent. However, before making use of their independence, we establish the key relationship between the deviation tendencies of $\widehat{V}^t$ (which is the estimator used by our algorithm) and $\widehat{V}_0^t$ (its hypothetical twin).

Lemma 5 ($\widehat{V}^t$ concentrates no slower than $\widehat{V}_0^t$). *For $\epsilon > 0$,*

$$\mathbb{P}\{\widehat{V}^t \ge V(\pi) + \epsilon\} \le \mathbb{P}\{\widehat{V}_0^t \ge V(\pi) + \epsilon\};$$
$$\mathbb{P}\{\widehat{V}^t \le V(\pi) - \epsilon\} \le \mathbb{P}\{\widehat{V}_0^t \le V(\pi) - \epsilon\}.$$

The proof of this lemma proceeds through an expansion of the moment generating functions of $\widehat{V}^t(\pi)$ and $\widehat{V}_0^t(\pi)$, in the manner demonstrated by Panconesi and Srinivasan [41, see Theorem 3.2].

3.2.4 Deviation of $\widehat{V}_0^t$. Since $\widehat{V}_0^t$ is a linear combination of *independent* random variables, we use a standard inequality to upper-bound the probability of its deviation from its expectation. Bernstein's Inequality [5, see Section 2.7] is as follows.

Lemma 6 (Bernstein's Inequality [5]). *Let $X_1, X_2, \ldots, X_m$ be independent random variables with finite variance such that $|X_i| \le b$ for some $b > 0$ almost surely for $1 \le i \le m$. Let $S = \sum_{i=1}^m (X_i - \mathbb{E}[X_i])$ and $v = \sum_{i=1}^m \mathbb{E}[(X_i)^2]$. Then, for $\alpha > 0$,*

$$\mathbb{P}\{S \ge \alpha\} \le \exp\left(-\frac{\alpha^2/2}{v + \frac{b\alpha}{3}}\right).$$

Suitable application of this lemma, detailed below, yields the following result on the deviation of $\widehat{V}_0^t$.

Lemma 7 (Deviation of $\widehat{V}_0^t$). *For $\epsilon \in (0, 1]$,*

$$\mathbb{P}\{\widehat{V}_0^t \ge V(\pi) + \epsilon\} \le \exp\left(-\frac{3}{8} n^t(\pi) \epsilon^2\right);$$
$$\mathbb{P}\{\widehat{V}_0^t \le V(\pi) - \epsilon\} \le \exp\left(-\frac{3}{8} n^t(\pi) \epsilon^2\right).$$

Proof. We set up S, v, and b for the application of Lemma 6. For $\sigma \in X(\pi)$, $1 \le i \le n^t(\sigma)$, define

$$B_1(\sigma, i) \stackrel{\text{def}}{=} \frac{\rho(\sigma)}{n^t(\sigma)} \left(B_0(\sigma, i) - q(\sigma)\right).$$

There are $|\mathcal{B}|$ such bounded and mutually-independent "B_1" variables, where each is a linear combination of a corresponding B_0 variable. Since $|\rho(\sigma)| \leq 1$ and $|B_0(\sigma,i) - q(\sigma)| \leq 1$, it follows that $|B_1(\sigma,i)| \leq \frac{1}{n^t(\sigma)} \leq \frac{1}{n^t(\pi)} \stackrel{\text{def}}{=} b$. Also, we observe from (11) that the sum of the B_1-variables is $S \stackrel{\text{def}}{=} \widehat{V}^t - V(\pi)$. Finally, we have

$$v \stackrel{\text{def}}{=} \sum_{\sigma \in X(\pi)} \sum_{1=1}^{n^t(\pi)} \mathbb{E}\left[(B_1(\sigma,i))^2\right] = \sum_{\sigma \in X(\pi)} \frac{q(\sigma)(1-q(\sigma))}{n^t(\sigma)}$$
$$\leq \sum_{\sigma \in X(\pi)} \frac{q(\sigma)}{n^t(\sigma)} \leq \sum_{\sigma \in X(\pi)} \frac{q(\sigma)}{n^t(\pi)} = \frac{1}{n^t(\pi)}.$$

Invoking Lemma 6 with these values or bounds for S, v, and b, we observe that

$$\mathbb{P}\{\widehat{V}_0^t \geq V(\pi) + \epsilon\} \leq \exp\left(-\frac{\epsilon^2/2}{\frac{1}{n^t(\pi)} + \frac{\epsilon}{3n^t(\pi)}}\right) \leq \exp\left(-\frac{3}{8} n^t(\pi)\epsilon^2\right),$$

where we have used the fact that $\epsilon < 1$. Repeating the proof with the negations of the B_1-variables yields the same upper bound for $\mathbb{P}\{\widehat{V}_0^t \leq V(\pi) - \epsilon\}$. □

3.2.5 Final step. Our proof thus far has established the legitimacy of substituting the (possibly dependent) B-variables used in our algorithm with their independent, twin B_0-variables for the purpose of upper-bounding the deviation of $\widehat{V}^t$. We have applied Bernstein's inequality to the corresponding twin $\widehat{V}_0^t$, but notice that Lemma 7 only holds for deviations $\epsilon \in (0,1]$. We show below that this constraint does not disrupt the claim of Theorem 2.

