# Non-Asymptotic Analysis of Monte Carlo Tree Search

Devavrat Shah
devavrat@mit.edu
LIDS, MIT

Qiaomin Xie
qiaomin.xie@cornell.edu
Cornell University

Zhi Xu
zhixu@mit.edu
LIDS, MIT

## ABSTRACT

In this work, we consider the popular tree-based search strategy within the framework of reinforcement learning, the Monte Carlo Tree Search (MCTS), in the context of infinite-horizon discounted cost Markov Decision Process (MDP) with deterministic transitions. While MCTS is believed to provide an approximate value function for a given state with enough simulations, cf. [20, 21], the claimed proof of this property is incomplete. This is due to the fact that the variant of MCTS, the Upper Confidence Bound for Trees (UCT), analyzed in prior works utilizes "logarithmic" bonus term for balancing exploration and exploitation within the tree-based search, following the insights from stochastic multi-arm bandit (MAB) literature, cf. [1, 3]. In effect, such an approach assumes that the regret of the underlying recursively dependent non-stationary MABs concentrates around their mean exponentially in the number of steps, which is unlikely to hold as pointed out in [2], even for stationary MABs.

As the key contribution of this work, we establish polynomial concentration property of regret for a class of *non-stationary* multi-arm bandits. This in turn establishes that the MCTS with appropriate *polynomial* rather than *logarithmic* bonus term in UCB has the claimed property of [20, 21]. Interestingly enough, empirically successful approaches (cf. [33]) utilize a similar polynomial form of MCTS as suggested by our result. Using this as a building block, we argue that MCTS, combined with nearest neighbor supervised learning, acts as a "policy improvement" operator, i.e., it iteratively improves value function approximation for *all* states, due to combining with supervised learning, despite evaluating at only finitely many states. In effect, we establish that to learn an $\varepsilon$-approximation of the value function for deterministic MDPs with respect to $\ell_\infty$ norm, MCTS combined with nearest neighbor requires a sample size scaling as $\widetilde{O}\big(\varepsilon^{-(d+4)}\big)$, where $d$ is the dimension of the state space. This is nearly optimal due to a minimax lower bound of $\widetilde{\Omega}\big(\varepsilon^{-(d+2)}\big)$ [30] suggesting the strength of the variant of MCTS we propose here and our resulting analysis.[1]

## KEYWORDS

Monte Carlo Tree Search, Non Stationary Multi-Arm Bandit, Reinforcement Learning

## 1 INTRODUCTION

Monte Carlo Tree Search (MCTS) is a search framework for finding optimal decisions, based on the search tree built by random sampling of the decision space [8, 25]. MCTS has been extensively used in sequential decision makings that have a tree representation, exemplified by games and planning problems with deterministic transitions. In sequential decision making, the value of a state would typically depend on future actions. Therefore, to determine the best action for the given state, one has to take future actions into account and MCTS does this by simulating future via effectively expanding all possible future actions recursively in the form of (decision-like) tree. Viewing each state as a node and each action as an edge, simulating the future $H \geq 1$ steps would correspond to a search tree of depth $H$. In essence, the optimal action at the root of such a tree is then determined by finding the optimal path in the tree.

Since MCTS was first introduced, many variations and enhancements have been proposed. Recently, MCTS has been combined with deep neural networks for reinforcement learning, achieving remarkable success for games of Go [31, 33], chess and shogi [32]. In particular, AlphaGo Zero (AGZ) [33] employs supervised learning to iteratively learn a policy/value function (represented by a neural network) based on samples generated via MCTS —MCTS uses the neural network to estimate the value of leaf nodes (states) for simulation guidance; the neural network parameters are then updated with sample data generated by MCTS and further re-incorporated into tree search (i.e., as a leaf value estimator) in the next iteration of querying MCTS.

Despite the wide application and empirical success of MCTS, there is only limited work on theoretical guarantees of MCTS and its variants. A notable exception is the work of [20] and [21], which propose running tree search by applying the Upper Confidence Bound algorithm — originally designed for stochastic multi-arm bandit (MAB) problems [1, 3] — to each node of the tree. This leads to the so-called UCT (Upper Confidence Bounds for Trees) algorithm, which is one of the popular forms of MCTS. In [20], certain asymptotic optimality property of UCT is claimed. The proof therein is, however, incomplete, as we discuss in greater details in Section 1.2. More importantly, UCT as suggested in [20] requires exponential concentration of regret for the underlying non-stationary MAB. Here, non-stationary means that the reward distribution of each arm is time-varying rather than independent and identical as in the case of stationary MAB. Such exponential concentration of regret, however, is unlikely to hold in general even for stationary MAB as pointed out in [2].

Indeed, rigorous analysis of MCTS is subtle, even though its asymptotic convergence may seem natural. A key challenge is that the tree policy (e.g., UCT) for selecting actions typically needs to balance exploration and exploitation, so the action selection process at each node is non-stationary (non-uniform) across multiple simulations. A more severe difficulty arises due to the hierarchical/iterative structure of tree search, which induces complicated probabilistic dependency between a node and the nodes within its sub-tree. Specifically, as part of simulation within MCTS, at each intermediate node (or state), the action is chosen based on the outcomes of the past simulation steps within the sub-tree of the node in consideration. Such strong dependencies across time (i.e., depending on the history) and space (i.e., depending on the sub-trees downstream) among nodes makes the analysis non-trivial. The goal

[1]Technical report. This version: January 2020. The authors would like to thank the review team at ACM SIGMETRICS for their detailed, constructive feedback.

of this paper is to address this challenge and provide a rigorous theoretical foundation for MCTS. In particular, we are interested in the following:

- What is the appropriate form of MCTS for which the asymptotic convergence property claimed in the literature (cf. [20, 21]) holds?
- Can we rigorously establish the "strong policy improvement" property of MCTS when combined with supervised learning as observed in the literature (e.g., in [33])? If yes, what is the quantitative form of it?
- Does supervised learning combined with MCTS lead to the optimal policy, asymptotically? If so, what is its finite-sample (non-asymptotic) performance?

## 1.1 Preview of Main Results

As the main contribution of this work, we provide affirmative answers to all of the above questions. In what follows, we provide a brief overview of our contributions and results.

**Non-stationary MAB and recursive polynomial concentration.** In stochastic Multi Arm Bandit (MAB), the goal is to discover the action (arm) with the best average reward while choosing as few non-optimal actions as possible in the process. The rewards for any given action is assumed to be i.i.d., leading to the UCB algorithm: at any time $t \geq 1$, an action with maximal index is chosen where the index of an action is the empirical average reward observed for the action plus a *logarithmic* bonus term $B_{t,s}$ that scales as $\sqrt{\log t/s}$, for an action that has been picked $s \leq t$ times. As mentioned, Monte Carlo Tree Search (MCTS) has a similar goal where reward depends on the future actions. To take future actions into consideration, MCTS effectively expands all possible future actions recursively in the form of (decision-like) tree. As such, determining the optimal future path corresponding to maximal reward starting at the root node of the MCTS tree requires solving multiple MABs, one per each intermediate node within the tree. Apart from the MABs associated with the leaf layer of the tree, all the MABs associated with the intermediate nodes turn out to have rewards that are generated by MAB algorithms for nodes downstream. This creates complicated, hierarchically inter-dependent MABs.

To determine the appropriate, UCB-like algorithm for MAB corresponding to each node of the MCTS tree, it is essential to understand the concentration property of rewards, i.e., concentration of regret for MABs associated with the nodes downstream. While the rewards at leaf level may enjoy exponential concentration due to independence, the regret of any algorithm even for such an MAB is unlikely to have exponential concentration in general, cf. [2, 26]. Further, the MAB of our interest has non-stationary rewards due to strong dependence across hierarchy. Indeed, an oversight of this complication led [20, 21] to suggest UCT inspired by the standard UCB algorithm for MABs with stationary, independent rewards.

As an important contribution of this work, we formulate an appropriate form of non-stationary MAB which correctly models the MAB at each node within the tree. In particular, assuming that the rewards, though non-stationary, satisfy certain *polynomial* concentration property. Then, we establish that under the UCB algorithm that chooses the arm with highest index, where index is defined as the empirical observed reward plus an appropriate *polynomial* (and *not exponential*) bonus term, a similar form of *polynomial* concentration holds for the induced regret. In particular, let $\bar{X}_t$ denote the empirical average of the rewards collected at a given node over $t$ visits of the node. Then, under the UCB algorithm with a bonus term that scale as $t^{\eta(1-\eta)}/s^{1-\eta}$, where $1/2 \leq \eta < 1$, we establish that (a) $\bar{X}_t$ converges to the optimal mean reward obtained by choosing the right action, and (b) $\bar{X}_t$ satisfy a polynomial concentration inequality around the optimal mean reward, i.e., the convergence rate is polynomial. The precise statement can be found as Theorem 3 in Section 5.

**Corrected UCT for MCTS and non-asymptotic analysis.** As desired, the non-stationary MAB enjoys a recursive polynomial concentration: starting from polynomially concentrated arm rewards, the proper UCB algorithm leads to a polynomially concentrated empirical reward. Hence, we immediately obtain that we can recursively define the UCB algorithm at each level in MCTS, starting from the leaf level, with appropriately chosen polynomial bonus terms $B_{t,s}$. In effect, setting $\eta = 1/2$, we obtain modified UCT where $B_{t,s}$ scales as $t^{1/4}/s^{1/2}$. This is in contrast to the $\sqrt{\log t/s}$ scaling in the standard UCB as well as UCT suggested in the literature, cf. [20, 21].

By recursively applying the convergence and concentration property of the non-stationary MAB for the resulting algorithm for MCTS, we establish that for any query state $s$ of a MDP with deterministic transitions, using a total of $n$ simulations of the MCTS, we can obtain a value function estimation within error $\delta\varepsilon_0 + O\big(n^{-1/2}\big)$ for some $\delta < 1$ (independent of $n$ but dependent on the depth of MCTS tree), if we start with a value function estimation for all the leaf nodes within error $\varepsilon_0$ . That is, MCTS is indeed asymptotically correct as was conjectured in the prior literature. For details, see Theorem 1 in Section 3.

**MCTS with supervised learning, strong policy improvement, and near optimality.** The result stated above for MCTS implies its "bootstrapping" property – if we start with a value function estimation for *all* state within error $\varepsilon$, then MCTS can produce estimation of value function for a *given query* state within error less than $\varepsilon$ with enough simulations. By coupling such improved estimations for a number of query states, combined with expressive enough supervised learning, one can hope to generalize such improved estimations of value function for *all* states. That is, MCTS coupled with supervised learning can be "strong policy improvement operator".

Indeed, this is precisely what we establish by utilizing nearest neighbor supervised learning. Specifically, we establish that with total of $\widetilde{O}\big(\frac{1}{\varepsilon^{4+d}}\big)$ number of samples, MCTS with nearest neighbor finds an $\varepsilon$-approximation of the optimal value function for deterministic MDPs with respect to $\ell_\infty$-norm; here $d$ is the dimension of the state space. This is nearly optimal in view of a minimax lower bound of $\widetilde{\Omega}\big(\frac{1}{\varepsilon^{2+d}}\big)$ [30]. For details, see Theorem 2 in Section 4.

**An Implication.** As mentioned earlier, the modified UCT policy per our result suggests using bonus term $B_{t,s}$ that scales as $t^{1/4}/s^{1/2}$ at each node within the MCTS. Interestingly enough, the empirical results of AGZ are obtained by utilizing $B_{t,s}$ that scales as $t^{1/2}/s$. This is qualitatively similar to what our results suggests and in contrast to the classical UCT.

**Summary.** In summary, our contributions are:

◦ We introduce a class of non-stationary MAB problems and establish the polynomial concentration property of the regret under an appropriate UCB algorithm which should be of independent interest.

◦ As a consequence, we suggest correction for UCT and establish the asymptotic correctness of thus modified MCTS, where the bonus term scale *polynomially* in contrast to *logarithmically* as was believed in the literature.

◦ Building on the above results, we establish that MCTS combined with supervised learning learns value function within $\varepsilon$ error, under a near optimal sample complexity of $\widetilde{O}(\varepsilon^{-d-4})$.

◦ Interesting enough, our result suggests qualitatively similar polynomial bonus term as the ones used in AlphaGo Zero.

## 1.2 Related work

Reinforcement learning [38] aims to approximate the optimal value function and policy directly from the observed data. A variety of algorithms have been developed for the so called tabular cases [43], as well as using functional approximation such as linear architectures [37]. More recent work approximates the value function/policy by deep neural networks [24, 28, 29, 33, 44], which can be trained using temporal-difference learning or Q-learning [22, 23, 42].

MCTS is an alternative approach, which as discussed, estimates the (optimal) value of states by building a search tree from Monte-Carlo simulations [8, 9, 11, 20]. [20] and [21] argue for the asymptotic convergence of MCTS with standard UCT. However, the proof is incomplete [39]. A key step towards proving the claimed result is to show the convergence and concentration properties of the regret for UCB under non-stationary reward distributions. In particular, to establish an exponential concentration of regret (Theorem 5, [21]), Lemma 14 is applied. However, it requires conditional independence of $\{Z_i\}$ sequence, which does not hold, hence making the conclusion of exponential concentration questionable. Therefore, the proof of the main result (Theorem 7, [21]), which applies Theorem 5 with an inductive argument, is incorrect as stated.

In fact, it may be infeasible to prove Theorem 5 in [21] as stated. For example, the work of [2] shows that for bandit problems, the regret under UCB concentrates around its expectation polynomially and *not exponentially* as desired in [21]. Further, [26] prove that for any strategy that does not use the knowledge of time horizon, it is infeasible to improve this polynomial concentration and establish exponential concentration. Our result is consistent with these fundamental bound of stationary MAB — we establish polynomial concentration of regret for non-stationary MAB, which plays a crucial role in our analysis of MCTS. Also see the work [25] for a discussion of the issues with logarithmic bonus terms for tree search.

While we focus on UCT in this paper, we note that there are other variants of MCTS developed for a diverse range of applications [10, 27, 36]. We refer to the survey on MCTS [8] for other variations and applications. Additionally, it is important to mention the work of [9] that explores the idea of using UCB for adaptive sampling in MDPs. However, their algorithm proceeds in a depth-first, recursive manner, and hence involves using UCB for a stationary MAB at each node. In contrast, the UCT algorithm we study involves non-stationary MABs, hence our analysis is significantly different from theirs. We refer the readers to the work by [20] and [11] for further discussion of this difference. Two other closely related papers are [40] and [19], which study a simplified MCTS for two-player zero-sum games. Compared to classical MCTS (e.g., UCT), both the setting and the algorithms are simpler: the game tree is given in advance, rather than being built gradually through samples; the algorithm in [40] operates on the tree in a bottom-up fashion with uniform sampling at the leaf nodes. As a result, the analysis is significantly simpler and it is unclear whether the techniques can be extended to analyze other variants of MCTS.

More recently, it has become popular to combine MCTS with neural network in deep reinforcement learning [4, 31–33]. In terms of theoretical results of MCTS-based reinforcement learning, the closest work to ours is [18]. The key algorithmic difference from ours lies in the leaf-node evaluator of the search tree: they use a combination of an estimated value function and an estimated policy. The latest observations at the root node are then used to update the value and policy functions (leaf-node evaluator) for the next iteration. They also give a finite sample analysis. However, their result and ours are quite different: in their analysis, the sample complexity of MCTS, as well as the approximation power of value/policy architectures, are *imposed as an assumption*; here, we *prove* an explicit finite-sample bound for MCTS and characterize the non-asymptotic error prorogation under MCTS with non-parametric regression for leaf-node evaluation. Therefore, they *do not* establish "strong policy improvement" property of the MCTS.

Finally, we remark that the iterative reinforcement learning algorithm of combining MCTS with supervised learning analyzed in this paper is motivated by AlphaGo Zero (AGZ) [33]. The theoretical results partly provide an affirmative support for the validity of this empirically successful approach. Nonetheless, we note that the considered algorithm does not capture all the ingredients of AGZ — there are important aspects of AGZ that require future investigations. In particular, AGZ contains both a value network and a policy network, implemented via deep neural networks. While the bonus terms $B_{t,s}$ in MCTS of AGZ scales polynomially as well, $B_{t,s}$ also incorporates current predictions from the policy network to guide the selection of actions during the simulation. In addition, the AGZ algorithm updates the networks incrementally, performing stochastic gradient descent after every few steps. In contrast, the value estimation in this work is updated only after each full iteration: we update the value function via nearest neighbor regression, after obtaining enough samples that properly cover the state space. A comprehensive analysis on AGZ that includes these ingredients and innovations is an important future direction towards fully explaining its empirical success.

## 1.3 Organization

Section 2 describes the setting of Markov Decision Process considered in this work. Section 3 describes the Monte Carlo Tree Search algorithm and the main result about its non-asymptotic analysis. Section 4 describes a reinforcement learning method that combines the Monte Carlo Tree Search with nearest neighbor supervised learning. It describes the finite-sample analysis of the method for finding $\varepsilon$ approximate value function with respect to $\ell_\infty$ norm. Section 5 introduces a form of non-stationary multi-arm bandit and an upper confidence bound policy for it. For this setting, we present the concentration of induced regret which serves as a key result for

establishing the property of MCTS. The proofs of all the technical results are delegated to Sections 6, 7 and Appendices.

