Maximum Entropy Monte-Carlo Planning

Chenjun Xiao[1] **Jincheng Mei**[1] **Ruitong Huang**[2] **Dale Schuurmans**[1] **Martin Müller**[1]
[1]University of Alberta
[2]Borealis AI
{chenjun, jmei2, daes, mmueller}@ualberta.com, ruitong.huang@borealisai.com

Abstract

We develop a new algorithm for online planning in large scale sequential decision problems that improves upon the worst case efficiency of UCT. The idea is to augment Monte-Carlo Tree Search (MCTS) with maximum entropy policy optimization, evaluating each search node by softmax values back-propagated from simulation. To establish the effectiveness of this approach, we first investigate the single-step decision problem, stochastic softmax bandits, and show that softmax values can be estimated at an optimal convergence rate in terms of mean squared error. We then extend this approach to general sequential decision making by developing a general MCTS algorithm, *Maximum Entropy for Tree Search* (MENTS). We prove that the probability of MENTS failing to identify the best decision at the root decays exponentially, which fundamentally improves the polynomial convergence rate of UCT. Our experimental results also demonstrate that MENTS is more sample efficient than UCT in both synthetic problems and Atari 2600 games.

1 Introduction

Monte Carlo planning algorithms have been widely applied in many challenging problems [13, 14]. One particularly powerful and general algorithm is the Monte Carlo Tree Search (MCTS) [4]. The key idea of MCTS is to construct a search tree of states that are evaluated by averaging over outcomes from simulations. MCTS provides several major advantages over traditional online planning methods. It breaks the curse of dimensionality by simulating state-action trajectories using a domain generative model, and building a search tree online by collecting information gathered during the simulations in an incremental manner. It can be combined with domain knowledge such as function approximations learned either online [20] or offline [13, 14]. It is highly selective, where bandit algorithm are applied to balance between exploring the most uncertain branches and exploiting the most promising ones [10]. MCTS has demonstrated outstanding empirical performance in many game playing problems, but most importantly, it is provable to converge to the optimal policy if the exploitation and exploration balanced appropriately [10, 8].

The convergence property of MCTS highly replies on the state value estimations. At each node of the search tree, the value estimation is also used to calculate the value of the action leading to that node. Hence, the convergence rate of the state value estimation influences the rate of convergence for states further up in the tree. However, the Monte Carlo value estimate (average over simulations outcomes) used in MCTS does not enjoy effective convergence guarantee when this value is back-propagated in the search tree, since for any given node, the sampling policy in the subtree is changing and the payoff sequences experienced will drift in time. In summary, the compounding error, caused by the structure of the search tree as well as the uncertainty of the Monte Carlo estimation, makes that UCT can only guarantee a polynomial convergence rate of finding the best action at the root.

Ideally, one would like to adopt a state value that can be efficiently estimated and back-propagated in a search tree. In this paper, we exploit the usage of *softmax value estimate* in MCTS based on the maximum entropy policy optimization framework. To establish the effectiveness of this approach,

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

we first propose a new *stochastic softmax bandit* framework for the single-step decision problem, and show that softmax values can be estimated in a sequential manner at an optimal convergence rate in terms of mean squared error. Our next contribution is to extend this approach to general sequential decision making by developing a general MCTS algorithm, *Maximum Entropy for Tree Search* (MENTS). We contribute new observations that the softmax state value can be efficiently back-propagated in the search tree, which enables the search algorithm to achieve faster convergence rate towards finding the optimal action at the root. Our theoretical analysis shows that MENTS enjoys an exponential convergence rate to the optimal solution, improving the polynomial convergence rate of UCT fundamentally. Our experiments also demonstrate that MENTS is much more sample efficient compared with UCT in practice.

2 Background

2.1 Online Planning in Markov Decision Process

We focus on the episodic Markov decision process (MDP) [1], which is formally defined as a 5-tuple $\{\mathcal{S}, \mathcal{A}, P, R, H\}$. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space. H is the maximum number of steps at each episode, P and R are the transition and reward functions, such that $P(\cdot|s,a)$ and $R(s,a)$ give the next state distribution and reward of taking action a at state s. We assume the transition and reward functions are deterministic for simplicity, while all of our techniques can easily generalize to the case with stochastic transitions and rewards, with an appropriate dependence on the variances of the transition and reward distributions. The solution of an MDP is a policy π that maps any state s to a probability distribution over actions. The optimal policy maximizes, on expectation, the cumulative sum of rewards, defined as,

$$G_t = \sum_{k=0}^{H+1} R_{t+k}, \qquad R_t = \begin{cases} R(s_t, a_t), & t \leq H \\ \nu(s_{H+1}), & t = H+1 \end{cases}$$

where we assume an oracle function ν that assigns stochastic evaluations for states at the end of episode. We note that this definition can also be used as a general formulation for planning algorithms in infinite horizon MDP, since H can be considered as the maximum search depth, and a stochastic evaluation function is applied at the end. We assume ν is subgaussian and has variance σ^2.