The first part of the theorem considers the event $E_1 \equiv V(\pi) \geq \text{ucb}^t(\pi,\delta)$, which is equivalently $\widehat{V}^t(\pi) \leq V(\pi) - \beta(n^t(\pi),\delta)$. Since $V(\pi)$ is at most 1, and $\widehat{V}^t(\pi)$ is non-negative, E_1 is logically impossible if $\beta(n^t(\pi),\delta) > 1$. The second part of the theorem considers the event $E_2 \equiv V(\pi) \leq \text{lcb}^t(\pi,\delta)$, which is equivalently $\min\{\widehat{V}^t(\pi), 1\} \geq V(\pi) + \beta(n^t(\pi),\delta)$. Notice our explicit inclusion of the "min" operator in the lower confidence bound. Once again, E_2 cannot possibly occur if $\beta(n^t(\pi),\delta) > 1$. In summary, then, it suffices to show (1) $\mathbb{P}\{\widehat{V}^t(\pi) \leq V(\pi) - \beta(n^t(\pi),\delta)\} \leq \delta$ and (2) $\mathbb{P}\{\widehat{V}^t(\pi) \geq V(\pi) + \beta(n^t(\pi),\delta)\} \leq \delta$, both under the condition that $\beta(n^t(\pi),\delta) \leq 1$. This is the precise statement of Lemma 7, when applied with $\epsilon = \beta(n^t(\pi),\delta)$.

4 ALGORITHMS AND ANALYSIS

Having established confidence bounds for policies, we apply them in algorithms for the PAC and regret-minimisation settings. Well-known algorithms Lucb [24] and Ucb [2] are suffixed with "-T" to denote their application on T-MDPs.

4.1 Lucb-T for PAC Setting

Lucb-T, specified as Algorithm 1, is identical to the original Lucb algorithm [24], with each policy akin to a bandit arm. After initialising counts, for each batch t, *two* policies are identified: one with the highest value estimate, and one with the highest upper confidence bound (UCB) among the other policies. Confidence bounds are computed for mistake probability $\delta_L(t) \stackrel{\text{def}}{=} \frac{\delta}{3|\Pi|t^{|\Sigma|+4}}$ for each $t \geq 1$. If the two identified policies are already separated to within ϵ (line 8), the first is returned; otherwise both policies are played. By this convention, there are at most $2(t-1)$ episodes up to batch t.

Algorithm 1 Lucb-T

1: Initialise $n^t(\sigma)$, $n_+^t(\sigma)$ to 0 for $\sigma \in \Sigma$.
2: **for** $t = 1, 2, \ldots$ **do**
3: **if** there exists $\sigma \in \Sigma$ such that $n_\sigma^t = 0$ **then**
4: Play an arbitrary policy $\pi \in Y(\sigma)$.
5: **else**
6: $\pi_1^t \leftarrow \text{argmax}_{\pi \in \Pi} \widehat{V}_\pi^t$.
7: $\pi_2^t \leftarrow \text{argmax}_{\pi \in \Pi \setminus \{\pi_1\}} \text{ucb}_\pi^t$.
8: **if** $\text{lcb}^t(\pi_1^t, \delta_{\text{L}}(t)) \geq \text{ucb}^t(\pi_2^t, \delta_{\text{L}}(t)) - \epsilon$ **then**
9: **return** π_1^t.
10: Play policies π_1^t and π_2^t.
11: Set $n^t(\cdot)$, $n_+^t(\cdot)$ based on policies played, outcome.

4.1.1 Efficient Implementation. For any given policy π, it is clear that $X(\pi)$, $V^t(\pi)$, and $\text{ucb}^t(\pi)$ can be computed using $\Theta(|\mathcal{S}||\mathcal{A}| + |\Sigma|)$ operations. However, notice that lines 6 and 7 in Algorithm 1 require identifying policies that maximise the value estimate or the upper confidence bound (UCB). We illustrate below that although $|\Pi| = |\mathcal{A}|^{|\mathcal{S}|}$, policies π_1^t and π_2^t can be computed using $\text{poly}(|\mathcal{S}|, |\mathcal{A}|)$ steps.

Counts for the visits of each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times (\mathcal{S} \cup \Sigma)$ are updated after each episode, and the empirical transition probabilities $\widehat{p}(s,a,s')$ are obtained by normalising. If $\widehat{p}$ is used in the RHS of (1), then the optimising policy π_1^t and its value $\widehat{V}^t(\pi_1^t)$ are obtained bottom-up by setting as action for each state any one that maximises the corresponding RHS in (1).

It gets more involved to compute π_2^t. In Appendix H, we specify a procedure PU to compute a policy with the highest UCB; denote this policy $\pi_U^t \stackrel{\text{def}}{=} \text{argmax}_{\pi \in \Pi} \text{ucb}^t(\pi, \delta_U(t))$. In practice, π_U^t will usually be different from π_1^t, in which case it will itself be π_2^t. PU incurs $O(|\mathcal{S}||\mathcal{A}||k_{\max}|+|\Sigma|)$ operations, where $k_{\max}$ is the maximum branching factor (upper-bounded by $|\mathcal{S}|$; in practice much smaller). Now, if it happens that $\pi_U^t = \pi_1^t$, then another bottom-up pass is performed to recursively compute a maximum-UCB policy that is different from π_1^t. The basis of the recursion is that the policy following a state-action pair must be identical to π_1^t at all but one child, and different from π_1^t for exactly one child. This process incurs essentially the same computation as PU. Our submission includes full code for all our algorithms (Appendix G), along with the test environments described in Section 5.