## 2 SETUP AND PROBLEM STATEMENT

**Formal Setup.** We consider the setup of discrete-time discounted Markov decision process (MDP). An MDP is described by a five-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{P} \equiv \mathcal{P}(s'|s,a)$ is the Markovian transition kernel, $\mathcal{R} \equiv \mathcal{R}(s,a)$ is a random reward function, and $\gamma \in (0,1)$ is a discount factor. At each time step, the system is in some state $s \in \mathcal{S}$. When an action $a \in \mathcal{A}$ is taken, the state transits to a next state $s' \in \mathcal{S}$ according to the transition kernel $\mathcal{P}$ and an immediate reward is generated according to $\mathcal{R}(s,a)$.

A stationary policy $\pi(a|s)$ gives the probability of performing action $a \in \mathcal{A}$ given the current state $s \in \mathcal{S}$. The *value* function for each state $s \in \mathcal{S}$ under policy $\pi$, denoted by $V^\pi(s)$, is defined as the expected discounted sum of rewards received following the policy $\pi$ from initial state $s$, i.e.,

$$V^\pi(s) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) | s_0 = s\Big].$$

The goal is to find an optimal policy $\pi^*$ that maximizes the value from each initial state. The optimal value function $V^*$ is defined as $V^*(s) = V^{\pi^*}(s) = \sup_\pi V^\pi(s)$, $\forall s \in \mathcal{S}$. It is well understood that such an optimal policy exists in reasonable generality. In this paper, we restrict our attention to the MDPs with the following assumptions:

Assumption 1 (MDP Regularity). *(A1.) The action space $\mathcal{A}$ is a finite set and the state space $\mathcal{S}$ is a compact subset of $d$ dimensional set; without loss of generality, let $\mathcal{S} = [0,1]^d$; (A2.) The immediate rewards are random variables, uniformly bounded such that $\mathcal{R}(s,a) \in [-R_{\max}, R_{\max}]$, $\forall s \in \mathcal{S}, a \in \mathcal{A}$ for some $R_{\max} > 0$; (A3.) The state transitions are deterministic, i.e. $\mathcal{P} \equiv \mathcal{P}(s'|s,a) \in \{0,1\}$ for all $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$.*

Define $\beta \triangleq 1/(1-\gamma)$ and $V_{\max} \triangleq \beta R_{\max}$. Since all the rewards are bounded by $R_{\max}$, it is easy to see that the absolute value of the value function for any state under any policy is bounded by $V_{\max}$ [14, 35].

**On Deterministic Transition.** We note that the deterministic transition in MDP should not be viewed as restriction or assumption. Traditional AI game research has been focused on deterministic games with a tree representation. It is this context within which historically MCTS was introduced, has been extensively studied and utilized in practice [8]. This includes the recent successes of MCTS in Go [33], Chess [32] and Atari games [15]. There is a long theoretical literature on the analysis of MCTS and related methods [5, 8, 17, 25] that considers deterministic transition. The principled extension of MCTS algorithm itself as well as theoretical results similar to ours for the stochastic setting are important future work. In particular, such extension would require non-trivial work. As to be described in the next section, for the Monte Carlo tree search variant, each edge is treated as an action, leading to a node (next state) at the next level. With stochastic transitions, each action could lead to several potential next states. That is, at a particular level, each action would necessarily be associated with several edges, connecting a single node at the current level to multiple nodes at the next level. Principally constructing such a fixed-depth tree, and subsequently aggregating the statistics for each action and designing/analyzing the resulting UCB-style algorithm on the tree are highly non-trivial. It is not immediate that the current results would easily extend to such scenarios.

**Value Function Iteration.** A classical approach to find optimal value function, $V^*$, is an iterative approach called value function iteration. The Bellman equation characterizes the optimal value function as

$$V^*(s) = \max_{a \in \mathcal{A}} \Big(\mathbb{E}[\mathcal{R}(s,a)] + \gamma V^*(s \circ a)\Big), \tag{1}$$

where $s \circ a \in \mathcal{S}$ is the notation to denote the state reached by applying action $a$ on state $s$. Under Assumption 1, the transitions are deterministic and hence $s \circ a$ represents a single, deterministic state rather than a random state.

The value function iteration effectively views (1) as a fixed point equation and tries to find a solution to it through a natural iteration. Precisely, let $V^{(t)}(\cdot)$ be the value function estimation in iteration $t$ with $V^{(0)}$ being arbitrarily initialized. Then, for $t \geq 0$, for all $s \in \mathcal{S}$,

$$V^{(t+1)}(s) = \max_{a \in \mathcal{A}} \Big(\mathbb{E}[\mathcal{R}(s,a)] + \gamma V^{(t)}(s \circ a)\Big). \tag{2}$$

It is well known (cf. [7]) that value iteration is contractive with respect to $\|\cdot\|_\infty$ norm for all $\gamma < 1$. Specifically, for $t \geq 0$, we have

$$\|V^{(t+1)} - V^*\|_\infty \leq \gamma \|V^{(t)} - V^*\|_\infty. \tag{3}$$

## 3 MONTE CARLO TREE SEARCH

Monte Carlo Tree Search (MCTS) has been quite popular recently in many reinforcement learning tasks. In effect, given a state $s \in \mathcal{S}$ and a value function estimate $\hat{V}$, it attempts to run the value function iteration for a fixed number of steps, say $H$, to evaluate $V^{(H)}(s)$ starting with $V^{(0)} = \hat{V}$ per (2). This, according to (3), would provide an estimate within error $\gamma^H \|\hat{V} - V^*\|_\infty$ — an excellent estimate of $V^*(s)$ if $H$ is large enough. The goal is to perform computation for value function iteration necessary to evaluate $V^{(H)}$ for state $s$ only and not necessarily for all states as required by traditional value function iteration. MCTS achieves this by simply 'unrolling' the associated 'computation tree'. Another challenge that MCTS overcomes is the fact that value function iteration as in (2) assumes knowledge of model so that it can compute $\mathbb{E}[\mathcal{R}(\cdot,\cdot)]$ for any state-action pair. But in reality, rewards are observed through samples, not a direct access to $\mathbb{E}[\mathcal{R}(\cdot,\cdot)]$. MCTS tries to utilize the samples in a careful manner to obtain accurate estimation for $V^{(H)}(s)$ over the computation tree suggested by the value function iteration as discussed above. The concern of careful use of samples naturally connects it to multi-arm bandit like setting.

Next, we present a detailed description of the MCTS algorithm in Section 3.1. This can be viewed as a *correction* of the algorithm presented in [20, 21]. We state its theoretical property in Section 3.2.

### 3.1 Algorithm

We provide details of a specific form of MCTS, which replaces the logarithmic bonus term of UCT with a polynomial one. Overall, we fix the search tree to be of depth $H$. Similar to most literature on this topic, it uses a variant of the Upper Confidence Bound (UCB) algorithm to select an action at each stage. At a leaf node (i.e., a state

at depth $H$), we use the current value oracle $\hat{V}$ to evaluate its value. Note that since we consider deterministic transitions, consequently, the tree is fixed once the root node (state) is chosen, and we use the notation $s \circ a$ to denote the next state after taking action $a$ at state $s$. Each edge represents a state-action pair, while each node represents a state. For clarity, we use superscript to distinguish quantities related to different depth. The pseudocode for the MCTS procedure is given in Algorithm 1, and Figure 1 shows the structure of the search tree and related notation.

**Algorithm 1** Fixed-Depth Monte Carlo Tree Search

1: **Input:** (1) current value oracle $\hat{V}$, root node $s^{(0)}$ and search depth $H$; (2) number of MCTS simulations $n$; (3) algorithmic constants, $\{\alpha^{(i)}\}_{i=1}^{H}$, $\{\beta^{(i)}\}_{i=1}^{H}$, $\{\xi^{(i)}\}_{i=1}^{H}$ and $\{\eta^{(i)}\}_{i=1}^{H}$.
2: **Initialization:** for each depth $h$, initialize the cumulative node value $\tilde{v}^{(h)}(s) = 0$ and visit count $N^{(h)}(s) = 0$ for every node $s$ and initialize the cumulative edge value $q^{(h)}(s, a) = 0$.
3: **for** each MCTS simulation $t = 1, 2, \ldots, n$ **do**
4: `/* Simulation: select actions until reaching depth H */`
5: **for** depth $h = 0, 1, 2, \ldots, H-1$ **do**
6: at state $s^{(h)}$ of depth $h$, select an action (edge) according to

$$a^{(h+1)} = \arg\max_{a \in \mathcal{A}} \left\{ \frac{q^{(h+1)}(s^{(h)}, a) + \gamma \tilde{v}^{(h+1)}(s^{(h)} \circ a)}{N^{(h+1)}(s^{(h)} \circ a)} + \frac{\left(\beta^{(h+1)}\right)^{1/\xi^{(h+1)}} \cdot \left(N^{(h)}(s^{(h)})\right)^{\alpha^{(h+1)}/\xi^{(h+1)}}}{\left(N^{(h+1)}(s^{(h)} \circ a)\right)^{1-\eta^{(h+1)}}} \right\}, \quad (4)$$

where dividing by zero is assumed to be $+\infty$.
7: upon taking the action $a^{(h+1)}$, receive a random reward $r^{(h+1)} \triangleq \mathcal{R}(s^{(h)}, a^{(h+1)})$ and transit to a new state $s^{(h+1)}$ at depth $h+1$.
8: **end for**
9: `/* Evaluation: call value oracle for leaf nodes */`
10: reach $s^{(H)}$ at depth $H$, call the current value oracle and let $\tilde{v}^{(H)}(s^{(H)}) = \hat{V}(s^{(H)})$.
11: `/* Update Statistics: quantities on the search path */`
12: **for** depth $h = 0, 1, 2, \ldots, H-1$ **do**
13: update statistics of nodes and edges that are on the search path of current simulation:
14: visit count:

$$N^{(h+1)}(s^{(h+1)}) = N^{(h+1)}(s^{(h+1)}) + 1$$

15: edge value:

$$q^{(h+1)}(s^{(h)}, a^{(h+1)}) = q^{(h+1)}(s^{(h)}, a^{(h+1)}) + r^{(h+1)}$$

16: node value:

$$\tilde{v}^{(h)}(s^{(h)}) = \tilde{v}^{(h)}(s^{(h)}) + r^{(h+1)} + \gamma r^{(h+2)} + \cdots + \gamma^{H-1-h} r^{(H)} + \gamma^{H-h} \tilde{v}^{(H)}(s^{(H)})$$

17: **end for**
18: **end for**
19: **Output:** average of the value for the root node $\tilde{v}^{(0)}(s^{(0)})/n$.

In Algorithm 1, there are certain sequences of algorithmic parameters required, namely, $\alpha$, $\beta$, $\xi$ and $\eta$. The choices for these constants will become clear in our non-asymptotic analysis. At a higher level, the constants for the last layer (i.e., depth $H$), $\alpha^{(H)}$, $\beta^{(H)}$, $\xi^{(H)}$ and $\eta^{(H)}$ depend on the properties of the leaf nodes, while the rest are recursively determined by the constants one layer below. We note that in selecting action $a^{(h+1)}$ at each depth $h$ (i.e., Line 6 of Algorithm 1), the upper confidence term is polynomial in $n$ while a typical UCB algorithm would be logarithmic in $n$, where $n$ is the number of visits to the corresponding state thus far. The logarithmic factor in the original UCB algorithm was motivated by the exponential tail probability bounds. In our case, it turns out that exponential tail bounds for each layer seems to be infeasible without further structural assumptions. As mentioned in Section 1.2, prior work [2, 26] has justified the polynomial concentration of the regret for the classical UCB in stochastic (independent rewards) multi-arm bandit setting. This implies that the concentration at intermediate depth (i.e., depth less than $H$) is at most polynomial. Indeed, we will prove these polynomial concentration bounds even for non-stationary (dependent, non-stationary rewards) multi-arm bandit that show up in MCTS and discuss separately in Section 5.

## 3.2 Analysis

Now, we state the following result on the non-asymptotic performance of the MCTS as described above.

THEOREM 1. *Consider an MDP satisfying Assumption 1. Let $H \geq 1$, and for $1/2 \leq \eta < 1$, let*

$$\eta^{(h)} = \eta^{(H)} \equiv \eta, \quad \forall\, h \in [H], \quad (5)$$

$$\alpha^{(h)} = \eta(1-\eta)\left(\alpha^{(h+1)} - 1\right), \quad \forall\, h \in [H-1], \quad (6)$$

$$\xi^{(h)} = \alpha^{(h+1)} - 1, \quad \forall h \in [H-1]. \quad (7)$$

*Suppose that a large enough $\xi^{(H)}$ is chosen such that $\alpha^{(1)} > 2$. Then, there exist corresponding constants $\{\beta^{(i)}\}_{i=1}^{H}$ such that for each query state $s \in \mathcal{S}$, the following claim holds for the output $\hat{V}_n(s)$ of MCTS with n simulations:*

$$\left| \mathbb{E}\left[\hat{V}_n(s)\right] - V^*(s) \right| \leq \gamma^H \varepsilon_0 + O\left(n^{\eta-1}\right),$$

*where $\varepsilon_0 = \|\hat{V} - V^*\|_\infty$ with $\hat{V}$ being the estimate of $V^*$ utilized by the MCTS algorithm for leaf nodes.*

Since $\eta \in [1/2, 1)$, Theorem 1 implies a best case convergence rate of $O(n^{-1/2})$ by setting $\eta = 1/2$. With these parameter choices, the bias term in the upper confidence bound (line 6 of Algorithm 1) scales as $\left(N^{(h)}(s^{(h)})\right)^{1/4} / \sqrt{N^{(h+1)}(s^{(h)} \circ a)}$, that is, in the form of $t^{1/4}/\sqrt{S}$ as mentioned in the introduction, where $t \equiv N^{(h)}(s^{(h)})$ is the number of times that state $s^{(h)}$ at depth $h$ has been visited, and $S \equiv N^{(h+1)}(s^{(h)} \circ a)$ is the number of times action $a$ has been selected at state $s^{(h)}$.

# 4 REINFORCEMENT LEARNING THROUGH MCTS WITH SUPERVISED LEARNING

Recently, MCTS has been utilized prominently in various empirical successes of reinforcement learning including AlphaGo Zero (AGZ). Here, MCTS is combined with expressive supervised learning method to iteratively improve the policy as well as the value

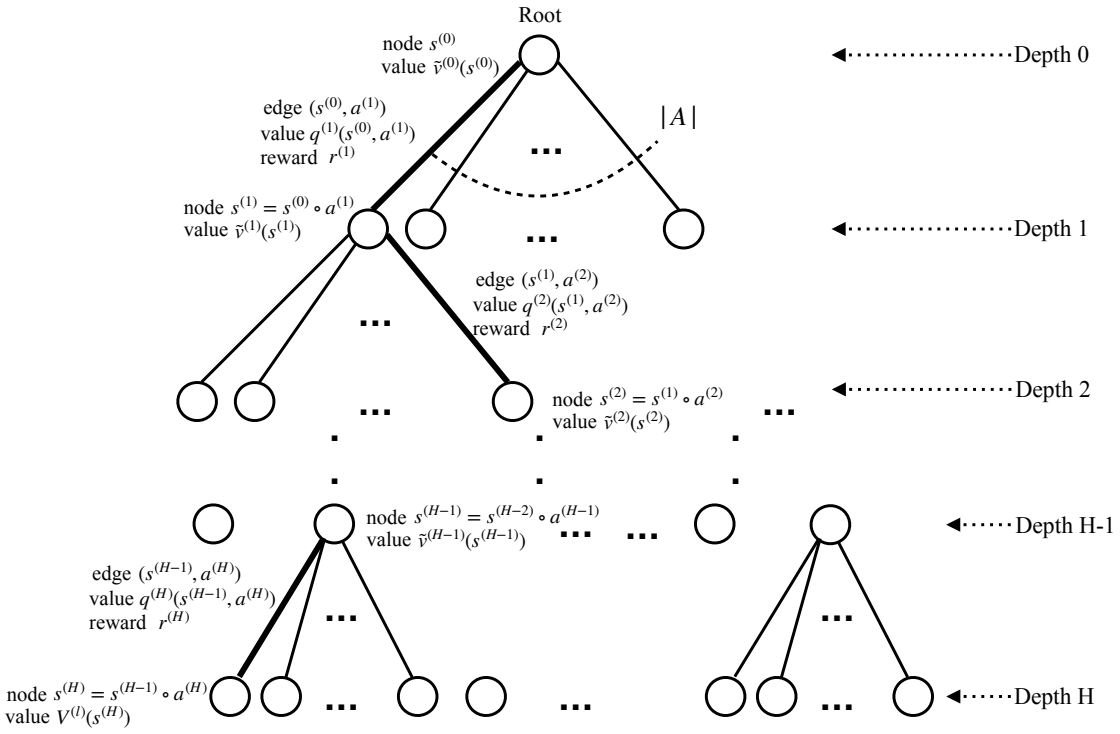


**Figure 1: Notation and a sample simulation path of MCTS (thick lines).**

function estimation. In effect, MCTS combined with supervised learning acts as a "policy improvement" operator.