For policy π, the *state value function* $V^\pi(s)$, is defined to be the expected sum of rewards from s, $V^\pi(s) = \mathbb{E}^\pi[G_t|s_t = s]$. The *state-action value function*, also known as the Q-value, is defined similarly, $Q^\pi(s,a) = \mathbb{E}^\pi[G_t|s_t = s, a_t = a]$. The *optimal value functions* are the maximum value achievable by any policy, $V^*(s) = \max_\pi V^\pi(s)$, $Q^*(s,a) = \max_\pi Q^\pi(s,a)$. The *optimal policy* is defined by the greedy policy with respect to Q^*, $\pi^*(s) = \text{argmax}_a Q^*(s,a)$. It is well known that the optimal values can be recursively defined by the Bellman optimality equation,

$$Q^*(s,a) = R(s,a) + \mathbb{E}_{s'|s,a}\left[V^*(s')\right], \qquad V^*(s) = \max_a Q^*(s,a). \tag{1}$$

We consider the *online planning* problem that uses a *generative model* of the MDP to compute the optimal policy at any input state, given a fixed sampling budget. The generative model is a randomized algorithm that can output the reward $R(s,a)$ and sample a next state s' from $P(\cdot|s,a)$, given a state-action pair (s,a) as the input. For example, in the game of Go, if the rules of the game are known, the next board state can be predicted exactly after a move. To solve the online planning problem, an algorithm uses the generative model to sample an episode at each round, and proposes an action for the input state after the sampling budget is expended. The performance of an online planning algorithm can be measured by its probability of proposing the optimal action for the state of interest.

2.2 Monte Carlo Tree Search and UCT

To solve the online planning task, Monte Carlo Tree Search (MCTS) builds a *look-ahead tree* $\mathcal{T}$ online in an incremental manner, and evaluates states with Monte Carlo simulations [4]. Each node in $\mathcal{T}$ is labeled by a state s, and stores a value estimate $Q(s,a)$ and visit count $N(s,a)$ for each action a. The estimate $Q(s,a)$ is the mean return of all simulations starting from s and a. The root of $\mathcal{T}$ is

[1] All of our approaches can extend to infinite horizon MDP.

labeled by the state of interest. At each iteration of the algorithm, one simulation starts from the root of the search tree, and proceeds in two stages: a *tree policy* is used to select actions while within the tree until a leaf of $\mathcal{T}$ is reached. An evaluation function is used at the leaf to obtain a simulation return. Typical choices of the evaluation function include function approximation with a neural network, and Monte Carlo simulations using a *roll-out policy*. The return is propagated upwards to all nodes along the path to the root. $\mathcal{T}$ is grown by expanding the leaf reached during the simulation.

Bandit algorithms are used to balance between exploring the most uncertain branches and exploiting the most promising ones. The UCT algorithm applies UCB1 as its tree policy to balance the growth of the search tree [10]. At each node of $\mathcal{T}$, its tree policy selects an action with the maximum upper confidence bound

$$\text{UCB}(s,a) = Q(s,a) + c\sqrt{\frac{\log N(s)}{N(s,a)}},$$

where $N(s) = \sum_a N(s,a)$, and c is a parameter controlling exploration. The UCT algorithm has proven to be effective in many practical problems. The most famous example is its usage in AlphaGo [13, 14]. UCT is asymptotically optimal: the value estimated by UCT converges in probability to the optimal value, $Q(s,a) \xrightarrow{p} Q^*(s,a), \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$. The probability of finding a suboptimal action at the root converges to zero at a rate of $O(\frac{1}{t})$, where t is the simulation budget [10].

2.3 Maximum Entropy Policy Optimization

The maximum entropy policy optimization problem, which augments the standard expected reward objective with a entropy regularizer, has recently drawn much attention in the reinforcement learning community [5, 6, 12]. Given K actions and the corresponding K-dimensional reward vector $\mathbf{r} \in \mathbb{R}^K$, the entropy regularized policy optimization problem finds a policy by solving

$$\max_{\pi} \Big\{ \pi \cdot \mathbf{r} + \tau \mathcal{H}(\pi) \Big\}. \tag{2}$$

where $\tau \geq 0$ is a user-specified temperature parameter which controlls the degree of exploration. The most intriguing fact about this problem is that it has a closed form solution. Define the *softmax* $\mathcal{F}_\tau$ and the *soft indmax* $\mathbf{f}_\tau$ functions,

$$\mathbf{f}_\tau(\mathbf{r}) = \exp\{(\mathbf{r} - \mathcal{F}_\tau(\mathbf{r}))/\tau\} \qquad \mathcal{F}_\tau(\mathbf{r}) = \tau \log \sum\nolimits_a \exp(r(a)/\tau).$$