4.1.2 Correctness. The correctness of Lucb-T follows from a union bound over mistake probabilities.

Proposition 8. *The probability that Lucb-T returns a policy $\pi \in \Pi$ such that $V(\pi) < V(\pi^\star) - \epsilon$ is at most δ.*

Proof. A non-ϵ-optimal policy can be returned only on the *bad* event that (1) on some batch $t \geq 1$, (2) for some sequence of play counts $(n^t(\sigma))_{\sigma \in \Sigma}$, and (3) for some policy $\pi \in \Pi$, the interval $[\text{lcb}^t(\pi, \delta_L(t)), \text{ucb}^t(\pi, \delta_L(t))]$ does *not* contain $V(\pi)$. On batch t, each play count must be between 1 and t. Applying Theorem 2, the probability of the bad event is at most $\sum_{t=1}^{\infty} t^{|\Sigma|}|\Pi| 2\delta_L(t) \leq \delta$. □

4.1.3 Sample Complexity. Sample-complexity analysis proceeds in the same manner as of Lucb [24], beginning with the definition of instance-specific "gaps". Define

$$V_2 \stackrel{\text{def}}{=} \max_{\pi \in \Pi_{\text{opt}} \setminus \{\pi^\star\}} V(\pi),$$

$$\Delta_\pi \stackrel{\text{def}}{=} \begin{cases} V(\pi^\star) - V_2 & \text{if } \pi = \pi^\star, \\ V(\pi^\star) - V(\pi) & \text{if } \pi \neq \pi^\star, \end{cases} \text{ and}$$

$$\Delta_\pi^\epsilon \stackrel{\text{def}}{=} \max\{\Delta_\pi, \epsilon\} \text{ for } \epsilon > 0.$$

For analysing Lucb-T, we additionally define for $\sigma \in \Sigma$,

$$\Delta_\sigma^\epsilon \stackrel{\text{def}}{=} \min_{\pi \in Y(\sigma)} \Delta_\pi^\epsilon.$$

We upper-bound the sample complexity of Lucb-T as follows.

Theorem 9. *For $\epsilon, \delta \in (0,1)$, the expected number of episodes taken by Lucb-T before termination is*

$$O\left(\sum_{\sigma \in \Sigma} \frac{|\Sigma|}{(\Delta_\sigma^\epsilon)^2}\left(\log\frac{1}{\delta} + \log \sum_{\sigma \in \Sigma} \frac{1}{(\Delta_\sigma^\epsilon)^2} + |\mathcal{S}| \log |\mathcal{A}|\right)\right).$$

Proof. The proof follows the same template as that of Lucb [24]. A policy π is called *needy* on episode t if its play count is smaller than a constant times $\frac{1}{(\Delta_\pi^\epsilon)^2} \ln \frac{1}{\delta_L(t)}$. A terminal state $\sigma \in \Sigma$ is called needy if it is the least-played state of some needy policy $\pi \in Y(\sigma)$. Notice that *non-needy* policies have sufficiently small width β. It is shown that if neither of the policies played on episode t are needy, then some policy must have violated its upper or lower confidence bound. By a union bound similar to that used in Proposition 8, the probability of such an event is at most a constant times $\frac{\delta}{t^3}$.

On the other hand, if indeed some needy policy is played on episode t, it means that some needy terminal state $\sigma \in \Sigma$ is played on episode t. Since there is a cap on the number of episodes in which σ can both be needy and get played, the total number of "good" episodes (which play needy σ's) is in the order of $\sum_{\sigma \in \Sigma} \frac{1}{(\Delta_\sigma^\epsilon)^2} \ln \frac{1}{\delta_L(t)}$ for sufficiently large $t \geq t^\star$. For an appropriate choice of $t^\star$, the probability of not stopping at or before $t^\star$ is at most $\frac{\delta}{(t^\star)^2}$—and this property implies the claimed upper bound. □

4.1.4 Tighter Upper Bound. Notice that $|\Sigma|$ factor in the sample-complexity upper bound from Theorem 9. This is an artefact of the union bound over variable play counts, which necessitated a $\frac{1}{t^{|\Sigma|}}$ factor in $\delta_L(t)$. An algorithmic change to Lucb-T can remove the $|\Sigma|$ factor. Rather than use all the available samples for each $\sigma \in X(\pi)$, we can use only the first $n^t(\pi)$ samples for each (possibly ignoring a lot of data for some terminal states). Since $n^t(\pi)$ has to be between 1 and t, the union is only over t (rather than $t^{|\Sigma|}$) events. In view of the same amount of data being used for each terminal state, we refer to this algorithmic variant as Lucb-T-Uniform.

Whereas Lucb-T only needs to store the counts $n^t(\sigma)$ and $n_+^t(\sigma)$ for $\sigma \in \Sigma$, Lucb-T-Uniform needs to store the entire sequence of outcomes (0's and 1's) from the plays of each $\sigma \in \Sigma$. Consequently, while Lucb-T uses polylog(t) memory, Lucb-T-Uniform needs $\Theta(t)$ memory. Thus, while the latter algorithm yields a superior upper bound, it is less convenient in practice, and in fact even performs worse on larger problem instances (coming up in Section 5).

4.2 Ucb-T for Regret-minimisation Setting

The same idea of viewing policies as bandit arms gives rise to Ucb-T (Algorithm 2), which implements the Ucb algorithm [1] for bandits. In this case, after initialising plays, a policy with the highest upper confidence bound is played on each episode. For episode $t \geq 1$, confidence bounds are for mistake probability $\delta_U(t) \stackrel{\text{def}}{=} \frac{1}{|\Pi| t^{|\Sigma|+4}}$.

Algorithm 2 Ucb-T

1: Initialise $n^t(\sigma), n_+^t(\sigma)$ to 0 for $\sigma \in \Sigma$
2: **for** $t = 1, 2, \ldots$ **do**
3: **if** there exists $\sigma \in \Sigma$ such that $n_\sigma^t = 0$ **then**
4: Play an arbitrary policy $\pi \in Y(\sigma)$.
5: **else**
6: $\pi^t \leftarrow \text{argmax}_{\pi \in \Pi} \text{ucb}^t(\pi, \delta_U(t))$.
7: Play policy π^t.
8: Set $n^t(\cdot), n_+^t(\cdot)$ based on policy played, outcome.

The computation of the policy maximising the UCB (line 6) is done by the PU procedure described in Appendix H.