Intuitively, MCTS produces an improved estimation of value function for a given state of interest, starting with a given estimation of value function by "unrolling" the "computation tree" associated with value function iteration. And MCTS achieves this using observations obtained through simulations. Establishing this improvement property rigorously was the primary goal of Section 3. Now, given such improved estimation of value function for finitely many states, a good supervised learning method can learn to generalize such an improvement to all states. If so, this is like performing value function iteration, but using simulations. Presenting such a policy and establishing such guarantees is the crux of this section.

To that end, we present a reinforcement learning method that combines MCTS with nearest neighbor supervised learning. For this method, we establish that indeed, with sufficient number of samples, the resulting policy improves the value function estimation just like value function iteration. Using this, we provide a finite-sample analysis for learning the optimal value function within a given tolerance. We find it nearly matching a minimax lower bound in [30] which we recall in Section 4.4, and thus establishes near minimax optimality of such a reinforcement learning method.

## 4.1 Reinforcement Learning Policy

Here we describe the policy to produce estimation of optimal value function $V^*$. Similar approach can be applied to obtain estimation of policy as well. Let $V^{(0)}$ be the initial estimation of $V^*$, and for simplicity, let $V^{(0)}(\cdot) = 0$. We describe a policy that iterates between use of MCTS and supervised learning to iteratively obtain estimation $V^{(\ell)}$ for $\ell \geq 1$, so that iteratively better estimation of $V^*$ is produced as $\ell$ increases. To that end, for $\ell \geq 1$,

- For appropriately sampled states $S^\ell = \{s_i\}_{i=1}^{m_\ell}$, apply MCTS to obtain improved estimations of value function $\{\hat{V}^{(\ell)}(s_i)\}_{i=1}^{m_\ell}$ using $V^{(\ell-1)}$ to evaluate leaf nodes during simulations.
- Using $\{(s_i, \hat{V}^{(\ell)}(s_i)\}_{i=1}^{m_\ell}$ with a variant of nearest neighbor supervised learning with parameter $\delta_\ell \in (0, 1)$, produce estimation $V^{(\ell)}$ of the optimal value function.

For completeness, the pseudo-code is provided in Algorithm 2.

## 4.2 Supervised Learning

For simplicity, we shall utilize the following variant of the nearest neighbor supervised learning parametrized by $\delta \in (0, 1)$. Given state space $\mathcal{S} = [0, 1]^d$, we wish to cover it with minimal (up to scaling) number of balls of radius $\delta$ (with respect to $\ell_2$-norm). To that end, since $\mathcal{S} = [0, 1]^d$, one such construction is where we have balls of radius $\delta$ with centers being $\{(\theta_1, \theta_2, \ldots, \theta_d) : \theta_1, \ldots, \theta_d \in \mathcal{Q}(\delta)\}$ where

$$\mathcal{Q}(\delta) = \left\{\frac{1}{2}\delta i \,:\, i \in \mathbb{Z}, 0 \leq i \leq \left\lfloor \frac{2}{\delta} \right\rfloor\right\} \cup \left\{1 - \frac{1}{2}\delta i \,:\, i \in \mathbb{Z}, 0 \leq i \leq \left\lfloor \frac{2}{\delta} \right\rfloor\right\}.$$

Let the collection of these balls be denoted by $c_1, \ldots, c_{K(\delta,d)}$ with $K(\delta, d) = |\mathcal{Q}(\delta)|$. It is easy to verify that $\mathcal{S} \subset \cup_{i \in [K(\delta,d)]} c_i$, $K(\delta, d) = \Theta(\delta^{-d})$ and $C_d \delta^d \leq \text{volume}(c_i \cap \mathcal{S}) \leq C'_d \delta^d$ for strictly positive constants $C_d, C'_d$ that depends on $d$ but not $\delta$. For any $s \in \mathcal{S}$, let $j(s) = \min\{j : s \in c_j\}$. Given observations $\{(s_i, \hat{V}^{(\ell)}(s_i)\}_{i=1}^{m_\ell}$, we

**Algorithm 2** Reinforcement Learning Policy

**Input:** initial value function oracle $V^{(0)}(s) = 0, \forall\, s \in \mathcal{S}$
**for** $l = 1, 2, \ldots, L$ **do**
  /* improvement via MCTS */
  uniformly and independently sample states $S^{\ell} = \{s_i\}_{i=1}^{m_\ell}$.
  **for** each sampled state $s_i$ **do**
    apply the MCTS algorithm, which takes as inputs the current value oracle $V^{(l-1)}$, the depth $H^{(l)}$, the number of simulation $n_l$, and the root node $s_i$, and outputs an improved estimate for $V^*(s_i)$:

$$\hat{V}^{(l)}(s_i) = \text{MCTS}\big(V^{(l-1)}, H^{(l)}, n_l, s_i\big) \tag{8}$$

  **end for**
  /* supervised learning */
  with the collected data $\mathcal{D}^{(l)} = \{(s_i, \hat{V}^{(l)}(s_i))\}_{i=1}^{m_l}$, build a new value oracle $V^{(l)}$ via a nearest neighbor regression with parameter $\delta_l$ :

$$V^{(l)}(s) = \text{Nearest Neigbhor}\big(\mathcal{D}^{(l)}, \delta_l, s\big),\ \forall\, s \in \mathcal{S}. \tag{9}$$

**end for**
**Output:** final value oracle $V^{(L)}$.

produce an estimate $V^{(\ell)}(s)$ for all $s \in \mathcal{S}$ as follows:

$$V^{(\ell)}(s) = \begin{cases} \frac{\sum_{i: s_i \in c_{j(s)}} \hat{V}^{(\ell)}(s_i)}{|\{i: s_i \in c_{j(s)}\}|}, & \text{if } |\{i : s_i \in c_{j(s)}\}| \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

## 4.3 Finite-Sample Analysis

For finite-sample analysis of the proposed reinforcement learning policy, we make the following structural assumption about the MDP. Specifically, we assume that the optimal value function (i.e., true regression function) is smooth in some sense. We note that some form of smoothness assumption for MDPs with continuous state/action space is typical for $\ell_\infty$ guarantee. The Lipschitz continuous assumption stated below is natural and representative in the literature on MDPs with continuous state spaces, cf. [6, 12, 13, 25, 45].

ASSUMPTION 2 (SMOOTHNESS). *The optimal value function $V^* : \mathcal{S} \to \mathbb{R}$ satisfies Lipschitz continuity with parameter $C$, i.e., $\forall s, s' \in \mathcal{S} = [0,1]^d$, $|V^*(s) - V^*(s')| \leq C\|s - s'\|_2$.*

Now we state the result characterizing the performance of the reinforcement learning policy described above. The proof is provided in Appendix D.

THEOREM 2. *Let Assumptions 1 and 2 hold. Let $\varepsilon > 0$ be a given error tolerance. Then, for $L = \Theta\Big(\log \frac{\varepsilon}{V_{\max}}\Big)$, with appropriately chosen $m_\ell, \delta_\ell$ for $\ell \in [L]$ as well as parameters in MCTS, the reinforcement learning algorithm produces estimation of value function $V^{(L)}$ such that*

$$\mathbb{E}\Big[\sup_{s \in \mathcal{S}} |V^{(L)}(s) - V^*(s)|\Big] \leq \varepsilon,$$

*by selecting $m_\ell$ states uniformly at random in $\mathcal{S}$ within iteration $\ell$. This, in total, requires $T$ number of state transitions (or samples), where*

$$T = O\Big(\varepsilon^{-\left(4+d\right)} \cdot \big(\log \frac{1}{\varepsilon}\big)^5\Big).$$

## 4.4 Minimax Lower Bound

Leveraging the the minimax lower bound for the problem of non-parametric regression [34, 41], recent work [30] establishes a lower bound on the sample complexity for reinforcement learning algorithms for general MDPs. Indeed the lower bound also holds for MDPs with deterministic transitions (the proof is provided in Appendix A), which is stated in the following proposition. We remark that the proof reduces a non-parametric regression problem to the problem of estimating the optimal value function of an MDP. Therefore, without further structural assumptions on the MDPs beyond the Lipshitz continuity, the resulting error rate $T^{-1/(2+d)}$ is known to be statistically minimax optimal. In this work, we focus on general deterministic MDPs. For subsets of MDPs with additional structures, the rate might be improved.

PROPOSITION 1. *Given an algorithm, let $V_T$ be the estimation of $V^*$ after $T$ samples of state transitions for the given MDP. Then, for each $\varepsilon \in (0,1)$, there exists an instance of deterministic MDP such that in order to achieve $\mathbb{P}\big[\big\|\hat{V}_T - V^*\big\|_\infty < \varepsilon\big] \geq \frac{1}{2}$, it must be that*

$$T \geq C' d \cdot \varepsilon^{-(d+2)} \cdot \log(\varepsilon^{-1}),$$

*where $C' > 0$ is a constant independent of the algorithm.*

Proposition 1 states that for any policy to learn the optimal value function within $\varepsilon$ approximation error, the number of samples required must scale as $\widetilde{\Omega}\big(\varepsilon^{-(2+d)}\big)$. Theorem 2 implies that the sample complexity of the proposed algorithm scales as $\widetilde{O}\big(\varepsilon^{-(4+d)}\big)$ (omitting the logarithmic factor). Hence, in terms of the dependence on the dimension, the proposed algorithm is nearly optimal. Optimizing the dependence of the sample complexity on other parameters is an important direction for future work.

# 5 NON-STATIONARY MULTI-ARM BANDIT

We introduce a class of non-stationary multi-arm bandit (MAB) problems, which will play a crucial role in analyzing the MCTS algorithm. To that end, let there be $K \geq 1$ arms or actions of interest. Let $X_{i,t}$ denote the random reward obtained by playing the arm $i \in [K]$ for the $t$th time with $t \geq 1$. Let $\bar{X}_{i,n} = \frac{1}{n}\sum_{t=1}^{n} X_{i,t}$ denote the empirical average of playing arm $i$ for $n$ times, and let $\mu_{i,n} = \mathbb{E}[\bar{X}_{i,n}]$ be its expectation. For each arm $i \in [K]$, the reward $X_{i,t}$ is bounded in $[-R, R]$ for some $R > 0$, and we assume that the reward sequence, $\{X_{i,t} : t \geq 1\}$, is a non-stationary process satisfying the following convergence and concentration properties:

A. (Convergence) the expectation $\mu_{i,n}$ converges to a value $\mu_i$, i.e.,

$$\mu_i = \lim_{n \to \infty} \mathbb{E}[\bar{X}_{i,n}]. \tag{11}$$

B. (Concentration) there exist three constants, $\beta > 1$, $\xi > 0$, and $1/2 \leq \eta < 1$ such that for every $z \geq 1$ and every integer $n \geq 1$,

$$\mathbb{P}\big(n\bar{X}_{i,n} - n\mu_i \geq n^\eta z\big) \leq \frac{\beta}{z^\xi},\ \mathbb{P}\big(n\bar{X}_{i,n} - n\mu_i \leq -n^\eta z\big) \leq \frac{\beta}{z^\xi}. \tag{12}$$

## 5.1 Algorithm

Consider applying the following variant of Upper Confidence Bound (UCB) algorithm to the above non-stationary MAB. Define upper confidence bound for arm or action $i$ when it is played $s$ times in total of $t \geq s$ time steps as

$$U_{i,s,t} = \bar{X}_{i,s} + B_{t,s}, \tag{13}$$

where $B_{t,s}$ is the "bonus term". Denote by $I_t$ the arm played at time $t \geq 1$. Then,

$$I_t \in \arg\max_{i\in[K]} \left\{\bar{X}_{i,T_i(t-1)} + B_{t-1,T_i(t-1)}\right\}, \tag{14}$$

where $T_i(t) = \sum_{l=1}^{t} \mathbb{I}\{I_l = i\}$ is the number of times arm $i$ has been played, up to (including) time $t$. We shall make specific selection of the bonus or bias term $B_{t,s}$ as

$$B_{t,s} = \frac{\beta^{1/\xi} \cdot t^{\alpha/\xi}}{s^{1-\eta}}. \tag{15}$$

A tie is broken arbitrarily when selecting an arm. In the above, $\alpha > 0$ is a tuning parameter that controls the exploration and exploitation trade-off. Let $\mu_* = \max_{i\in[K]} \mu_i$ be the optimal value with respect to the converged expectation, and $i_* \in \arg\max_{i\in[K]} \mu_i$ be the corresponding optimal arm. We assume that the optimal arm is unique. Let $\delta_{i*,n} = \mu_{i_*,n} - \mu_{i_*}$, which measures how fast the mean of the optimal non-stationary arm converges. For simplicity, quantities related to the optimal arm $i_*$ will be simply denoted with subscript $*$, e.g., $\delta_{*,n} = \delta_{i_*,n}$. Finally, denote by $\Delta_{\min} = \min_{i\in[K], i\neq i_*} \Delta_i$ the gap between the optimal arm and the second optimal arm with notation $\Delta_i = \mu_* - \mu_i$.

## 5.2 Analysis

Let $\bar{X}_n \triangleq \frac{1}{n}\sum_{i=1}^{K} T_i(n)\bar{X}_{i,T_i(n)}$ denote the empirical average of rewards collected under the UCB algorithm (14). Then, $\bar{X}_n$ satisfies the following convergence and concentration properties.

THEOREM 3. *Consider a non-stationary MAB satisfying (11) and (12). Suppose that algorithm (14) is applied with parameter $\alpha$ such that $\xi\eta(1-\eta) \leq \alpha < \xi(1-\eta)$ and $\alpha > 2$. Then, the following holds:*

*A. Convergence:*

$$\left|\mathbb{E}[\bar{X}_n] - \mu_*\right| \leq |\delta_{*,n}| + \frac{2R(K-1)\cdot\left(\left(\frac{2}{\Delta_{\min}}\cdot\beta^{\frac{1}{\xi}}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}} + \frac{2}{\alpha-2} + 1\right)}{n}.$$

*B. Concentration: there exist constants, $\beta' > 1$ and $\xi' > 0$ and $1/2 \leq \eta' < 1$ such that for every $n \geq 1$ and every $z \geq 1$,*

$$\mathbb{P}\left(n\bar{X}_n - n\mu_* \geq n^{\eta'} z\right) \leq \frac{\beta'}{z^{\xi'}},$$

$$\mathbb{P}\left(n\bar{X}_n - n\mu_* \leq -n^{\eta'} z\right) \leq \frac{\beta'}{z^{\xi'}},$$

*where $\eta' = \frac{\alpha}{\xi(1-\eta)}$, $\xi' = \alpha - 1$, $\beta'$ depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$.*

# 6 PROOF OF THEOREM 3

We establish the convergence and concentration properties of the variant of the Upper Confidence Bound algorithm described in Section 5 and specified through (13), (14) and (15).

**Establishing the Convergence Property.** We define a useful notation

$$\Phi(n,\delta) = n^{\eta}\left(\frac{\beta}{\delta}\right)^{1/\xi}. \tag{16}$$

We begin with a useful lemma, which shows that the probability that a non-optimal arm or action has a large upper confidence is polynomially small. Proof is provided in Appendix B.1.

LEMMA 1. *Let $i \in [K], i \neq i_*$ be a sub-optimal arm and define*

$$A_i(t) \triangleq \min_{u\in\mathbb{N}}\left\{\frac{\Phi(u,t^{-\alpha})}{u} \leq \frac{\Delta_i}{2}\right\} = \left\lceil\left(\frac{2}{\Delta_i}\cdot\beta^{1/\xi}\cdot t^{\alpha/\xi}\right)^{\frac{1}{1-\eta}}\right\rceil. \tag{17}$$

*For each $s$ and $t$ such that, $A_i(t) \leq s \leq t$, we have*

$$\mathbb{P}(U_{i,s,t} > \mu_*) \leq t^{-\alpha}.$$

Lemma 1 implies that as long as each arm is played enough, the sub-optimal ones become less likely to be selected. This allows us to upper bound the expected number of sub-optimal plays as follows.

LEMMA 2. *Let $i \in [K], i \neq i_*$, then*

$$\mathbb{E}[T_i(n)] \leq \left(\frac{2}{\Delta_i}\cdot\beta^{\frac{1}{\xi}}\right)^{\frac{1}{1-\eta}}\cdot n^{\frac{\alpha}{\xi(1-\eta)}} + \frac{2}{\alpha-2} + 1.$$

Proof of Lemma 2 is deferred to Appendix B.2.