4.2.1 Regret. The structure put forth by Auer et al. [2] to upper-bound the regret of Ucb also extends to Ucb-T. The same can also be interpreted through the terminology of "needy" policies and terminal states, presented in the proof of Theorem 9, but with two differences. First, only the plays of non-optimal policies—the elements of $\Pi \setminus \Pi_{\text{opt}}$—contribute to the regret. Second, for $\sigma \in \Sigma$, we conservatively upper-bound the number of needy plays of σ by assuming it is played by the policy in $Y(\sigma) \setminus \Pi_{\text{opt}}$ with the *smallest* gap. Conservatively, each such play still contributes regret according to the policy in $Y(\sigma) \setminus \Pi_{\text{opt}}$ with the *largest* gap. Formally, for $\sigma \in \Sigma$,

$$\Delta_\sigma^{\min} \stackrel{\text{def}}{=} \min_{\pi \in Y(\sigma) \setminus \Pi_{\text{opt}}} \Delta_\pi; \; \Delta_\sigma^{\max} \stackrel{\text{def}}{=} \max_{\pi \in Y(\sigma) \setminus \Pi_{\text{opt}}} \Delta_\pi.$$

We obtain the following upper bound on regret.

Theorem 10. *There exists $c > 0$ such that for $T \geq 2$, the regret R_T of the Ucb-T algorithm satisfies*

$$R_T \leq c \cdot \sum_{\sigma \in \Sigma, Y(\sigma) \not\subseteq \Pi_{opt}} \left(\frac{|\Sigma| \Delta_\sigma^{\max}}{(\Delta_\sigma^{\min})^2} \ln T + \ln |\Pi|\right).$$

The detailed proof is given in Appendix C. As discussed in Section 4.1.4, the $|\Sigma|$ factor can be removed, at the expense of additional memory and compute. In view of its limited practical value, we skip this variant for regret-minimisation.

5 EXPERIMENTAL EVALUATION

We evaluate our algorithms on two standard benchmark games, Kuhn Poker [31] and Leduc Poker [47], as well as a third game, Reconnaissance Blind Tic Tac Toe (RBT) [48]. Full descriptions are given in Appendix D. All three games are 2-player, zero-sum, with hidden information. We refer to the first and second players as x and o. Each "state" in these games is an action-observation history. Kuhn Poker and Leduc Poker, common benchmarks for Nash-equilibrium computation, have 6 and 144 states, respectively. RBT, designed as a smaller version of Reconnaissance Blind Chess, has roughly 10^7 states for x and 2×10^7 for o. To the best of our knowledge, our results are the first to demonstrate the feasibility of learning on RBT.

5.1 Benefit of Sharing Data

On the smaller Kuhn Poker game, we compare LUCB-T and LUCB-T-UNIFORM with vanilla LUCB [24], which treats each deterministic policy as a separate bandit arm, with no data being shared across arms. Table 1 shows the average stopping times when player x learns to play agains o's equilibrium strategy (complementary results are in appendix E). Notice that indeed LUCB is the most economical. The slack in the analysis of LUCB-T explains this observation; Kuhn Poker is a small game: with 3 cards there are only 64 policies. We devise a 5-card generalisation of Kuhn Poker (refer to appendix D) which has 1024 deterministic policies. On this variant, it becomes apparent that LUCB (whose sample complexity scales exponentially in the number of states) falls behind the "tree" variants. This shortcoming precludes the use of LUCB in larger games. Interestingly, we also notice that LUCB-T-UNIFORM, which outperforms LUCB-T on the 3-card version, performs worse on the 5-card version. Ignoring a lot of informative data hurts LUCB-T-UNIFORM in practice as problem sizes get larger.

5.2 Comparisons with Baselines on Larger Games

Much of the literature on exploration in MDPs has not empirically evaluated or provided code for algorithms, which, even if implemented, do not perform well with default hyperparameter settings. We were able to implement and fine-tune working versions of BPI-UCRL [53] and MDP-GAPE [22], which we use as baselines for comparison with our PAC algorithm, LUCB-T. For regret minimisation, the algorithms with the best instance-dependent theoretical bounds are MVP [59], AMB [57] and STRONGEULER [13], for which we were unable to source working implementations. However, algorithms in the games literature have been implemented and tested on a number of different applications. We compare UCB-T with MCCFR [32], On-path flipping (OPF) [16], and UCT [29]. Most of these baselines have hyperparameters to balance between exploration and exploitation. We use simplified confidence bounds in our implementations of LUCB-T, UCB-T, BPI-UCRL and MDP-GAPE. We have tuned these hyperparameters and bounds to the best of our ability for optimising performance. Implementation-related details are given in Appendix G.

Figure 2 shows performance plots for the larger games of Leduc Poker and RBT. In all the games we fix an ϵ-Nash policy for the opponent player, computed using CFR+ [50]. For the PAC setting, we follow the standard practice of showing the "value gap" $\mathbb{E}[V(\pi^\star) - V(\pi_1^t)]$ as learning proceeds [25], since stopping times are prohibitively large. For LUCB-T we use the same δ value for all our experiments. We observe consistently competitive performance for our algorithms across all three games for both players, for both the PAC and regret paradigms. Our algorithms scale well with game size, widening the gap between other algorithms on large problem instances, with RBT player o (Figure 2d) especially notable. For RBT, since the PAC algorithms BPI-UCRL and MDP-GAPE did not perform well, we have instead plotted a comparison with the regret minimising algorithms (figures 2c and 2d). Although our algorithms dominate all others that have provable performance guarantees, we notice that the popular UCT algorithm incurs lower regret on smaller problems. By publishing our code, we hope to attract more attention to the empirical evaluation of on-line learning algorithms for games, with the aim of reconciling gaps between theory and practice.

6 RELATED WORK AND DISCUSSION

While we are not aware of previous work specifically tailored to T-MDPs, there is indeed a vast literature on on-line learning in MDPs, as well as (imperfect-information) games. Unlike our contribution that benefits algorithms both for the PAC and the regret-minimisation settings, most previous work is tailored to one or the other. Specific goals have included *minimax PAC bounds* [12, 35], *instance-dependent PAC bounds* [22, 53, 54], *PAC RL with a generative model* [3, 23, 58], *reward-free exploration* [21, 26], *minimax regret bounds* [12, 59], and *instance-dependent regret bounds* [11, 13, 46].

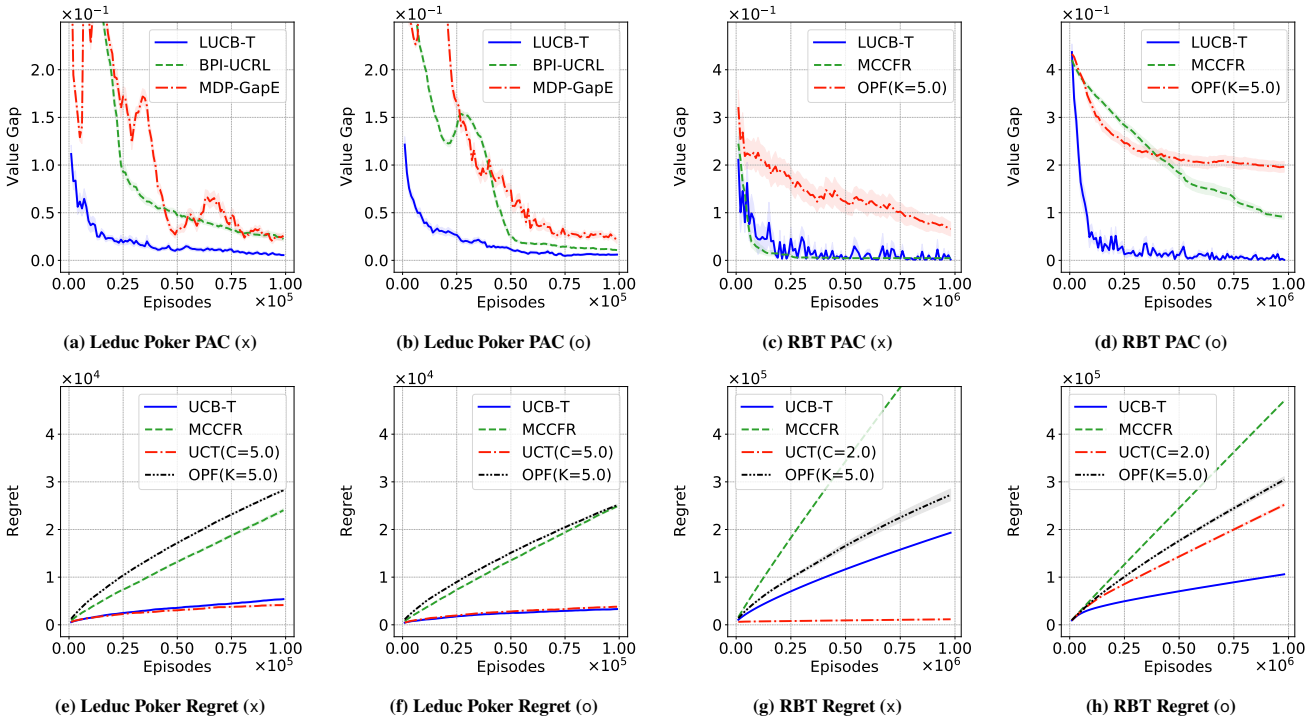
In the PAC setting, the theoretical results most relevant to ours have focused on furnishing sample-complexity upper bounds [22, 53, 55]. For these algorithms, the sample complexity upper bound typically takes the form of $\tilde{O}(C(\mathcal{M}, \epsilon) \log(1/\delta))$, where $C(\mathcal{M}, \epsilon)$ is an instance-dependent quantity capturing the hardness of learning. These bounds invariably have an inverse dependence on the probability that a state will be visited ($q(\sigma)$ in our work)—which can be arbitrarily small. Whereas our assumption of known rewards lets us avoid this dependence, available *lower bounds* suggest that it cannot be avoided in general MDPs [34]. It is also to be mentioned that upper bounds across different analyses remain incomparable to due varying definitions of the value gaps (Δ_σ in our work). We compile a brief technical survey of this line of results in Appendix F.

A related topic in this context pertains to "optimistic" algorithms, in which the sampling rule selects actions at each state greedily with respect to some UCB quantity. Wagenmaker et al. [55] show that optimistic algorithms fail to achieve the instance-optimal sample complexity bound, and that in general more aggressive exploration is necessary. As yet, we are unaware of a sample-complexity lower bound specific to T-MDPs.

Formal bounds on the *regret* of on-line learning algorithms for MDPs are of the form $C(\mathcal{M}) \log T$, where $C(\mathcal{M})$ is a quantity that typically depends on *sub-optimality gaps* as well as variance-related terms [13, 46]. An important result of Simchowitz and Jamieson [46] is that optimistic algorithms must incur an additional regret term that depends inversely on $\Delta_{\texttt{min}}$ —the minimum gap among all

Table 1: PAC stopping times for player x in 3-card and 5-card Kuhn Poker. Results average 10 runs, and show one standard error.