*Completing Proof of Convergence.* By the triangle inequality,

$$\left|\mu_* - \mathbb{E}[\bar{X}_n]\right| = \left|\mu_* - \mu_{*,n}\right| + \left|\mu_{*,n} - \mathbb{E}[\bar{X}_n]\right| = \left|\delta_{*,n}\right| + \left|\mu_{*,n} - \mathbb{E}[\bar{X}_n]\right|.$$

The second term can be bounded as follows:

$$\begin{aligned}
&n\left|\mu_{*,n} - \mathbb{E}[\bar{X}_n]\right| \\
&= \left|\mathbb{E}\Big[\sum_{t=1}^{n} X_{i_*,t}\Big] - \mathbb{E}\Big[\sum_{i=1}^{K} T_i(n)\bar{X}_{i,T_i(n)}\Big]\right| \\
&\leq \left|\mathbb{E}\Big[\sum_{t=1}^{n} X_{i_*,t}\Big] - \mathbb{E}\Big[T_*(n)\bar{X}_{i_*,T_*(n)}\Big]\right| + \left|\mathbb{E}\Big[\sum_{i=1,i\neq i_*}^{K} T_i(n)\bar{X}_{i,T_i(n)}\Big]\right| \\
&= \left|\mathbb{E}\Big[\sum_{t=T_*(n)+1}^{n} X_{i_*,t}\Big]\right| + \left|\mathbb{E}\Big[\sum_{i=1,i\neq i_*}^{K} T_i(n)\bar{X}_{i,T_i(n)}\Big]\right|.
\end{aligned} \tag{18}$$

Recall that the reward sequences are assumed to be bounded in $[-R, R]$. Therefore, the first term of (18) can be bounded as follows:

$$\left|\mathbb{E}\Big[\sum_{t=T_*(n)+1}^{n} X_{i_*,t}\Big]\right| \leq \mathbb{E}\Big[\sum_{t=T_*(n)+1}^{n} |X_{i_*,t}|\Big] \leq R\cdot\mathbb{E}\Big[\sum_{i=1,i\neq i_*}^{K} T_i(n)\Big].$$

The second term can also be bounded as:

$$\left|\mathbb{E}\Big[\sum_{i=1,i\neq i_*}^{K} T_i(n)\bar{X}_{i,T_i(n)}\Big]\right| \leq R\cdot\mathbb{E}\Big[\sum_{i=1,i\neq i_*}^{K} T_i(n)\Big].$$

Hence, we obtain that

$$\left|\mu_* - \mathbb{E}[\bar{X}_n]\right| = \left|\delta_{*,n}\right| + \left|\mu_{*,n} - \mathbb{E}[\bar{X}_n]\right| \leq \left|\delta_{*,n}\right| + \frac{2R\cdot\mathbb{E}\left[\sum_{i=1,i\neq i_*}^{K} T_i(n)\right]}{n}.$$

Combining the above bounds and Lemma 2 yields the desired convergence result in Theorem 3.

**Establishing the Concentration Property.** Having proved the convergence property, the next step is to show that a similar concentration property (cf. (12)) also holds for $\bar{X}_n$. We aim to precisely capture the relationship between the original constants assumed in

the assumption and the new constants obtained for $\bar{X}_n$. To begin with, recall the definition of $A_i(t)$ in Lemma 1 and define

$$A(t) = \max_{i\in[K]} A_i(t) = \left\lceil \left(\frac{2}{\Delta_{\min}} \cdot \beta^{1/\xi}\right)^{\frac{1}{1-\eta}} \cdot t^{\frac{\alpha}{\xi(1-\eta)}} \right\rceil. \tag{19}$$

It can be checked that replacing $\beta$ with any larger number still makes the concentration inequalities (12) hold. Without loss of generality, we hence let $\beta$ be large enough so that $\frac{2}{\Delta_{\min}} \cdot \beta^{1/\xi} > 1$. We further denote by $N_p$ the first time such that $t \geq A(t)$, i.e.,

$$N_p = \min\{t \geq 1 \,:\, t \geq A(t)\} \;=\; \Theta\Big(\big(\frac{2^\xi \beta}{\Delta_{\min}^\xi}\big)^{\frac{1}{\xi(1-\eta)-\alpha}}\Big). \tag{20}$$

We first state the following concentration property, which will be further refined to match the desired form in Theorem 3. We defer the proof to Appendix B.3.

LEMMA 3. *For any $n \geq N_p$ and $x \geq 1$, let $r_0 = n^\eta + 2R(K-1)\big(3 + A(n)\big)$. Then,*

$$\mathbb{P}\Big(n\bar{X}_n - n\mu_* \geq r_0 x\Big) \leq \frac{\beta}{x^\xi} + \frac{2(K-1)}{(\alpha-1)\big((1+A(n))x\big)^{\alpha-1}},$$
$$\mathbb{P}\Big(n\bar{X}_n - n\mu_* \leq -r_0 x\Big) \leq \frac{\beta}{x^\xi} + \frac{2(K-1)}{(\alpha-1)\big((1+A(n))x\big)^{\alpha-1}}.$$

Lemma 3 confirms that indeed, as $n$ becomes large, the average $\bar{X}_n$ also satisfies certain concentration inequalities. However, the particular form of concentration in Theorem 3 does not quite match the form of concentration in Theorem 3 which we conclude next.

*Completing Proof of Concentration Property.* Let $N_p'$ be a constant defined as follows:

$$N_p' = \min\big\{t \geq 1 : t \geq A(t) \text{ and } 2RA(t) \geq t^\eta + 2R(4K-3)\big\}.$$

Recall the definition of $A(t)$ and that $\alpha \geq \xi\eta(1-\eta)$ and $\alpha < \xi(1-\eta)$. Hence, $N_p'$ is guaranteed to exist. In addition, note that by definition, $N_p' \geq N_p$. For each $n \geq N_p'$,

$$\begin{aligned}
&2RK\Big(\frac{2}{\Delta_{\min}} \cdot \beta^{1/\xi}\Big)^{\frac{1}{1-\eta}} \cdot n^{\frac{\alpha}{\xi(1-\eta)}} \\
&= 2RK\Big[\Big(\frac{2}{\Delta_{\min}} \cdot \beta^{1/\xi}\Big)^{\frac{1}{1-\eta}} \cdot n^{\frac{\alpha}{\xi(1-\eta)}} + 1 - 1\Big] \\
&\geq 2RKA(n) - 2RK \\
&= 2R(K-1)A(n) + 2RA(n) - 2RK \\
&\geq 2R(K-1)A(n) + n^\eta + 2R(4K-3) - 2RK \\
&= 2R(K-1)(A(n)+3) + n^\eta = r_0
\end{aligned}$$

Now, let us apply Lemma 3: for every $n \geq N_p'$ and $x \geq 1$, we have

$$\begin{aligned}
&\mathbb{P}\Big(n\bar{X}_n - n\mu_* \geq n^{\frac{\alpha}{\xi(1-\eta)}}\Big[2RK\Big(\frac{2}{\Delta_{\min}} \cdot \beta^{1/\xi}\Big)^{\frac{1}{1-\eta}}\Big]x\Big) \\
&\leq \mathbb{P}\Big(n\bar{X}_n - n\mu_* \geq r_0 x\Big) \\
&\leq \frac{\beta}{x^\xi} + \frac{2(K-1)}{(\alpha-1)\big((1+A(n))x\big)^{\alpha-1}} \\
&\leq \frac{2\max(\beta, \frac{2(K-1)}{(\alpha-1)(1+A(N_p'))^{\alpha-1}})}{x^{\alpha-1}},
\end{aligned} \tag{21}$$

where the last inequality follows because $n \geq N_p'$ and $A(n)$ is a non-decreasing function. In addition, since $\alpha < \xi(1-\eta) < \xi$, we have $\alpha - 1 < \xi$. For convenience, we define a constant

$$c_1 \triangleq 2RK\Big(\frac{2}{\Delta_{\min}} \cdot \beta^{1/\xi}\Big)^{\frac{1}{1-\eta}}. \tag{22}$$

Equivalently, by a change of variable, i.e., letting $z = c_1 x$, then for every $n \geq N_p'$ and $z \geq 1$, we obtain that

$$\mathbb{P}\Big(n\bar{X}_n - n\mu_* \geq n^{\frac{\alpha}{\xi(1-\eta)}} z\Big) \leq \frac{2c_1^{\alpha-1} \cdot \max\big(\beta, \frac{2(K-1)}{(\alpha-1)(1+A(N_p'))^{\alpha-1}}\big)}{z^{\alpha-1}}. \tag{23}$$

The above inequality holds because: (1) if $z \geq c_1$, then (23) directly follows from (21); (2) if $1 \leq z \leq c_1$, then the R.H.S. of (23) is at least 1 (by assumption, $\beta > 1$) and the inequality trivially holds. The concentration inequality, i.e., Eq. (23), is now almost the same as the desired form in Theorem 3. The only difference is that it only holds for $n \geq N_p'$. This is not hard to resolve. The easiest approach, which we show in the following, is to refine the constants to ensure that when $1 \leq n < N_p'$, Eq. (23) is trivially true. To this end, we note that $|n\bar{X}_n - n\mu_*| \leq 2Rn$. For each $1 \leq n < N_p'$, there is a corresponding $\bar{z}(n)$ such that $n^{\frac{\alpha}{\xi(1-\eta)}}\bar{z}(n) = 2Rn$. That is,

$$\bar{z}(n) \triangleq 2Rn^{1-\frac{\alpha}{\xi(1-\eta)}}, \quad 1 \leq n < N_p'.$$

This implies that for each $1 \leq n < N_p'$, the following inequality trivially holds:

$$\mathbb{P}\Big(n\bar{X}_n - n\mu_* \geq n^{\frac{\alpha}{\xi(1-\eta)}} z\Big) \leq \frac{\bar{z}(n)^{\alpha-1}}{z^{\alpha-1}}, \quad \forall\, z \geq 1.$$

To see why, note that for each $1 \leq n < N_p'$: (1) if $z \geq \bar{z}(n)$, then $n^{\frac{\alpha}{\xi(1-\eta)}} z \geq 2Rn$ and the above probability should be 0. Hence, any positive number on the R.H.S. makes the inequality trivially true; (2) if $1 \leq z < \bar{z}(n)$, the R.H.S. is at least 1, which again makes the inequality hold. For convenience, define

$$c_2 \triangleq \max_{1\leq n<N_p'} \bar{z}(n) = 2R(N_p'-1)^{1-\frac{\alpha}{\xi(1-\eta)}}. \tag{24}$$

Then, it is easy to see that for every $n \geq 1$ and every $z \geq 1$, we have

$$\mathbb{P}\big(n\bar{X}_n - n\mu_* \geq n^{\eta'} z\big) \leq \frac{\beta'}{z^{\xi'}},$$

where the constants are given by

$$\eta' = \frac{\alpha}{\xi(1-\eta)}, \tag{25}$$
$$\xi' = \alpha - 1, \tag{26}$$
$$\beta' = \max\Big\{c_2,\; 2c_1^{\alpha-1} \cdot \max\big(\beta, \frac{2(K-1)}{(\alpha-1)(1+A(N_p'))^{\alpha-1}}\big)\Big\}. \tag{27}$$

Finally, notice that since $\alpha \geq \xi\eta(1-\eta)$ and $\alpha < \xi(1-\eta)$, we have $1/2 \leq \eta \leq \eta' < 1$. Note that per (22), $c_1$ depends on $R, K, \Delta_{\min}, \beta, \xi$ and $\eta$. In addition, $c_2$ depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$ and $N_p'$ depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$. Therefore, $\beta'$ depends on $R, K, \Delta_{\min}, \beta, \xi, \alpha, \eta$. The other direction follows exactly the same reasoning, and this completes the proof of Theorem 3.

# 7 ANALYSIS OF MCTS AND PROOF OF THEOREM 1

In this section, we give a complete analysis for the fixed-depth Monte Carlo Tree Search (MCTS) algorithm illustrated in Algorithm 1 and prove Theorem 1. In effect, as discussed in Section 3, one can view a depth-$H$ MCTS as a simulated version of $H$ steps value function iterations. Given the current value function proxy $\hat{V}$, let $V^{(H)}(\cdot)$ be the value function estimation after $H$ steps of value function iteration starting with the proxy $\hat{V}$. Then, we prove the result in two parts. First, we argue that due to the MCTS sampling process, the mean of the empirical estimation of value function at the query node $s$, or the root node of MCTS tree, is within $O(n^{\eta-1})$ of $V^{(H)}(s)$ after $n$ simulations, with the given proxy $\hat{V}$ being the input to the MCTS algorithm. Second, we argue that $V^{(H)}(s)$ is within $\gamma^H\|\hat{V}-V^*\|_\infty \le \gamma^H\varepsilon_0$ of the optimal value function. Putting this together leads to Theorem 1.

We start by a preliminary probabilistic lemma in Section 7.1 that will be useful throughout. Sections 7.2 and 7.3 argue the first part of the proof as explained above. Section 7.4 provides proof of the second part. And Section 7.5 concludes the proof of Theorem 1.

## 7.1 Preliminary

We state the following probabilistic lemma that is useful throughout. Proof can be found in Appendix C.1.

LEMMA 4. *Consider real-valued random variables $X_i, Y_i$ for $i \ge 1$ such that $X$s are independent and identically distributed taking values in $[-B, B]$ for some $B > 0$, $X$s are independent of $Y$s, and $Y$s satisfy*

*A. Convergence: for $n \ge 1$, with notation $\bar{Y}_n = \frac{1}{n}\big(\sum_{i=1}^n Y_i\big)$,*

$$\lim_{n\to\infty} \mathbb{E}[\bar{Y}_n] = \mu_Y.$$

*B. Concentration: there exist constants, $\beta > 1$, $\xi > 0$, $1/2 \le \eta < 1$ such that for $n \ge 1$ and $z \ge 1$,*

$$\mathbb{P}\big(n\bar{Y}_n - n\mu_Y \ge n^\eta z\big) \le \frac{\beta}{z^\xi}, \quad \mathbb{P}\big(n\bar{Y}_n - n\mu_Y \le -n^\eta z\big) \le \frac{\beta}{z^\xi}.$$

*Let $Z_i = X_i + \rho Y_i$ for some $\rho > 0$. Then, $Z$s satisfy*

*A. Convergence: for $n \ge 1$, with notation $\bar{Z}_n = \frac{1}{n}\big(\sum_{i=1}^n Z_i\big)$, and $\mu_X = \mathbb{E}[X_1]$,*

$$\lim_{n\to\infty} \mathbb{E}[\bar{Z}_n] = \mu_X + \rho\mu_Y.$$

*B. Concentration: there exists a constant $\beta' > 1$ depending upon $\rho, \xi, \beta$ and $B$, such that for $n \ge 1$ and $z \ge 1$,*

$$\mathbb{P}\big(n\bar{Z}_n - n(\mu_X + \rho\mu_Y) \ge n^\eta z\big) \le \frac{\beta'}{z^\xi},$$

$$\mathbb{P}\big(n\bar{Z}_n - n(\mu_X + \rho\mu_Y) \le -n^\eta z\big) \le \frac{\beta'}{z^\xi}.$$

## 7.2 Analyzing Leaf Level $H$

The goal is to understand the empirical reward observed at the query node for MCTS or the root node of the MCTS tree. In particular, we argue that the mean of the empirical reward at the root node is within $O(n^{\eta-1})$ of the mean reward obtained at it assuming access to infinitely many samples. We start by analyzing the reward collected at the nodes that are at leaf level $H$ and level $H-1$.

The nodes at leaf level, i.e., level $H$, are children of nodes at level $H-1$ in the MCTS tree. Suppose there are $n_{H-1}$ nodes at level $H-1$ corresponding to states $s_{1,H-1}, \ldots, s_{n_{H-1},H-1} \in \mathcal{S}$. Consider node $i \in [n_{H-1}]$ at level $H-1$, corresponding to state $s_{i,H-1}$. As part of the algorithm, whenever this node is visited, one of the $K$ feasible actions is taken. When an action $a \in [K]$ is taken, the node $s'_H = s_{i,H-1} \circ a$, at the leaf level $H$ is reached. This results in reward at node $s_{i,H-1}$ (at level $H-1$) being equal to $\mathcal{R}(s_{i,H-1}, a) + \gamma\tilde{v}^{(H)}(s'_H)$. Here, for each $s \in \mathcal{S}$ and $a \in [K]$, the reward $\mathcal{R}(s,a)$ is an independent, bounded random variable taking value in $[-R_{\max}, R_{\max}]$ with distribution dependent on $s, a$; $\tilde{v}^{(H)}(\cdot)$ is the input of value function proxy to the MCTS algorithm denoted as $\hat{V}(\cdot)$, and $\gamma \in [0,1)$ is the discount factor. Recall that $\varepsilon_0 = \|\hat{V} - V^*\|_\infty$ and $\|V^*\|_\infty \le V_{\max}$. Therefore, $\|\tilde{v}^{(H)}\|_\infty = \|\hat{V}\|_\infty \le V_{\max} + \varepsilon_0$, and the reward collected at node $s_{i,H-1}$ by following any action is bounded, in absolute value, by $\tilde{R}^{(H-1)}_{\max} = R_{\max} + \gamma(V_{\max} + \varepsilon_0)$.