Algorithm	Experiment Parameters			
	3-card Kuhn Poker		5-card Kuhn Poker	
	player x, $\epsilon = 0.05, \delta = 0.05$	player x, $\epsilon = 0.1, \delta = 0.1$	player x, $\epsilon = 0.05, \delta = 0.05$	player x, $\epsilon = 0.1, \delta = 0.1$
LUCB	$\mathbf{0.473 \times 10^6} \pm 5.8\%$	$\mathbf{0.114 \times 10^6} \pm 7.3\%$	$7.947 \times 10^6 \pm 2.3\%$	$2.138 \times 10^6 \pm 2.2\%$
LUCB-T	$2.170 \times 10^6 \pm 5.4\%$	$0.531 \times 10^6 \pm 4.8\%$	$\mathbf{3.637 \times 10^6} \pm 3.7\%$	$\mathbf{0.942 \times 10^6} \pm 3.5\%$
LUCB-T-UNIFORM	$2.117 \times 10^6 \pm 4.5\%$	$0.508 \times 10^6 \pm 6.0\%$	$4.880 \times 10^6 \pm 8.7\%$	$1.194 \times 10^6 \pm 5.5\%$



(a) Leduc Poker PAC (x) (b) Leduc Poker PAC (o) (c) RBT PAC (x) (d) RBT PAC (o)

(e) Leduc Poker Regret (x) (f) Leduc Poker Regret (o) (g) RBT Regret (x) (h) RBT Regret (o)

Figure 2: Performance for Leduc Poker and RBT. The top row (a-d) corresponds to the PAC setting, and the bottom row (e-h) to regret minimisation results. All plots are averaged over 50 experiments for Leduc Poker and 25 experiments for RBT, showing outcomes for both players x and o. Error bars correspond to one standard error.

state-action pairs. Xu et al. [57] show that non-optimistic algorithms can reduce the unfavourable dependency on $\Delta_{\texttt{min}}$ by introducing the AMB algorithm, which eliminates this dependency in MDPs with a single optimal action at each state. Neither LUCB-T nor UCB-T is an optimistic algorithm. Although the latter is greedy with respect to a UCB, the UCB itself is for an entire policy (rather than a local state-action pair), computed using the PU procedure (Section 4.1, Appendix H).

The literature on imperfect-information extensive form games (IIEFGs) [30, 44] has predominantly focused on the two-player zero-sum setting. The most common solution concept for IIEFGs is the Nash equilibrium, which is an assignment of strategies to the players such that neither player can gain by unilaterally deviating. Since the exact computation of Nash equilibria is expensive, the preferred alternative is iterative approaches to compute ϵ-Nash equilibria. The most popular class of algorithms in this regard are from the regret minimisation paradigm. In particular, counterfactual regret minimisation (CFR) [60] and its variants [7, 8, 16, 32, 50] are state-of-the-art algorithms that have led to the development of superhuman agents for large IIEFGs such as Heads-Up Limit Texas Hold'em Poker and Heads-Up No Limit Texas Hold'em Poker [6, 9, 10, 38]. The problem we address in this paper is a degenerate type of game, in which one player is fixed, and so algorithms such as MCCFR [32] and OPF [16] are guaranteed to converge to the best response. This behaviour of theirs is indeed apparent from figures 2(e)-(h), but the rates of convergence are much slower than UCB-T, which has specifically been designed for T-MDPs. A theoretical counterpart to this empirical observation is that the regret and sample-complexity bounds from the games literature [4] are *worst-case* (in terms of ϵ) rather than instance-specific (in terms of value gaps Δ_σ).

7 CONCLUSION AND FUTURE WORK

In this paper, we have generalised well-known bandit algorithms LUCB and UCB to T-MDPs. Our main tool is a concentration inequality (the basis for Theorem 2) that facilitates the values of multiple policies (an exponentially-sized set) to be estimated simultaneously from a common pool of (polynomially-sized) data. The resulting confidence bound fits naturally into both LUCB and UCB, and enables polynomial (in time and memory) computation at each decision making step. Our confidence bound may be of independent interest for other analyses involving learning in sequential tasks.

We present theoretical support for our algorithms through upper bounds on sample complexity and regret. We also present supporting empirical results from three IIEFGs, on the task of learning a best response against a fixed opponent. RBT is an especially promising benchmark due to its much larger scale (10–20 million states) compared to current alternatives.

ACKNOWLEDGMENTS

We thank Manas Thakur for helping us parallelize our CFR+ implementation for RBT.

REFERENCES

[1] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration Tradeoffs. *J. Mach. Learn. Res.* 3 (2002), 397–422. https://jmlr.org/papers/v3/auer02a.html

[2] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* 47, 2-3 (2002), 235–256. https://doi.org/10.1023/A:1013689704352

[3] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. 2013. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Mach. Learn.* 91, 3 (2013), 325–349. https://doi.org/10.1007/S10994-013-5368-1

[4] Yu Bai, Chi Jin, Song Mei, and Tiancheng Yu. 2022. Near-optimal learning of extensive-form games with imperfect information. In *International Conference on Machine Learning*. PMLR, 1337–1382.

[5] Stephane Boucheron, Gabor Lugosi, and Pascal Massart. 2016. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.

[6] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2017. Heads-up limit hold'em poker is solved. *Commun. ACM* 60, 11 (oct 2017), 81–88. https://doi.org/10.1145/3131284

[7] Noam Brown, Christian Kroer, and Tuomas Sandholm. 2017. Dynamic thresholding and pruning for regret minimization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[8] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. Deep counterfactual regret minimization. In *International conference on machine learning*. PMLR, 793–802.

[9] Noam Brown and Tuomas Sandholm. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359, 6374 (2018), 418–424. https://doi.org/10.1126/science.aao1733 arXiv:https://www.science.org/doi/pdf/10.1126/science.aao1733

[10] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890.

[11] Shulun Chen, Runlong Zhou, Zihan Zhang, Maryam Fazel, and Simon S. Du. 2025. Sharp Gap-Dependent Variance-Aware Regret Bounds for Tabular MDPs. arXiv:2506.06521 [cs.LG] https://arxiv.org/abs/2506.06521

[12] Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. 2019. Policy Certificates: Towards Accountable Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 1507–1516. http://proceedings.mlr.press/v97/dann19a.html

[13] Chris Dann, Teodor V. Marinov, Mehryar Mohri, and Julian Zimmert. 2021. Beyond value-function gaps: improved instance-dependent regret bounds for episodic reinforcement learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21)*. Curran Associates Inc., Red Hook, NY, USA, Article 1, 12 pages.