As part of the MCTS algorithm as described in (4), when node $s_{i,H-1}$ is visited for the $t+1$ time with $t \ge 0$, the action taken is

$$\arg\max_{a\in\mathcal{A}}\Bigg\{\frac{1}{u_a}\sum_{j=1}^{u_a}\Big(r(s_{i,H-1},a)(j) + \gamma\tilde{v}^{(H)}(s_{i,H-1}\circ a)(j)\Big) + \frac{\big(\beta^{(H)}\big)^{1/\xi^{(H)}}\cdot\big(t\big)^{\alpha^{(H)}/\xi^{(H)}}}{\big(u_a\big)^{1-\eta^{(H)}}}\Bigg\},$$

where $u_a \le t$ is the number of times action $a$ has been chosen thus far at state $s_{i,H-1}$ in the $t$ visits so far, $r(s_{i,H-1},a)(j)$ is the $j$th sample of random variable per distribution $\mathcal{R}(s_{i,H-1},a)$, and $\tilde{v}^{(H)}(s_{i,H-1}\circ a)(j)$ is the reward evaluated at leaf node $s_{i,H-1}\circ a$ for the $j$th time. Note that for all $j$, the reward evaluated at leaf node $s_{i,H-1}\circ a$ is the same and equals to $\tilde{v}^{(H)}(\cdot)$, the input value function proxy for the algorithm. When $u_a = 0$, we use notation $\infty$ to represent quantity inside the arg max. The net discounted reward collected by node $s_{i,H-1}$ during its total of $t \ge 1$ visits is simply the sum of rewards obtained by selecting the actions per the policy – which includes the reward associated with taking an action and the evaluation of $\tilde{v}^{(H)}(\cdot)$ for appropriate leaf node, discounted by $\gamma$. In effect, at each node $s_{i,H-1}$, we are using the UCB policy described in Section 5 with parameters $\alpha^{(H)}, \beta^{(H)}, \xi^{(H)}, \eta^{(H)}$ with $K$ possible actions, where the rewards collected by playing any of these $K$ actions each time is simply the summation of bounded independent and identical (for a given action) random variable and a deterministic evaluation. By applying Lemma 4, where $X$s correspond to independent rewards, $\rho = \gamma$, and $Y$s correspond to deterministic evaluations of $\tilde{v}^{(H)}(\cdot)$, we obtain that for given $\xi^{(H)} > 0$ and $\eta^{(H)} \in [\frac{1}{2}, 1)$, there exists $\beta^{(H)}$ such that the collected rewards at $s_{i,H-1}$ (i.e., sum of i.i.d. reward and deterministic evaluations) satisfy the convergence property cf. (11) and concentration property cf. (12) stated in Section 5. Therefore, by an application of Theorem 3, we conclude Lemma 5 stated below.

We define some notations first:

$$\mu_a^{(H-1)}(s_{i,H-1}) = \mathbb{E}[\mathcal{R}(s_{i,H-1}, a)] + \gamma \tilde{v}^{(H)}(s_{i,H-1} \circ a),$$
$$\mu_*^{(H-1)}(s_{i,H-1}) = \max_{a\in[K]} \mu_a^{(H-1)}(s_{i,H-1})$$
$$a_*^{(H-1)}(s_{i,H-1}) \in \arg\max_{a\in[K]} \mu_a^{(H-1)}(s_{i,H-1}) \tag{28}$$
$$\Delta_{\min}^{(H-1)}(s_{i,H-1}) = \mu_*^{(H-1)}(s_{i,H-1}) - \max_{a\neq a_*^{(H-1)}(s_{i,H-1})} \mu_a^{(H-1)}(s_{i,H-1}).$$

We shall assume that the maximizer in the set $\arg\max_{a\in[K]} \mu_a^{(H-1)}(s_{i,H-1})$ is unique, i.e. $\Delta_{\min}^{(H-1)}(s_{i,H-1}) > 0$. And further note that all rewards belong to $[-\tilde{R}_{\max}^{(H-1)}, \tilde{R}_{\max}^{(H-1)}]$.

LEMMA 5. *Consider a node corresponding to state $s_{i,H-1}$ at level $H-1$ within the MCTS for $i \in [n_{H-1}]$. Let $\tilde{v}^{(H-1)}(s_{i,H-1})_n$ be the total discounted reward collected at $s_{i,H-1}$ during $n \geq 1$ visits of it, to one of its $K$ leaf nodes under the UCB policy. Then, for the choice of appropriately large $\beta^{(H)} > 0$, for a given $\xi^{(H)} > 0$, $\eta^{(H)} \in [\frac{1}{2}, 1)$ and $\alpha^{(H)} > 2$, we have*

*A. Convergence:*

$$\left|\mathbb{E}\left[\frac{1}{n}\tilde{v}^{(H-1)}(s_{i,H-1})_n\right] - \mu_*^{(H-1)}(s_{i,H-1})\right|$$
$$\leq \frac{2\tilde{R}_{\max}^{(H-1)}(K-1)\cdot\left(\left(\frac{2(\beta^{(H)})^{\frac{1}{\xi^{(H)}}}}{\Delta_{\min}^{(H-1)}(s_{i,H-1})}\right)^{\frac{1}{1-\eta^{(H)}}}\cdot n^{\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}} + \frac{2}{\alpha^{(H)}-2} + 1\right)}{n}.$$

*B. Concentration: there exist constants, $\beta' > 1$ and $\xi' > 0$ and $1/2 \leq \eta' < 1$ such that for every $n \geq 1$ and every $z \geq 1$,*

$$\mathbb{P}\left(\tilde{v}^{(H-1)}(s_{i,H-1})_n - n\mu_*^{(H-1)}(s_{i,H-1}) \geq n^{\eta'}z\right) \leq \frac{\beta'}{z^{\xi'}},$$
$$\mathbb{P}\left(\tilde{v}^{(H-1)}(s_{i,H-1})_n - n\mu_*^{(H-1)}(s_{i,H-1}) \leq -n^{\eta'}z\right) \leq \frac{\beta'}{z^{\xi'}},$$

*where $\eta' = \frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}$, $\xi' = \alpha^{(H)} - 1$, and $\beta'$ is a large enough constant that is function of parameters $\alpha^{(H)}, \beta^{(H)}, \xi^{(H)}, \eta^{(H)}, \tilde{R}_{\max}^{(H-1)}, K, \Delta_{\min}^{(H-1)}(s_{i,H-1})$.*

Let $\Delta_{\min}^{(H-1)} = \min_{i\in[n_{H-1}]} \Delta_{\min}^{(H-1)}(s_{i,H-1})$. Then, the rate of convergence for each node $s_{i,H-1}$, $i \in [n_{H-1}]$ can be uniformly simplified as

$$\delta_n^{(H-1)} = \frac{2\tilde{R}_{\max}^{(H-1)}(K-1)\cdot\left(\left(\frac{2(\beta^{(H)})^{\frac{1}{\xi^{(H)}}}}{\Delta_{\min}^{(H-1)}}\right)^{\frac{1}{1-\eta^{(H)}}}\cdot n^{\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}} + \frac{2}{\alpha^{(H)}-2} + 1\right)}{n}$$
$$= \Theta\left(n^{\frac{\alpha^{(H)}}{\xi^{(H)}(1-\eta^{(H)})}-1}\right)$$
$$\overset{(a)}{=} O\left(n^{\eta-1}\right),$$

where (a) holds since $\alpha^{(H)} = \xi^{(H)}(1-\eta^{(H)})\eta^{(H)}$, $\eta^{(H)} = \eta$. It is worth remarking that $\mu_*^{(H-1)}(s_{i,H-1})$, as defined in (28), is precisely the value function estimation for $s_{i,H-1}$ at the end of one step of value iteration starting with $\hat{V}$.

## 7.3 Recursion: Going From Level $h$ to $h-1$.

Lemma 5 suggests that the necessary assumption of Theorem 3, i.e. (11) and (12), are satisfied by $\tilde{v}_n^{(H-1)}$ for each node or state at level $H-1$, with $\alpha^{(H-1)}, \xi^{(H-1)}, \eta^{(H-1)}$ as defined per relationship (5) - (7) and with appropriately defined large enough constant $\beta^{(H-1)}$. We shall argue that result similar to Lemma 5, but for node at level $H-2$, continues to hold with parameters $\alpha^{(H-2)}, \xi^{(H-2)}, \eta^{(H-2)}$ as defined per relationship (5) - (7) and with appropriately defined large enough constant $\beta^{(H-2)}$. And similar argument will continue to apply going from level $h$ to $h-1$ for all $h \leq H-1$. That is, we shall assume that the necessary assumption of Theorem 3, i.e. (11) and (12), holds for $\tilde{v}^{(h)}(\cdot)$, for all nodes at level $h$ with $\alpha^{(h)}, \xi^{(h)}, \eta^{(h)}$ as defined per relationship (5) - (7) and with appropriately defined large enough constant $\beta^{(h)}$, and then argue that such holds for nodes at level $h-1$ as well. This will, using mathematical induction, allow us to prove the results for all $h \geq 1$.

To that end, consider any node at level $h-1$. Let there be $n_{h-1}$ nodes at level $h-1$ corresponding to states $s_{1,h-1}, \ldots, s_{n_{h-1},h-1} \in \mathcal{S}$. Consider a node corresponding to state $s_{i,h-1}$ at level $h-1$ within the MCTS for $i \in [n_{h-1}]$. As part of the algorithm, whenever this node is visited, one of the $K$ feasible action is taken. When an action $a \in [K]$ is taken, the node $s'_h = s_{i,h-1} \circ a$, at the level $h$ is reached. This results in reward at node $s_{i,h-1}$ at level $h-1$ being equal to $\mathcal{R}(s_{i,h-1}, a) + \gamma\tilde{v}^{(h)}(s'_h)$. As noted before, $\mathcal{R}(s,a)$ is an independent, bounded valued random variable while $\tilde{v}^{(h)}(\cdot)$ is effectively collected by following a path all the way to the leaf level. Inductively, we assume that $\tilde{v}^{(h)}(\cdot)$ satisfies the convergence and concentration property for each node or state at level $h$, with $\alpha^{(h)}, \xi^{(h)}, \eta^{(h)}$ as defined per relationship (5) - (7) and with appropriately defined large enough constant $\beta^{(h)}$. Therefore, by an application of Lemma 4, it follows that this combined reward continues to satisfy (11) and (12), with $\alpha^{(h)}, \xi^{(h)}, \eta^{(h)}$ as defined per relationship (5) - (7) and with a large enough constant which we shall denote as $\beta^{(h)}$. These constants are used by the MCTS policy. By an application of Theorem 3, we can obtain the following Lemma 6 regarding the convergence and concentration properties for the reward sequence collected at node $s_{i,h-1}$ at level $h-1$. Similar to the notation in Eq. (28), let

$$\mu_a^{(h-1)}(s_{i,h-1}) = \mathbb{E}[\mathcal{R}(s_{i,h-1}, a)] + \gamma\mu_*^{(h)}(s_{i,h-1} \circ a)$$
$$\mu_*^{(h-1)}(s_{i,h-1}) = \max_{a\in[K]} \mu_a^{(h-1)}(s_{i,h-1})$$
$$a_*^{(h-1)}(s_{i,h-1}) \in \arg\max_{a\in[K]} \mu_a^{(h-1)}(s_{i,h-1}) \tag{29}$$
$$\Delta_{\min}^{(h-1)}(s_{i,h-1}) = \mu_*^{(h-1)}(s_{i,h-1}) - \max_{a\neq a_*^{(h-1)}(s_{i,h-1})} \mu_a^{(h-1)}(s_{i,h-1}).$$

Again, we shall assume that the maximizer in the set $\arg\max_{a\in[K]} \mu_a^{(h-1)}(s_{i,h-1})$ is unique, i.e. $\Delta_{\min}^{(h-1)}(s_{i,h-1}) > 0$. Define $\tilde{R}_{\max}^{(h-1)} = R_{\max} + \gamma\tilde{R}_{\max}^{(h)}$, where $\tilde{R}^{(H)} = V_{\max} + \varepsilon_0$. Note that all rewards collected at level $h-1$ belong to $[-\tilde{R}_{\max}^{(h-1)}, \tilde{R}_{\max}^{(h-1)}]$.

LEMMA 6. *Consider a node corresponding to state $s_{i,h-1}$ at level $h-1$ within the MCTS for $i \in [n_{h-1}]$. Let $\tilde{v}^{(h-1)}(s_{i,h-1})_n$ be the total discounted reward collected at $s_{i,h-1}$ during $n \geq 1$ visits. Then, for the choice of appropriately large $\beta^{(h)} > 0$, for a given $\xi^{(h)} > 0$, $\eta^{(h)} \in [\frac{1}{2}, 1)$ and $\alpha^{(h)} > 2$, we have*

*A. Convergence:*

$$\left|\mathbb{E}\Big[\frac{1}{n}\tilde{v}^{(h-1)}(s_{i,h-1})_n\Big] - \mu_*^{(h-1)}(s_{i,h-1})\right|$$
$$\leq \frac{2\tilde{R}_{\max}^{(h-1)}(K-1)\cdot\Big(\big(\frac{2(\beta^{(h)})^{\frac{1}{\xi^{(h)}}}}{\Delta_{\min}^{(h-1)}(s_{i,h-1})}\big)^{\frac{1}{1-\eta^{(h)}}}\cdot n^{\frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}} + \frac{2}{\alpha^{(h)}-2}+1\Big)}{n}.$$

*B. Concentration: there exist constants, $\beta' > 1$ and $\xi' > 0$ and $1/2 \leq \eta' < 1$ such that for $n \geq 1$, $z \geq 1$,*

$$\mathbb{P}\big(\tilde{v}^{(h-1)}(s_{i,h-1})_n - n\mu_*^{(h-1)}(s_{i,h-1}) \geq n^{\eta'}z\big) \leq \frac{\beta'}{z^{\xi'}},$$
$$\mathbb{P}\big(\tilde{v}^{(h-1)}(s_{i,h-1})_n - n\mu_*^{(h-1)}(s_{i,h-1}) \leq -n^{\eta'}z\big) \leq \frac{\beta'}{z^{\xi'}},$$

*where $\eta' = \frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}$, $\xi' = \alpha^{(h)} - 1$, and $\beta'$ is a large enough constant that is function of parameters $\alpha^{(h)}, \beta^{(h)}, \xi^{(h)}, \eta^{(h)}, \tilde{R}_{\max}^{(h-1)}, K, \Delta_{\min}^{(h-1)}(s_{i,h-1})$.*

As before, let us define $\Delta_{\min}^{(h-1)} = \min_{i\in[n_{h-1}]}\Delta_{\min}^{(h-1)}(s_{i,h-1})$. Similarly, we can show that for every node $s_{i,h-1}$, $i \in [n_{h-1}]$, the rate of convergence in Lemma 6 can be uniformly simplified as

$$\delta_n^{(h-1)} = \frac{2\tilde{R}_{\max}^{(h-1)}(K-1)\cdot\Big(\big(\frac{2(\beta^{(h)})^{\frac{1}{\xi^{(h)}}}}{\Delta_{\min}^{(h-1)}}\big)^{\frac{1}{1-\eta^{(h)}}}\cdot n^{\frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}} + \frac{2}{\alpha^{(h)}-2}+1\Big)}{n}$$
$$= \Theta\Big(n^{\frac{\alpha^{(h)}}{\xi^{(h)}(1-\eta^{(h)})}-1}\Big) = O\big(n^{\eta-1}\big),$$

where the last equality holds as $\alpha^{(h)} = \xi^{(h)}(1-\eta^{(h)})\eta^{(h)}$ and $\eta^{(h)} = \eta$. Again, it is worth remarking, inductively, that $\mu_*^{(h-1)}(s_{i,h-1})$ is precisely the value function estimation for $s_{i,h-1}$ at the end of $H - h + 1$ steps of value iteration starting with $\hat{V}$.

*Remark (Recursive Relation among Parameters).* With the above development, we are ready to elaborate our choice of parameters in Theorem 1, defined recursively via Eqs. (5)-(7). In essence, those parameter requirements originate from our analysis of the non-stationary MAB, i.e., Theorem 3. Recall that from our previous analysis, the key to establish the MCTS guarantee is to recursively argue the convergence and the polynomial concentration properties at each level; that is, we recursively solve the non-stationary MAB problem at each level. In order to do so, we apply our result on the non-stationary MAB (Theorem 3) recursively at each level. Importantly, recall that Theorem 3 only holds when $\xi\eta(1-\eta) \leq \alpha < \xi(1-\eta)$ and $\alpha > 2$, under which it leads to the recursive conclusions $\eta' = \frac{\alpha}{\xi(1-\eta)}$ and $\xi' = \alpha - 1$. Using our notation with superscript indicating the levels, this means that apart from the parameters at the leaf level (level $H$) which could be freely chosen, we must choose parameters of other levels recursively so that the following conditions hold:

$$\alpha^{(h)} > 2, \quad \xi^{(h)}\eta^{(h)}(1-\eta^{(h)}) \leq \alpha^{(h)} < \xi^{(h)}(1-\eta^{(h)}),$$
$$\xi^{(h)} = \alpha^{(h+1)} - 1 \text{ and } \eta^{(h)} = \frac{\alpha^{(h+1)}}{\xi^{(h+1)}(1-\eta^{(h+1)})}.$$

It is not hard to see that the conditions in Theorem 1 guarantee the above. There might be other sequences of parameters satisfying the requirements, but our particular choice gives cleaner analysis as presented in this paper.

## 7.4 Error Analysis for Value Function Iteration

We now move to the second part of the proof. The value function iteration improves the estimation of optimal value function by iterating Bellman equation. In effect, the MCTS tree is "unrolling" $H$ steps of such an iteration. Precisely, let $V^{(h)}(\cdot)$ denote the value function after $h$ iterations starting with $V^{(0)} = \hat{V}$. By definition, for any $h \geq 0$ and $s \in \mathcal{S}$,

$$V^{(h+1)}(s) = \max_{a\in[K]}\Big(\mathbb{E}[\mathcal{R}(s,a)] + \gamma V^{(h)}(s\circ a)\Big). \tag{30}$$

Recall that value iteration is contractive with respect to $\|\cdot\|_\infty$ norm (cf. [7]). That is, for any $h \geq 0$,

$$\|V^{(h+1)} - V^*\|_\infty \leq \gamma\|V^{(h)} - V^*\|_\infty. \tag{31}$$

As remarked earlier, $\mu_*^{(h-1)}(s_{i,h-1})$, the mean reward collected at node $s_{i,h-1}$ for $i \in [n_{h-1}]$ for any $h \geq 1$, is precisely $V^{(H-h+1)}(s_{i,h-1})$ starting with $V^{(0)} = \hat{V}$, the input to MCTS policy. Therefore, the mean reward collected at root node $s^{(0)}$ of the MCTS tree satisfies $\mu_*^{(0)}(s^{(0)}) = V^{(H)}(s^{(0)})$. Using (31), we obtain the following Lemma.

LEMMA 7. *The mean reward collected under the MCTS policy at root note $s^{(0)}$, $\mu_*^{(0)}(s^{(0)})$, starting with input value function proxy $\hat{V}$ is such that*

$$|\mu_*^{(0)}(s^{(0)}) - V^*(s^{(0)})| \leq \gamma^H\|\hat{V} - V^*\|_\infty. \tag{32}$$

## 7.5 Completing Proof of Theorem 1

In summary, using Lemma 6, we conclude that the recursive relationship going from level $h$ to $h-1$ holds for all $h \geq 1$ with level 0 being the root. At root $s^{(0)}$, the query state that is input to the MCTS policy, we have that after $n$ total simulations of MCTS, the empirical average of the rewards over these $n$ trial, $\frac{1}{n}\tilde{v}^{(0)}(s_0)_n$ is such that (using the fact that $\alpha^{(0)} = \xi^{(0)}(1-\eta^{(0)})\eta^{(0)}$)

$$\left|\mathbb{E}\Big[\frac{1}{n}\tilde{v}^{(0)}(s_0)_n\Big] - \mu_*^{(0)}\right| = O\Big(n^{\frac{\alpha^{(0)}}{\xi^{(0)}(1-\eta^{(0)})}-1}\Big) = O\Big(n^{\eta-1}\Big), \tag{33}$$

where $\mu_*^{(0)}$ is the value function estimation for $s^{(0)}$ after $H$ iterations of value function iteration starting with $\hat{V}$. By Lemma 7, we have

$$|\mu_*^{(0)} - V^*(s^{(0)})| \leq \gamma^H\varepsilon_0, \tag{34}$$

since $\varepsilon_0 = \|\hat{V} - V^*\|_\infty$. Combining (33) and (34),

$$\left|\mathbb{E}\Big[\frac{1}{n}\tilde{v}^{(0)}(s_0)_n\Big] - V^*(s^{(0)})\right| \leq \gamma^H\varepsilon_0 + O\Big(n^{\eta-1}\Big). \tag{35}$$

This concludes the proof of Theorem 1.

# 8 CONCLUSION

In this paper, we introduce a *correction* of the popular Monte Carlo Tree Search (MCTS) policy for improved value function estimation for a given state, using an existing value function estimation for the entire state space. This correction was obtained through careful, rigorous analysis of a non-stationary Multi-Arm Bandit where rewards are dependent and non-stationary. In particular, we analyzed a variant of the classical Upper Confidence Bound policy for such an MAB. Using this as a building block, we establish rigorous performance guarantees for the *corrected* version of MCTS proposed in

this work. This, to the best of our knowledge, is the first mathematically correct analysis of the UCT policy despite its popularity since it has been proposed in literature [20, 21]. We further establish that the proposed MCTS policy, when combined with nearest neighbor supervised learning, leads to near optimal sample complexity for obtaining estimation of value function within a given tolerance, where the optimality is in the minimax sense. This suggests the tightness of our analysis as well as the utility of the MCTS policy.

We take a note that much of this work was inspired by the success of AlphaGo Zero (AGZ) which utilizes MCTS combined with supervised learning. Interestingly enough, the correction of MCTS suggested by our analysis is qualitatively similar to the version of MCTS utilized by AGZ as reported in practice. This seeming coincidence may suggest further avenue for practical utility of versions of the MCTS proposed in this work and is an interesting direction for future work.

# REFERENCES


[1] Rajeev Agrawal. 1995. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability* 27, 4 (1995), 1054–1078.

[2] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410, 19 (2009), 1876–1902.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[4] Kamyar Azizzadenesheli, Brandon Yang, Weitang Liu, Emma Brunskill, Zachary C. Lipton, and Animashree Anandkumar. 2018. Sample-Efficient Deep RL with Generative Adversarial Tree Search. *CoRR* abs/1806.05780 (2018).

[5] Peter Bartlett, Victor Gabillon, Jennifer Healey, and Michal Valko. 2019. Scale-free adaptive planning for deterministic dynamics & discounted rewards. In *International Conference on Machine Learning*. 495–504.

[6] D. Bertsekas. 1975. Convergence of discretization procedures in dynamic programming. *IEEE Trans. Automat. Control* 20, 3 (1975), 415–419.

[7] D.P. Bertsekas. 2017. *Dynamic Programming and Optimal Control*. Athena Scientific.

[8] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* 4, 1 (2012), 1–43.

[9] Hyeong Soo Chang, Michael C. Fu, Jiaqiao Hu, and Steven I Marcus. 2005. An adaptive sampling algorithm for solving Markov decision processes. *Operations Research* 53, 1 (2005), 126–139.

[10] Pierre-Arnaud Coquelin and Rémi Munos. 2007. Bandit algorithms for tree search. *arXiv preprint cs/0703062* (2007).

[11] Rémi Coulom. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*. Springer, 72–83.

[12] F. Dufour and T. Prieto-Rumeau. 2012. Approximation of Markov decision processes with general state space. *Journal of Mathematical Analysis and applications* 388, 2 (2012), 1254–1267.

[13] F. Dufour and T. Prieto-Rumeau. 2013. Finite linear programming approximations of constrained discounted Markov decision processes. *SIAM Journal on Control and Optimization* 51, 2 (2013), 1298–1324.

[14] E. Even-Dar and Y. Mansour. 2004. Learning Rates for Q-learning. *JMLR* 5 (Dec. 2004).

[15] Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard L Lewis, and Xiaoshi Wang. 2014. Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. In *Advances in neural information processing systems*. 3338–3346.

[16] W Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of Americal Statistics Association* 58 (1963).

[17] Jean-Francois Hren and Rémi Munos. 2008. Optimistic planning of deterministic systems. In *European Workshop on Reinforcement Learning*. Springer, 151–164.

[18] Daniel R. Jiang, Emmanuel Ekwedike, and Han Liu. 2018. Feedback-Based Tree Search for Reinforcement Learning. In *International conference on machine learning*.

[19] Emilie Kaufmann and Wouter M Koolen. 2017. Monte-carlo tree search by best arm identification. In *Advances in Neural Information Processing Systems*. 4897–4906.

[20] Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*. Springer, 282–293.

[21] Levente Kocsis, Csaba Szepesvári, and Jan Willemson. 2006. Improved monte-carlo search. *Univ. Tartu, Estonia, Tech. Rep* (2006).

[22] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. 1928–1937.

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[25] Rémi Munos et al. 2014. From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning* 7, 1 (2014), 1–129.

[26] Antoine Salomon and Jean-Yves Audibert. 2011. Deviations of stochastic bandit regret. In *International Conference on Algorithmic Learning Theory*. Springer, 159–173.

[27] Maarten P. D. Schadd, Mark H. M. Winands, H. Jaap van den Herik, Guillaume M. J. B. Chaslot, and Jos W. H. M. Uiterwijk. 2008. Single-Player Monte-Carlo Tree Search. In *Computers and Games*, H. Jaap van den Herik, Xinhe Xu, Zongmin Ma, and Mark H. M. Winands (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–12.

[28] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International Conference on Machine Learning*. 1889–1897.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[30] Devavrat Shah and Qiaomin Xie. 2018. Q-learning with Nearest Neighbors. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 3115–3125.

[31] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[32] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *arXiv preprint arXiv:1712.01815* (2017).

[33] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.

[34] Charles J. Stone. 1982. Optimal Global Rates of Convergence for Nonparametric Regression. *The Annals of Statistics* (1982), 1040–1053.

[35] Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. 2006. PAC model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 881–888.

[36] Nathan R. Sturtevant. 2008. An Analysis of UCT in Multi-player Games. In *Computers and Games*, H. Jaap van den Herik, Xinhe Xu, Zongmin Ma, and Mark H. M. Winands (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 37–49.

[37] Richard S. Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3, 1 (1988), 9–44.

[38] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.

[39] Csaba Szepesvári. 2019. Personal communication.

[40] Kazuki Teraoka, Kohei Hatano, and Eiji Takimoto. 2014. Efficient sampling method for monte carlo tree search problem. *IEICE TRANSACTIONS on Information and Systems* 97, 3 (2014), 392–398.

[41] Alexandre B. Tsybakov. 2009. *Introduction to Nonparametric Estimation*. Springer.

[42] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-Learning.. In *AAAI*, Vol. 2. Phoenix, AZ, 5.

[43] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.

[44] Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. 2019. Harnessing Structures for Value-Based Planning and Reinforcement Learning. *arXiv preprint arXiv:1909.12255* (2019).

[45] Zhuora Yang, Yuchen Xie, and Zhaoran Wang. 2019. A theoretical analysis of deep Q-learning. *arXiv preprint arXiv:1901.00137* (2019).

# A PROOF OF PROPOSITION 1

The recent work [30] establishes a lower bound on the sample complexity for reinforcement learning algorithms on MDPs. We follow a similar argument to establish a lower bound on the sample complexity for MDPs with deterministic transitions. We provide the proof for completeness. The key idea is to connect the problem of estimating the value function to the problem of non-parametric regression, and then leveraging known minimax lower bound for the latter. In particular, we show that a class of non-parametric regression problem can be embedded in an MDP with deterministic transitions, so any algorithm for the latter can be used to solve the former. Prior work on non-parametric regression [34, 41] establishes that a certain number of observations is *necessary* to achieve a given accuracy using *any* algorithms, hence leading to a corresponding necessary condition for the sample size of estimating the value function in an MDP problem. We now provide the details.

**Step 1. Non-parametric regression.** Consider the following non-parametric regression problem: Let $\mathcal{S} := [0,1]^d$ and assume that we have $T$ data pairs $(x_1, y_1), \ldots, (x_T, y_T)$ such that conditioned on $x_1, \ldots, x_n$, the random variables $y_1, \ldots, y_n$ are independent and satisfy

$$\mathbb{E}[y_t | x_t] = f(x_t), \qquad x_t \in \mathcal{S} \tag{36}$$

where $f : \mathcal{S} \to \mathbb{R}$ is the unknown regression function. Suppose that the conditional distribution of $y_t$ given $x_t = x$ is a Bernoulli distribution with mean $f(x)$. We also assume that $f$ is 1-Lipschitz continuous with respect to the Euclidean norm, i.e.,

$$|f(x) - f(x_0)| \le |x - x_0|, \quad \forall x, x_0 \in \mathcal{S}.$$

Let $\mathcal{F}$ be the collection of all 1-Lipschitz continuous function on $\mathcal{X}$, i.e.,

$$\mathcal{F} = \{h | h \text{ is a 1-Lipschitz function on } \mathcal{S}\},$$

The goal is to estimate $f$ given the observations $(x_1, y_1), \ldots, (x_T, y_T)$ and the prior knowledge that $f \in \mathcal{F}$.

It is easy to verify that the above problem is a special case of the non-parametric regression problem considered in the work by [34] (in particular, Example 2 therein). Let $\hat{f}_T$ denote an arbitrary (measurable) estimator of $f$ based on the training samples $(x_1, y_1), \ldots, (x_T, y_T)$. By Theorem 1 in [34], we have the following result: there exists a $c > 0$ such that

$$\lim_{T\to\infty} \inf_{\hat{f}_T} \sup_{f\in\mathcal{F}} \mathbb{P}\left( \left\| \hat{f}_T - f \right\|_\infty \ge c \left( \frac{\log T}{T} \right)^{\frac{1}{2+d}} \right) = 1, \tag{37}$$

where infimum is over all possible estimators $\hat{f}_T$. Translating this result to the non-asymptotic regime, we obtain the following theorem.

THEOREM 4. *Under the above stated assumptions, for any number $\delta \in (0,1)$, there exits $c > 0$ and $T_\delta$ such that*

$$\inf_{\hat{f}_T} \sup_{f\in\mathcal{F}} \mathbb{P}\left( \left\| \hat{f}_T - f \right\|_\infty \ge c \left( \frac{\log T}{T} \right)^{\frac{1}{2+d}} \right) \ge \delta, \qquad \textit{for all } T \ge T_\delta.$$

**Step 2. MDP with deterministic transitions.** Consider a class of discrete-time discounted MDPs $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where

$\mathcal{S} = [0,1]^d$,
$\mathcal{A}$ is finite,
$\forall\, (x, a)$, there exists a unique $x' \in \mathcal{S}$ s.t. $\mathcal{P}(x'|x, a) = 1$,
$r(x, a) = r(x)$ for all $a$,
$\gamma = 0$.

In words, the transition is deterministic, the expected reward is independent of the action taken and the current state, and only immediate reward matters.

Let $R_t$ be the observed reward at step $t$. We assume that given $x_t$, the random variable $R_t$ is independent of $(x_1, \ldots, x_{t-1})$, and follows a Bernoulli distribution $\text{Bernoulli}(r(x_t))$. The expected reward function $r(\cdot)$ is assumed to be 1-Lipschitz and bounded. It is easy to see that for all $x \in \mathcal{S}$, $a \in \mathcal{A}$,

$$V^*(x) = r(x). \tag{38}$$

**Step 3. Reduction from regression to MDP.** Given a non-parametric regression problem as described in Step 1, we may reduce it to the problem of estimating the value function $V^*$ of the MDP described in Step 2. To do this, we set

$$r(x) = f(x), \qquad \forall x \in \mathcal{S}$$

and

$$R_t = y_t, \qquad t = 1, 2, \ldots, T.$$

In this case, it follows from equations (38) that the value function is given by $V^* = f$. Moreover, the expected reward function $r(\cdot)$ is 1-Lipschitz, so the assumptions of the MDP in Step 2 are satisfied. This reduction shows that the MDP problem is at least as hard as the nonparametric regression problem, so a lower bound for the latter is also a lower bound for the former.

Applying Theorem 4 yields the following result: for any number $\delta \in (0,1)$, there exist some numbers $c > 0$ and $T_\delta > 0$, such that

$$\inf_{\hat{V}_T} \sup_{V^*\in\mathcal{F}} \mathbb{P}\left[ \left\| \hat{V}_T - V^* \right\|_\infty \ge c \left( \frac{\log T}{T} \right)^{\frac{1}{2+d}} \right] \ge \delta, \qquad \text{for all } T \ge T_\delta.$$

Consequently, for any reinforcement learning algorithm $\hat{V}_T$ and any sufficiently small $\varepsilon > 0$, there exists an MDP problem with deterministic transitions such that in order to achieve

$$\mathbb{P}\left[ \left\| \hat{V}_T - V^* \right\|_\infty < \varepsilon \right] \ge 1 - \delta,$$

one must have

$$T \ge C' d \left( \frac{1}{\varepsilon} \right)^{2+d} \log\left( \frac{1}{\varepsilon} \right),$$

where $C' > 0$ is a constant. The statement of Proposition 1 follows by selecting $\delta = \frac{1}{2}$.

# B ADDITIONAL PROOFS FOR THEOREM 3

## B.1 Proof of Lemma 1

PROOF. By the choice of $A_i(t)$, $s$ and $t$, we have $B_{t,s} = \frac{\Phi(s,t^{-\alpha})}{s} \le \frac{\Phi(A_i(t),t^{-\alpha})}{A_i(t)} \le \frac{\Delta_i}{2}$. Therefore,

$$\begin{aligned}
\mathbb{P}(U_{i,s,t} > \mu_*) &= \mathbb{P}(\bar{X}_{i,s} + B_{t,s} > \mu_*) \\
&= \mathbb{P}\left(\bar{X}_{i,s} - \mu_i > \Delta_i - B_{t,s}\right) \\
&\le \mathbb{P}\left(\bar{X}_{i,s} - \mu_i > B_{t,s}\right) && \Delta_i \ge 2B_{t,s} \\
&\le t^{-\alpha}. && \text{by concentration (12).}
\end{aligned}$$

□

## B.2 Proof of Lemma 2

PROOF. If a sub-optimal arm $i$ is chosen at time $t+1$, i.e., $I_{t+1} = i$, then at least one of the following two equations must be true: with notation $T_*(\cdot) = T_{i_*}(\cdot)$,

$$U_{i_*,T_*(t),t} \le \mu_* \,, \tag{39}$$
$$U_{i,T_i(t),t} > \mu_* \,. \tag{40}$$

Indeed, if both inequalities are false, we have $U_{i_*,T_*(t),t} > \mu_* \ge U_{i,T_i(t),t}$, which is a contradiction to $I_{t+1} = i$. We now use this fact to prove Lemma 2.