[14] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR* abs/2501.12948 (2025). https://doi.org/10.48550/ARXIV.2501.12948 arXiv:2501.12948

[15] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2006. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *J. Mach. Learn. Res.* 7 (2006), 1079–1105. https://jmlr.org/papers/v7/evendar06a.html

[16] Gabriele Farina and Tuomas Sandholm. 2021. Model-free online learning in unknown sequential decision making problems and games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 5381–5390.

[17] Claude-Nicolas Fiechter. 1994. Efficient Reinforcement Learning. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory, COLT 1994, New Brunswick, NJ, USA, July 12-15, 1994*, Manfred K. Warmuth (Ed.). ACM, 88–97. https://doi.org/10.1145/180139.181019

[18] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. 2012. Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 3221–3229. https://proceedings.neurips.cc/paper/2012/hash/8b0d268963dd0cfb808aac48a549829f-Abstract.html

[19] Kevin Garbe and Jan Vondrak. 2018. Concentration of Lipschitz Functions of Negatively Dependent Variables. arXiv:1804.10084 [math.PR] https://arxiv.org/abs/1804.10084

[20] Ryan W. Gardner, Gino Perrotta, Anvay Shah, Shivaram Kalyanakrishnan, Kevin A. Wang, Gregory Clark, Timo Bertram, Johannes Fürnkranz, Martin Müller, Brady P. Garrison, Prithviraj Dasgupta, and Saeid Rezaei. 2022. The Machine Reconnaissance Blind Chess Tournament of NeurIPS 2022. In *Proceedings of the NeurIPS 2022 Competitions Track (Proceedings of Machine Learning Research, Vol. 220)*, Marco Ciccone, Gustavo Stolovitzky, and Jacob Albrecht (Eds.). PMLR, 119–132. https://proceedings.mlr.press/v220/gardner23a.html

[21] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. 2020. Reward-Free Exploration for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4870–4879. http://proceedings.mlr.press/v119/jin20d.html

[22] Anders Jonsson, Emilie Kaufmann, Pierre Menard, Omar Darwiche Domingues, Edouard Leurent, and Michal Valko. 2020. Planning in Markov Decision Processes with Gap-Dependent Sample Complexity. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1253–1263. https://proceedings.neurips.cc/paper_files/paper/2020/file/0d85eb24e2add96ff1a7021f83c1abc9-Paper.pdf

[23] Shivaram Kalyanakrishnan, Sheel Shah, and Santhosh Kumar Guguloth. 2025. A View of the Certainty-Equivalence Method for PAC RL as an Application of the Trajectory Tree Method. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1079–1087.

[24] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. 2012. PAC Subset Selection in Stochastic Multi-armed Bandits. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress. http://icml.cc/2012/papers/359.pdf

[25] Emilie Kaufmann and Shivaram Kalyanakrishnan. 2013. Information Complexity in Bandit Subset Selection. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA (JMLR Workshop and Conference Proceedings, Vol. 30)*, Shai Shalev-Shwartz and Ingo Steinwart (Eds.). JMLR.org, 228–251. http://proceedings.mlr.press/v30/Kaufmann13.html

[26] Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. 2021. Adaptive Reward-Free Exploration. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory (Proceedings of Machine Learning Research, Vol. 132)*, Vitaly Feldman, Katrina Ligett, and Sivan Sabato (Eds.). PMLR, 865–891. https://proceedings.mlr.press/v132/kaufmann21a.html

[27] Michael J. Kearns and Satinder Singh. 2002. Near-Optimal Reinforcement Learning in Polynomial Time. *Mach. Learn.* 49, 2-3 (2002), 209–232. https://doi.org/10.1023/A:1017984413808

[28] Jens Kober, J. Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *Int. J. Robotics Res.* 32, 11 (2013), 1238–1274. https://doi.org/10.1177/0278364913495721

[29] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 4212)*, Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou (Eds.). Springer, 282–293. https://doi.org/10.1007/11871842_29

[30] Harold W Kuhn. 1953. Extensive games and the problem of information. *Contributions to the Theory of Games* 2, 28 (1953), 193–216.

[31] Harold W Kuhn and Albert W Tucker. 1950. A Simplified Two-Person Poker. *Contributions to the Theory of Games, volume 1 of Annals of Mathematics Studies, 24, 97–103. Princeton, New Jersey: Princeton University Press* (1950).

[32] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. 2009. Monte Carlo sampling for regret minimization in extensive games. *Advances in neural information processing systems* 22 (2009).

[33] Tor Lattimore and Marcus Hutter. 2014. Near-optimal PAC bounds for discounted MDPs. *Theor. Comput. Sci.* 558 (2014), 125–143. https://doi.org/10.1016/J.TCS.2014.09.029

[34] Aymen Al Marjani, Andrea Tirinzoni, and Emilie Kaufmann. 2023. Towards Instance-Optimality in Online PAC Reinforcement Learning. *CoRR* abs/2311.05638 (2023). https://doi.org/10.48550/ARXIV.2311.05638 arXiv:2311.05638

[35] Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. 2021. Fast active learning for pure exploration in reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings*

of Machine Learning Research, Vol. 139), Marina Meila and Tong Zhang (Eds.). PMLR, 7599–7608. http://proceedings.mlr.press/v139/menard21a.html

[36] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR* abs/1312.5602 (2013). arXiv:1312.5602 http://arxiv.org/abs/1312.5602

[37] John E. Moody and Matthew Saffell. 1998. Reinforcement Learning for Trading. In *Advances in Neural Information Processing Systems 11, [NIPS Conference, Denver, Colorado, USA, November 30 - December 5, 1998]*, Michael J. Kearns, Sara A. Solla, and David A. Cohn (Eds.). The MIT Press, 917–923. http://papers.nips.cc/paper/1551-reinforcement-learning-for-trading

[38] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 6337 (2017), 508–513.