**Case 1:** $n > A_i(n)$. Note that such $n$ exists because $A_i(n)$ grows with a polynomial order $O\big(n^{\frac{\alpha}{\xi(1-\eta)}}\big)$ and $\alpha < \xi(1-\eta)$, i.e., $A_i(n) = o(n)$. Then,

$$\begin{aligned}
T_i(n) &= \sum_{t=0}^{n-1} \mathbb{I}\{I_{t+1} = i\} = 1 + \sum_{t=K}^{n-1} \mathbb{I}\{I_{t+1} = i\} \\
&\overset{(a)}{=} 1 + \sum_{t=K}^{n-1} \big(\mathbb{I}\{I_{t+1} = i, T_i(t) < A_i(n)\} + \mathbb{I}\{I_{t+1} = i, T_i(t) \ge A_i(n)\}\big) \\
&\le A_i(n) + \sum_{t=K}^{n-1} \mathbb{I}\{I_{t+1} = i, T_i(t) \ge A_i(n)\},
\end{aligned}$$

where equality (a) follows from the fact that $B_{t,s} = \infty$ if $s = 0$.

To analyze the above summation, we note that from (39) and (40),

$$\begin{aligned}
&\mathbb{I}\{I_{t+1} = i, T_i(t) \ge A_i(n)\} \\
&\le \mathbb{I}\{U_{i_*,T_*(t),t} \le \mu_* \text{ or } U_{i,T_i(t),t} > \mu_*, T_i(t) \ge A_i(n)\} \\
&\le \mathbb{I}\{U_{i,T_i(t),t} > \mu_*, T_i(t) \ge A_i(n)\} + \mathbb{I}\{U_{i_*,T_*(t),t} \le \mu_*, T_i(t) \ge A_i(n)\} \\
&\le \mathbb{I}\{U_{i,T_i(t),t} > \mu_*, T_i(t) \ge A_i(n)\} + \mathbb{I}\{U_{i_*,T_*(t),t} \le \mu_*\} \\
&= \mathbb{I}\{\exists\, s : A_i(n) \le s \le t, \text{ s.t. } U_{i,s,t} > \mu_*\} \\
&+ \mathbb{I}\{\exists\, s_* : 1 \le s_* \le t, \text{ s.t. } U_{i_*,s_*,t} \le \mu_*\}.
\end{aligned}$$

To summarize, we have proved that

$$\begin{aligned}
\mathbb{E}[T_i(n)] &\le A_i(n) + \sum_{t=A_i(n)}^{n-1} \mathbb{P}\Big((39) \text{ or } (40) \text{ is true, and } T_i(t) \ge A_i(n)\Big) \\
&\le A_i(n) + \sum_{t=A_i(n)}^{n-1} \Big[\mathbb{P}\big(\underbrace{\exists\, s : A_i(n) \le s \le t, \text{ s.t. } U_{i,s,t} > \mu_*}_{E_1}\big) \\
&+ \mathbb{P}\big(\underbrace{\exists\, s_* : 1 \le s_* \le t, \text{ s.t. } U_{i_*,s_*,t} \le \mu_*}_{E_2}\big)\Big]. \qquad (41)
\end{aligned}$$

To complete the proof of Lemma 2, it suffices to bound the probabilities of the two events $E_1$ and $E_2$. To this end, we use a union bound:

$$\mathbb{P}(E_1) \le \sum_{s=A_i(n)}^{t} \mathbb{P}(U_{i,s,t} > \mu_*) \overset{(a)}{\le} \sum_{s=A_i(n)}^{t} t^{-\alpha} \le t \cdot t^{-\alpha} = t^{1-\alpha},$$

where the step $(a)$ follows from $A_i(n) \ge A_i(t)$ and Lemma 1. We bound $\mathbb{P}(E_2)$ in a similar way:

$$\begin{aligned}
\mathbb{P}(E_2) &\le \sum_{s_*=1}^{t} \mathbb{P}(U_{i_*,s_*,t} \le \mu_*) = \sum_{s_*=1}^{t} \mathbb{P}\left(\bar{X}_{i_*,s_*} + B_{t,s_*} \le \mu_*\right) \\
&\overset{(a)}{\le} \sum_{s_*=1}^{t} t^{-\alpha} \le t^{1-\alpha},
\end{aligned}$$

where step $(a)$ follows from concentration (cf. (12)). By substituting the bounds of $\mathbb{P}(E_1)$ and $\mathbb{P}(E_2)$ into (41), we have:

$$\begin{aligned}
\mathbb{E}[T_i(n)] &\le A_i(n) + \sum_{t=A_i(n)}^{n-1} 2t^{1-\alpha} \\
&\le A_i(n) + \int_{A_i(n)-1}^{\infty} 2t^{1-\alpha} dt && \alpha > 2 \\
&= A_i(n) + \frac{2\big(A_i(n)-1\big)^{2-\alpha}}{\alpha - 2} \\
&\le A_i(n) + \frac{2}{\alpha - 2} \\
&\le \left(\frac{2}{\Delta_i} \cdot \beta^{1/\xi}\right)^{\frac{1}{1-\eta}} \cdot n^{\frac{\alpha}{\xi(1-\eta)}} + \frac{2}{\alpha - 2} + 1.
\end{aligned}$$

**Case 2:** $n \le A_i(n)$. Note that if $n$ is such that $n \le A_i(n)$, then the above bound trivially holds because $T_i(n) \le n \le A_i(n)$. This completes the proof of Lemma 2. □

## B.3 Proof of Lemma 3

PROOF. We first prove one direction, namely, $\mathbb{P}(n\mu_* - n\bar{X}_n \ge r_0 x)$. The other direction follows the similar steps, and we will comment on that at the end of this proof. The general idea underlying the proof is to rewrite the quantity $n\mu_* - n\bar{X}_n$ as sums of terms that can be bounded using previous lemmas or assumptions. To begin with, note

that

$$\begin{aligned} n\mu_* - n\bar{X}_n &= n\mu_* - \sum_{i=1}^{K} T_i(n)\bar{X}_{i,T_i(n)} \\ &= n\mu_* - \sum_{t=1}^{T_*(n)} X_{i_*,t} - \sum_{i\neq i_*} T_i(n)\bar{X}_{i,T_i(n)} \\ &= n\mu_* - \sum_{t=1}^{n} X_{i_*,t} + \sum_{t=T_*(n)+1}^{n} X_{i_*,t} - \sum_{i\neq i_*}\sum_{t=1}^{T_i(n)} X_{i,t} \\ &\leq n\mu_* - \sum_{t=1}^{n} X_{i_*,t} + 2R\sum_{i\neq i_*} T_i(n), \end{aligned}$$

because $X_{i,t} \in [-R, R]$ for all $i, t$. Therefore, we have

$$\begin{aligned} \mathbb{P}\big(n\mu_* - n\bar{X}_n \geq r_0 x\big) &\leq \mathbb{P}\Big(n\mu_* - \sum_{t=1}^{n} X_{i_*,t} + 2R\sum_{i\neq i_*} T_i(n) \geq r_0 x\Big) \\ &\leq \mathbb{P}\Big(n\mu_* - \sum_{t=1}^{n} X_{i_*,t} \geq n^{\eta} x\Big) \\ &\quad + \sum_{i\neq i_*} \mathbb{P}\big(T_i(n) \geq (3 + A(n))x\big), \end{aligned} \tag{42}$$

where the last inequality follows from the union bound.

To prove the theorem, we now bound the two terms in (42). By our concentration assumption, we can upper bound the first term as follows:

$$\mathbb{P}\Big(n\mu_* - \sum_{t=1}^{n} X_{i_*,t} \geq n^{\eta} x\Big) \leq \frac{\beta}{x^{\xi}}. \tag{43}$$

Next, we bound each term in the summation of (42). Fix $n$ and a sub-optimal edge $i$. Let $u$ be an integer satisfying $u \geq A(n)$. For any $\tau \in \mathbb{R}$, consider the following two events:

$$E_1 = \big\{\text{For each integer } t \in [u, n],\ \text{we have } U_{i,u,t} \leq \tau\big\},$$

$$E_2 = \big\{\text{For each integer } s \in [1, n-u],\ \text{we have } U_{i_*,s,u+s} > \tau\big\}.$$

As a first step, we want to show that

$$E_1 \cap E_2 \Rightarrow T_i(n) \leq u. \tag{44}$$

To this end, let us condition on both events $E_1$ and $E_2$. Recall that $B_{t,s}$ is non-decreasing with respect to $t$. Then, for each $s$ such that $1 \leq s \leq n-u$, and each $t$ such that $u+s \leq t \leq n$, it holds that

$$U_{i_*,s,t} = \bar{X}_{i_*,s} + B_{t,s} \geq \bar{X}_{i_*,s} + B_{u+s,s} = U_{i_*,s,u+s} > \tau \geq U_{i,u,t}.$$

This implies that $T_i(n) \leq u$. To see why, suppose that $T_i(n) > u$ and denote by $t'$ the first time that arm $i$ has been played $u$ times, i.e., $t' = \min\{t : t \leq n, T_i(t) = u\}$. Note that by definition $t' \geq u + T_*(t')$. Hence, for any time t such that $t' < t \leq n$, the above inequality implies that $U_{i_*,T_*(t),t} > U_{i,u,t}$. That is, $i^*$ always has a higher upper confidence bound than $i$, and arm $i$ will not be selected, i.e., arm $i$ will not be played the $(u+1)$-th time. This contradicts our assumption that $T_i(n) > u$, and hence we have the inequality $T_i(n) \leq u$.

To summarize, we have established the fact that $E_1 \cap E_2 \Rightarrow T_i(n) \leq u$. As a result, we have:

$$\begin{aligned} \{T_i(n) > u\} &\subset \big(E_1^c \cup E_2^c\big) \\ &= \big(\big\{\exists\, t :\ u \leq t \leq n \text{ s.t. } U_{i,u,t} > \tau\big\} \\ &\quad \cup \big\{\exists\, s : 1 \leq s \leq n-u,\ \text{s.t. } U_{i_*,s,u+s} \leq \tau\big\}\big). \end{aligned}$$

Using union bound, we obtain that

$$\mathbb{P}\big(T_i(n) > u\big) \leq \sum_{t=u}^{n} \mathbb{P}(U_{i,u,t} > \tau) + \sum_{s=1}^{n-u} \mathbb{P}\big(U_{i_*,s,u+s} \leq \tau\big). \tag{45}$$

Note that for the above bound, we are free to choose $u$ and $\tau$ as long as $u \geq A(n)$. To connect with our goal (cf. (42)), in the following, we set $u = \lfloor(1 + A(n))x\rfloor + 1$ (recall that $x \geq 1$) and $\tau = \mu_*$ to bound $\mathbb{P}(T_i(n) > u)$. Since $u \geq A(n) \geq A_i(n)$, by Lemma 1, we have

$$\begin{aligned} \sum_{t=u}^{n} \mathbb{P}(U_{i,u,t} > \mu_*) &\leq \sum_{t=u}^{n} t^{-\alpha} \leq \int_{u-1}^{\infty} t^{-\alpha} dt = \frac{(u-1)^{1-\alpha}}{\alpha - 1} \\ &= \frac{(\lfloor(1 + A(n))x\rfloor)^{1-\alpha}}{\alpha - 1} \leq \frac{\Big((1 + A(n))x\Big)^{1-\alpha}}{\alpha - 1}. \end{aligned}$$

As for the second summation in the R.H.S. of (45), we have that

$$\begin{aligned} &\sum_{s=1}^{n-u} \mathbb{P}\big(U_{i_*,s,u+s} \leq \tau\big) \\ &= \sum_{s=1}^{n-u} \mathbb{P}(U_{i_*,s,u+s} \leq \mu_*) \\ &= \sum_{s=1}^{n-u} \mathbb{P}\Big(\bar{X}_{i_*,s} + B_{u+s,s} \leq \mu_*\Big) \\ &\leq \sum_{s=1}^{n-u} (s+u)^{-\alpha} \\ &= \sum_{t=1+u}^{n} t^{-\alpha} \\ &\leq \int_{u-1}^{\infty} t^{-\alpha} dt = \frac{(u-1)^{1-\alpha}}{\alpha - 1} \leq \frac{\Big((1 + A(n))x\Big)^{1-\alpha}}{\alpha - 1}, \end{aligned}$$

where the first inequality follows from the concentration property, cf. (12). Combining the above inequalities and note that $(3 + A(n))x > \lfloor(1 + A(n))x\rfloor + 1$:

$$\mathbb{P}\big(T_i(n) \geq (3 + A(n))x\big) \leq \mathbb{P}\big(T_i(n) > u\big) \leq \frac{2\Big(\big(1 + A(n)\big)x\Big)^{1-\alpha}}{\alpha - 1}. \tag{46}$$

Substituting (43) and (46) into (42), we obtain

$$\mathbb{P}\big(n\mu_* - n\bar{X}_n \geq r_0 x\big) \leq \frac{\beta}{x^{\xi}} + \sum_{i\neq i_*} \frac{2\Big(\big(1 + A(n)\big)x\Big)^{1-\alpha}}{\alpha - 1},$$

which is the desired inequality in Lemma 3.

To complete the proof, we need to consider the other direction, i.e., $\mathbb{P}(n\bar{X}_n - n\mu_* \geq r_0 x)$. The proof is almost identical. Note that

$$\begin{aligned} n\bar{X}_n - n\mu_* &= \sum_{i=1}^{K} T_i(n)\bar{X}_{i,T_i(n)} - n\mu_* \\ &= \sum_{t=1}^{n} X_{i_*,t} - n\mu_* - \sum_{t=T_*(n)+1}^{n} X_{i_*,t} + \sum_{i\neq i_*}\sum_{t=1}^{T_i(n)} X_{i,t} \\ &\leq \sum_{t=1}^{n} X_{i_*,t} - n\mu_* + 2R\sum_{i\neq i_*} T_i(n), \end{aligned}$$

because $X_{i,t} \in [-R, R]$ for all $i, t$. Therefore,

$$\begin{aligned}
&\mathbb{P}\big(n\bar{X}_n - n\mu_* \geq r_0 x\big) \\
&\leq \mathbb{P}\Big(\sum_{t=1}^{n} X_{i_*,t} - n\mu_* + 2R\sum_{i\neq i_*} T_i(n) \geq r_0 x\Big) \\
&\leq \mathbb{P}\Big(\sum_{t=1}^{n} X_{i_*,t} - n\mu_* \geq n^{\eta} x\Big) + \sum_{i\neq i_*} \mathbb{P}\big(T_i(n) \geq (3 + A_i(n))x\big).
\end{aligned}$$

The desired inequality then follows exactly from the same reasoning of our previous proof. □

# C ADDITIONAL PROOFS FOR THEOREM 1

## C.1 Proof of Lemma 4

Proof. The convergence property, $\lim_{n\to\infty} \mathbb{E}[\bar{Z}_n] = \mu_X + \rho\mu_Y$, follows simply by linearity of expectation. For concentration, consider the following: since $X$s are i.i.d. bounded random variables taking value in $[-B, B]$, by Hoeffding's inequality [16], we have that for $t \geq 0$,

$$\begin{aligned}
\mathbb{P}\big(n\bar{X}_n - n\mu_X \geq nt\big) &\leq \exp\Big(-\frac{t^2 n}{2B^2}\Big), \qquad (47)\\
\mathbb{P}\big(n\bar{X}_n - n\mu_X \leq -nt\big) &\leq \exp\Big(-\frac{t^2 n}{2B^2}\Big).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\Big(n\bar{Z}_n - n(\mu_X + \rho\mu_Y) \geq n^{\eta} z\Big) \\
&\leq \mathbb{P}\Big(n\bar{X}_n - n\mu_X \geq \frac{n^{\eta} z}{2}\Big) + \mathbb{P}\Big(n\bar{Y}_n - n\mu_Y \geq \frac{n^{\eta} z}{2\rho}\Big) \\
&\leq \exp\Big(-\frac{z^2 n^{2\eta-1}}{8B^2}\Big) + \frac{\beta 2^{\xi}\rho^{\xi}}{z^{\xi}} \\
&\leq \frac{\beta'}{z^{\xi}}, \qquad (48)
\end{aligned}$$

where $\beta'$ is a large enough constant depending upon $\rho, \xi, \beta$ and $B$. The other-side of the inequality follows similarly. This completes the proof. □

# D PROOF OF THEOREM 2

First, we establish a useful property of nearest neighbor supervised learning presented in Section 4.2. This is stated in Section D.1. We will use it, along with the guarantees obtained for MCTS in Theorem 1 to establish Theorem 2 in Section D.2. Throughout, we shall assume the setup of Theorem 2.

## D.1 Guarantees for Supervised Learning

Let $\delta \in (0, 1)$ be given. As stated in Section 4.2, let $K(\delta, d) = \Theta(\delta^{-d})$ be the collection of balls of radius $\delta$, say $c_i$, $i \in [K(\delta, d)]$, so that they cover $\mathcal{S}$, i.e. $\mathcal{S} \subset \cup_{i\in[K(\varepsilon,d)]} c_i$. Also, by construction, each of these balls have intersection with $\mathcal{S}$ whose volume is at least $C_d\delta^d$. Let $S = \{s_i : i \in [N]\}$ denote $N$ state samples from $\mathcal{S}$ uniformly at random and independent of each other. For each state $s \in \mathcal{S}$, let $V : \mathcal{S} \to [-V_{\max}, V_{\max}]$ be such that $|\mathbb{E}[V(s)] - V^*(s)| \leq \Delta$. Let the nearest neighbor supervised learning described in Section 4.2 produce estimate $\hat{V} : \mathcal{S} \to \mathbb{R}$ using labeled data points $(s_i, V(s_i))_{i\in[N]}$. Then, we claim the following guarantee. Proof can be found in Section D.3.

Lemma 8. *Under the above described setup, as long as* $N \geq 32\max(1, \delta^{-2}V_{\max}^2)C_d^{-1}\delta^{-d}\log\frac{K(\delta,d)}{\delta}$, *i.e.,* $N = \Omega(d\delta^{-d-2}\log\delta^{-1})$,

$$\mathbb{E}\Big[\sup_{s\in\mathcal{S}} |\hat{V}(s) - V^*(s)|\Big] \leq \Delta + (C+1)\delta + \frac{4V_{\max}\delta^2}{K(\delta, d)}. \qquad (49)$$

## D.2 Establishing Theorem 2

Using Theorem 1 and Lemma 8, we complete the proof of Theorem 2 under appropriate choice of algorithmic parameters. We start by setting some notation.

To that end, the algorithm as described in Section 4.1 iterates between MCTS and supervised learning. In particular, let $\ell \geq 1$ denote the iteration index. Let $m_\ell$ be the number of states that are sampled uniformly at random, independently, over $\mathcal{S}$ in this iteration, denoted as $S^{(\ell)} = \{s_i^{(\ell)} : i \in [m_\ell]\}$. Let $V^{(\ell-1)}$ be the input of value function from prior iteration, using which the MCTS algorithm with $n_\ell$ simulations obtains improved estimates of value function for states in $S^{(\ell)}$ denoted as $\hat{V}^{(\ell)}(s_i^{(\ell)})$, $i \in [m_\ell]$. Using $(s_i^{(\ell)}, \hat{V}^{(\ell)}(s_i^{(\ell)}))_{i\in[m_\ell]}$, the nearest neighbor supervised learning as described above with balls of appropriate radius $\delta_\ell \in (0, 1)$ produces estimate $V^{(\ell)}$ for all states in $\mathcal{S}$. Let $\mathcal{F}^{(\ell)}$ denote the smallest $\sigma$-algebra containing all information pertaining to the algorithm (both MCTS and supervised learning). Define the error under MCTS in iteration $\ell$ as

$$\varepsilon_{\text{mcts}}^{(\ell)} = \mathbb{E}\Big[\sup_{s\in\mathcal{S}} \big|\mathbb{E}\big[\hat{V}^{(\ell)}(s)\big|\mathcal{F}^{(\ell-1)}\big] - V^*(s)\big|\Big]. \qquad (50)$$

And, the error for supervised learning in iteration $\ell$ as

$$\theta_{\text{sl}}^{(\ell)} = \sup_{s\in\mathcal{S}} \big|V^{(\ell)}(s) - V^*(s)\big|, \text{ and } \varepsilon_{\text{sl}}^{(\ell)} = \mathbb{E}\big[\theta_{\text{sl}}^{(\ell)}\big]. \qquad (51)$$

Recall that in the beginning, we set $V^{(0)}(s) = 0$ for all $s \in \mathcal{S}$. Since $V^*(\cdot) \in [-V_{\max}, V_{\max}]$, we have that $\varepsilon_{\text{sl}}^{(0)} \leq V_{\max}$. Further, it is easy to see that if the leaf estimates (i.e., the output of the supervised learning from the previous iteration) is bounded in $[-V_{\max}, V_{\max}]$, then the output of the MCTS algorithm is always bounded in $[-V_{\max}, V_{\max}]$. That is, since $V^{(0)}(s) = 0$ and the nearest neighbor supervised learning produces estimate $V^{(l)}$ via simple averaging, inductively, the output of the MCTS algorithm is always bounded in $[-V_{\max}, V_{\max}]$ throughout every iteration.

With the notation as set up above, it follows that for a given $\delta_\ell \in (0, 1)$ with $m_\ell$ satisfying condition of Lemma 8, i.e. $m_\ell = \Omega(d\delta_\ell^{-d-2}\log\delta_\ell^{-1})$, and with the nearest neighbor supervised learning using $\delta_\ell$ radius balls for estimation, we have the following recursion:

$$\varepsilon_{\text{sl}}^{(\ell)} \leq \varepsilon_{\text{mcts}}^{(\ell)} + (C+1)\delta_\ell + \frac{4V_{\max}\delta_\ell^2}{K(\delta_\ell, d)} \leq \varepsilon_{\text{mcts}}^{(\ell)} + C'\delta_\ell, \qquad (52)$$

where $C'$ is a large enough constant, since $\frac{\delta_\ell^2}{K(\delta_\ell,d)} = \Theta(d\delta_\ell^{d+2})$ which is $O(\delta_\ell)$ for all $\delta_\ell \in (0, 1)$. By Theorem 1, for iteration $\ell + 1$ that uses the output of supervised learning estimate, $V^{(\ell)}$, as the input to the MCTS algorithm, we obtain

$$\big|\mathbb{E}\big[\hat{V}^{(\ell+1)}(s)\big|\mathcal{F}^{(\ell)}\big] - V^*(s)\big| \leq \gamma^{H^{(\ell+1)}}\mathbb{E}\big[\theta_{\text{sl}}^{(\ell)}\big|\mathcal{F}^{(\ell)}\big] + O\big(n_{\ell+1}^{\eta-1}\big), \forall s \in \mathcal{S}, \qquad (53)$$

where $\eta \in [1/2, 1)$ is the constant utilized by MCTS with fixed height of tree being $H^{(\ell+1)}$. This then implies that

$$\begin{aligned}\varepsilon_{\text{mcts}}^{(\ell+1)} &= \mathbb{E}\big[\sup_{s\in\mathcal{S}} \big|\mathbb{E}\big[\hat{V}^{(\ell+1)}(s)\big|\mathcal{F}^{(\ell)}\big] - V^*(s)\big|\big] \\ &\le \gamma^{H^{(\ell+1)}}\mathbb{E}\Big[\mathbb{E}\big[\theta_{\text{sl}}^{(\ell)}\big|\mathcal{F}^{(\ell)}\big]\Big] + O\big(n_{\ell+1}^{\eta-1}\big) \\ &\le \gamma^{H^{(\ell+1)}}\Big(\varepsilon_{\text{mcts}}^{(\ell)} + C'\delta_\ell\Big) + O\big(n_{\ell+1}^{\eta-1}\big). \end{aligned} \tag{54}$$

Denote by $\lambda \triangleq (\frac{\varepsilon}{V_{\max}})^{1/L}$. Note that since the final desired error $\varepsilon$ should be less than $V_{\max}$ (otherwise, the problem is trivial by just outputing 0 as the final estimates for all the states), we have $\lambda < 1$. Let us set the algorithmic parameters for MCTS and nearest neighbor supervised learning as follows: for each $\ell \ge 1$,

$$H^{(\ell)} = \Big\lceil \log_\gamma \frac{\lambda}{8}\Big\rceil, \delta_\ell = \frac{3V_{\max}}{4C'}\lambda^\ell, n_\ell = \kappa_l\Big(\frac{8}{V_{\max}\lambda^\ell}\Big)^{\frac{1}{1-\eta}}, \tag{55}$$

where $\kappa_l > 0$ is a sufficiently large constant such that $O\big(n_\ell^{\eta-1}\big) = \frac{V_{\max}}{8}\lambda^\ell$. Substituting these values into Eq. (54) yields

$$\varepsilon_{\text{mcts}}^{(\ell+1)} = \mathbb{E}\Big[\sup_{s\in\mathcal{S}}\big|\mathbb{E}\big[\hat{V}^{(\ell+1)}(s)|\mathcal{F}^{(\ell)}\big] - V^*(s)\big|\Big] \le \frac{\lambda}{8}\varepsilon_{\text{mcts}}^{(\ell)} + \frac{7V_{\max}}{32}\lambda^{\ell+1}.$$

Note that by (53) and (55), and the fact that $\varepsilon_{\text{sl}}^{(0)} \le V_{\max}$, we have

$$\varepsilon_{\text{mcts}}^{(1)} \le \frac{\lambda}{8}\varepsilon_{\text{sl}}^{(0)} + \frac{\lambda}{8}V_{\max} \le \frac{\lambda}{4}V_{\max}.$$

It then follows inductively that

$$\varepsilon_{\text{mcts}}^{(\ell)} \le \lambda^{\ell-1}\varepsilon_{\text{mcts}}^{(1)} = \frac{V_{\max}}{4}\lambda^\ell.$$

As for the supervised learning oracle, $\forall s \in \mathcal{S}$, Eq. (52) implies

$$\mathbb{E}\big[\sup_{s\in\mathcal{S}}\big|V^{(\ell)}(s) - V^*(s)\big|\big] \le \varepsilon_{\text{mcts}}^{(\ell)} + \frac{3V_{\max}}{4}\lambda^\ell \le V_{\max}\lambda^\ell.$$

This implies that

$$\mathbb{E}\big[\sup_{s\in\mathcal{S}}\big|V^{(L)}(s) - V^*(s)\big|\big] \le V_{\max}\lambda^L = \varepsilon.$$

We now calculate the sample complexity, i.e., the total number of state transitions required for the algorithm. During the $\ell$-th iteration, each query of MCTS oracle requires $n_\ell$ simulations. Recall that the number of querying MCTS oracle, i.e., the size of training set $\mathcal{S}^{(\ell)}$ for the nearest neighbor supervised step, should satisfy $m_\ell = \Omega(d\delta_\ell^{-d-2}\log\delta_\ell^{-1})$ (cf. Lemma 8). From Eq. (55), we have

$$H^{(\ell)} = c_0'\log\lambda^{-1}, \quad \delta^{(\ell)} = c_1\lambda^\ell, \text{and} \quad n_\ell = c_2'\lambda^{-\ell/(1-\eta)},$$

where $c_0', c_1, c_2'$, are constants independent of $\lambda$ and $\ell$. Note that each simulation of MCTS samples $H^{(\ell)}$ state transitions. Hence, the number of state transitions at the $\ell$-th iteration is given by

$$M^{(\ell)} = m_\ell n_\ell H^{(\ell)}.$$

Therefore, the total number of state transitions after $L$ iterations is

$$\sum_{l=1}^{L} M^{(\ell)} = \sum_{\ell=1}^{L} m_\ell \cdot n_\ell \cdot H^{(\ell)} = O\Big(\varepsilon^{-\left(2+1/(1-\eta)+d\right)} \cdot \big(\log\frac{1}{\varepsilon}\big)^5\Big).$$

That is, for optimal choice of $\eta = 1/2$, the total number of state transitions is $O\big(\varepsilon^{-(4+d)} \cdot \big(\log\frac{1}{\varepsilon}\big)^5\big)$.

## D.3 Proof of Lemma 8

PROOF. Given $N$ samples $s_i, i \in [N]$ that are sampled independently and uniformly at random over $\mathcal{S}$, and given the fact that each ball $c_i,\ i \in [K(\delta, d)]$ has at least $C_d\delta^d$ volume shared with $\mathcal{S}$, each of the sample falls within a given ball with probability at least $C_d\delta^d$. Let $N_i,\ i \in [K(\delta, d)]$ denote the number of samples amongst $N$ samples in ball $c_i$.

Now the number of samples falling in any given ball is lower bounded by a Binomial random variable with parameter $N, C_d\delta^d$. By Chernoff bound for Binomial variable with parameter $n, p$, we have that

$$\mathbb{P}(B(n,p) \le np/2) \le \exp\big(-\frac{np}{8}\big).$$

Therefore, with an application of union bound, each ball has at least $0.5C_d\delta^d N$ samples with probability at least $1 - K(\delta, d)\exp\big(-C_d\delta^d N/8\big)$. That is, for $N = 32\max(1, \delta^{-2}V_{\max}^2)C_d^{-1}\delta^{-d}[\log(K(\delta, d)+\log\delta^{-1}]$, each ball has at least $\Gamma = 16\max(1, \delta^{-2}V_{\max}^2)(\log K(\delta, d)+\log\delta^{-1})$ samples with probability at least $1 - \frac{\delta^2}{K(\delta,d)}$. Define event

$$\mathcal{E}_1 = \{N_i \ge 16\max(1, \delta^{-2}V_{\max}^2)(\log K(\delta, d) + \log\delta^{-1}),\ \forall\, i \in [K(\delta, d)]\}. \tag{56}$$

Then

$$\mathbb{P}(\mathcal{E}_1^c) \le \frac{\delta^2}{K(\delta, d)}.$$

Now, for any $s \in \mathcal{S}$, the nearest neighbor supervised learning described in Section 4.2 produces estimate $\hat{V}(s)$ equal to the average value of observations for samples falling in ball $c_{j(s)}$. Let $N_{j(s)}$ denote the number of samples in ball $c_{j(s)}$. To that end,

$$\begin{aligned}\big|\hat{V}(s) - V^*(s)\big| &= \Big|\frac{1}{N_{j(s)}}\Big(\sum_{i:s_i\in c_{j(s)}} V(s_i) - V^*(s)\Big)\Big| \\ &= \Big|\frac{1}{N_{j(s)}}\Big(\sum_{i:s_i\in c_{j(s)}} V(s_i) - \mathbb{E}[V(s_i)]\Big)\Big| \\ &\quad + \Big|\frac{1}{N_{j(s)}}\Big(\sum_{i:s_i\in c_{j(s)}} \mathbb{E}[V(s_i)] - V^*(s_i)\Big)\Big| \\ &\quad + \Big|\frac{1}{N_{j(s)}}\Big(\sum_{i:s_i\in c_{j(s)}} V^*(s_i) - V^*(s)\Big)\Big|. \end{aligned}$$

For the first term, since for each $s_i \in c_{j(s)}$, $V(s_i)$ is produced using independent randomness via MCTS, and since the output $V(s_i)$ is a bounded random variable, using Hoeffding's inequality, it follows that

$$\mathbb{P}\Big(\Big|\frac{1}{N_{j(s)}}\Big(\sum_{i:s_i\in c_{j(s)}} V(s_i) - \mathbb{E}[V(s_i)]\Big)\Big| \ge \Delta_1\Big) \le 2\exp\Big(-\frac{N_{j(s)}\Delta_1^2}{8V_{\max}^2}\Big).$$

The second term is no more than $\Delta$ due to the guarantee given by MCTS as assumed in the setup. And finally, the third term is no more than $C\delta$ due to Lipschitzness of $V^*$. To summarize, with probability at least $1 - 2\exp\Big(-\frac{N_{j(s)}\Delta_1^2}{8V_{\max}^2}\Big)$, we have that

$$\big|\hat{V}(s) - V^*(s)\big| \le \Delta_1 + \Delta + C\delta.$$

As can be noticed, the algorithm produces the same estimate for all $s \in \mathcal{S}$ such that they map to the same ball. And there are $K(\delta, d)$ such balls. Therefore, using union bound, it follows that with probability at least $1 - 2K(\delta, d) \exp\Big( - \frac{(\min_{i \in [K(\delta,d)]} N_i)\Delta_1^2}{8V_{\max}^2}\Big)$,

$$\sup_{s \in \mathcal{S}} \left|\hat{V}(s) - V^*(s)\right| \leq \Delta_1 + \Delta + C\delta.$$

Under event $\mathcal{E}_1$, $\min_{i \in [K(\delta,d)]} N_i \geq 16 \max(1, \delta^{-2} V_{\max}^2)(\log K(\delta, d) + \log \delta^{-1})$. Therefore, under event $\mathcal{E}_1$, by choosing $\Delta_1 = \delta$, we have

$$\sup_{s \in \mathcal{S}} \left|\hat{V}(s) - V^*(s)\right| \leq \Delta + (C+1)\delta,$$

with probability at least $1 - \frac{2\delta^2}{K(\delta,d)}$. When event $\mathcal{E}_1$ does not hold or the above does not hold, we have trivial error bound of $2V_{\max}$ on the error. Therefore, we conclude that

$$\mathbb{E}\Big[ \sup_{s \in \mathcal{S}} \left|\hat{V}(s) - V^*(s)\right| \Big] \leq \Delta + (C+1)\delta + \frac{4V_{\max}\delta^2}{K(\delta, d)}.$$

□