[39] Andrew Y. Ng, H. Jin Kim, Michael I. Jordan, and Shankar Sastry. 2003. Autonomous Helicopter Flight via Reinforcement Learning. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf (Eds.). MIT Press, 799–806. https://proceedings.neurips.cc/paper/2003/hash/b427426b8acd2c2e53827970f2c2f526-Abstract.html

[40] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

[41] Alessandro Panconesi and Aravind Srinivasan. 1997. Randomized Distributed Edge Coloring via an Extension of the Chernoff–Hoeffding Bounds. *SIAM J. Comput.* 26, 2 (1997), 350–368. https://doi.org/10.1137/S0097539793250767 arXiv:https://doi.org/10.1137/S0097539793250767

[42] Gino Perrotta, Ryan W. Gardner, Corey Lowman, Mohammad Taufeeque, Nitish Tongia, Shivaram Kalyanakrishnan, Gregory Clark, Kevin Wang, Eitan Rothberg, Brady P. Garrison, Prithviraj Dasgupta, Callum Canavan, and Lucas McCabe. 2022. The Second NeurIPS Tournament of Reconnaissance Blind Chess. In *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track (Proceedings of Machine Learning Research, Vol. 176)*, Douwe Kiela, Marco Ciccone, and Barbara Caputo (Eds.). PMLR, 53–65. https://proceedings.mlr.press/v176/perrotta22a.html

[43] Brian Sheppard. 2002. World-championship-caliber Scrabble SCRABBLE® is a registered trademark. All intellectual property rights in and to the game are owned in the USA by Hasbro Inc., in Canada by Hasbro Canada Corporation, and throughout the rest of the world by J.W. Spear & Sons Limited of Maidenhead, Berkshire, England, a subsidiary of Mattel Inc. *Artificial Intelligence* 134, 1 (2002), 241–275. https://doi.org/10.1016/S0004-3702(01)00166-7

[44] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

[45] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nat.* 550, 7676 (2017), 354–359. https://doi.org/10.1038/NATURE24270

[46] Max Simchowitz and Kevin Jamieson. 2019. Non-asymptotic gap-dependent regret bounds for tabular MDPs. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, Article 104, 10 pages.

[47] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and D. Chris Rayner. 2012. Bayes' Bluff: Opponent Modelling in Poker. *CoRR* abs/1207.1411 (2012). arXiv:1207.1411 http://arxiv.org/abs/1207.1411

[48] András Attila Sulyok and Kristóf Karacs. 2022. Towards Using Fully Observable Policies for POMDPs. arXiv:2207.11737 [cs.LG]

[49] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press. http://www.incompleteideas.net/book/the-book-2nd.html

[50] Oskari Tammelin. 2014. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042* (2014).

[51] Yuandong Tian, Qucheng Gong, and Yu Jiang. 2020. Joint Policy Search for Multi-agent Collaboration with Imperfect Information. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/e64f346817ce0c93d7166546ac8ce683-Abstract.html

[52] Andrea Tirinzoni, Aymen Al-Marjani, and Emilie Kaufmann. 2022. Near instance-optimal PAC reinforcement learning for deterministic MDPs. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 639, 14 pages.

[53] Andrea Tirinzoni, Aymen Al Marjani, and Emilie Kaufmann. 2023. Optimistic PAC Reinforcement Learning: the Instance-Dependent View. In *International Conference on Algorithmic Learning Theory, February 20-23, 2023, Singapore (Proceedings of Machine Learning Research, Vol. 201)*, Shipra Agrawal and Francesco Orabona (Eds.). PMLR, 1460–1480. https://proceedings.mlr.press/v201/tirinzoni23a.html

[54] Andrew Wagenmaker and Kevin G. Jamieson. 2022. Instance-Dependent Near-Optimal Policy Identification in Linear MDPs via Online Experiment Design. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/27bf08fe91a31495099a0b9febcc9592-Abstract-Conference.html

[55] Andrew J. Wagenmaker, Max Simchowitz, and Kevin Jamieson. 2022. Beyond No Regret: Instance-Dependent PAC Reinforcement Learning. In *Conference on Learning Theory, 2-5 July 2022, London, UK (Proceedings of Machine Learning Research, Vol. 178)*, Po-Ling Loh and Maxim Raginsky (Eds.). PMLR, 358–418. https://proceedings.mlr.press/v178/wagenmaker22a.html

[56] Peter R. Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J. Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Raj Kompella, HaoChih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D. Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. 2022. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nat.* 602, 7896 (2022), 223–228. https://doi.org/10.1038/S41586-021-04357-7

[57] Haike Xu, Tengyu Ma, and Simon Du. 2021. Fine-Grained Gap-Dependent Bounds for Tabular MDPs via Adaptive Multi-Step Bootstrap. In *Proceedings of Thirty Fourth Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 134)*, Mikhail Belkin and Samory Kpotufe (Eds.). PMLR, 4438–4472. https://proceedings.mlr.press/v134/xu21a.html

[58] Andrea Zanette, Mykel J. Kochenderfer, and Emma Brunskill. 2019. Almost Horizon-Free Structure-Aware Best Policy Identification with a Generative Model. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 5626–5635. https://proceedings.neurips.cc/paper/2019/hash/a724b9124acc7b5058ed75a31a9c2919-Abstract.html

[59] Zihan Zhang, Yuxin Chen, Jason D Lee, and Simon S Du. 2024. Settling the sample complexity of online reinforcement learning. In *Proceedings of Thirty Seventh Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 247)*, Shipra Agrawal and Aaron Roth (Eds.). PMLR, 5213–5219. https://proceedings.mlr.press/v247/zhang24a.html

[60] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. Regret minimization in games with incomplete information. *Advances in neural information processing systems* 20 (2007).