Note that the softmax $\mathcal{F}_\tau$ outputs a scalar while the soft indmax $\mathbf{f}_\tau$ maps any reward vector $\mathbf{r}$ to a Boltzmann policy. $\mathcal{F}_\tau(\mathbf{r})$, $\mathbf{f}_\tau(\mathbf{r})$ and (2) are connected by as shown in [5, 12],

$$\mathcal{F}_\tau(\mathbf{r}) = \max_{\pi} \Big\{ \pi \cdot \mathbf{r} + \tau \mathcal{H}(\pi) \Big\} = \mathbf{f}_\tau(\mathbf{r}) \cdot \mathbf{r} + \tau \mathcal{H}(\mathbf{f}_\tau(\mathbf{r})). \tag{3}$$

This relation suggests the softmax value is an upper bound on the maximum value, and the gap can be upper bounded by the product of τ and the maximum entropy. Note that as $\tau \to 0$, (2) approaches the standard expected reward objective, where the optimal solution is the hard-max policy. Therefore, it is straightforward to generalize the entropy regularized optimization to define the *softmax value functions*, by replacing the hard-max operator in (1) with the softmax operators [5, 12],

$$Q^*_{\text{sft}}(s,a) = R(s,a) + \mathbb{E}_{s'|s,a}\left[V^*_{\text{sft}}(s')\right], \qquad V^*_{\text{sft}}(s) = \tau \log \sum\nolimits_a \exp\Big\{ Q^*_{\text{sft}}(s,a)/\tau \Big\}. \tag{4}$$

Finally, according to (3), we can characterize the optimal *softmax policy* by,

$$\pi^*_{\text{sft}}(a|s) = \exp\Big\{ \left(Q^*_{\text{sft}}(s,a) - V^*_{\text{sft}}(s)\right)/\tau \Big\}. \tag{5}$$

In this paper, we combine the maximum entropy policy optimization framework with MCTS, by estimating the softmax values backpropagated from simulations. Specifically, we show that the softmax values can be efficiently backpropagated in the search tree, which leads to a faster convergence rate to the optimal policy at the root.

3 Softmax Value Estimation in Stochastic Bandit

We begin by introducing the *stochastic softmax bandit* problem. We provide an asymptotical lower bound of this problem, propose a new bandit algorithm for it and show a tight upper bound on its convergence rate. Our upper bound matches the lower bound not only in order, but also in the coefficient of the dominating term. All proofs are provided in the supplementary material.

3.1 The Stochastic Softmax Bandit

Consider a stochastic bandit setting with arms set $\mathcal{A}$. At each round t, a learner chooses an action $A_t \in \mathcal{A}$. Next, the environment samples a random reward R_t and reveals it to the learner. Let $r(a)$ be the expected value of the reward distribution of action $a \in \mathcal{A}$. We assume $r(a) \in [0, 1]$, and that all reward distributions are σ^2-subgaussian [2]. For round t, we define $N_t(a)$ as the number of times a is chosen so far, and $\hat{r}_t(a)$ as the empirical estimate of $r(a)$,

$$N_t(a) = \sum_{i=1}^{t} \mathbb{I}\{A_t = a\} \qquad \hat{r}_t(a) = \sum_{i=1}^{t} \mathbb{I}\{A_i = a\} R_i / N_t(a),$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Let $\mathbf{r} \in [0, 1]^K$ be the vector of expected rewards, and $\hat{\mathbf{r}}_t$ be the empirical estimates of $\mathbf{r}$ at round t. We denote $\pi^*_{\text{sft}} = \mathbf{f}_\tau(\mathbf{r})$ the optimal soft indmax policy defined by the mean reward vector $\mathbf{r}$. The stochastic bandit setting can be considered as a special case of an episodic MDP with $H = 1$.

In a stochastic softmax bandit problem, instead of finding the policy with maximum expected reward as in original stochastic bandits [11], our objective is to estimate the softmax value $V^*_{\text{sft}} = \mathcal{F}_\tau(\mathbf{r})$ for some $\tau > 0$. We define $U^* = \sum_a \exp\{r(a)/\tau\}$ and $U_t = \sum_a \exp\{\hat{r}_t(a)/\tau\}$, and propose to use the estimator $V_t = \mathcal{F}_\tau(\hat{\mathbf{r}}_t) = \tau \log U_t$. Our goal is to find a sequential sampling algorithm that can minimize the mean squared error, $\mathcal{E}_t = \mathbb{E}[(U^* - U_t)^2]$. The randomness in $\mathcal{E}_t$ comes from both the sampling algorithm and the observed rewards. Our first result gives a lower bound on $\mathcal{E}_t$.

Theorem 1. *In the stochastic softmax bandit problem, for any algorithm that achieves $\mathcal{E}_t = O(\frac{1}{t})$, there exists a problem setting such that*

$$\lim_{t\to\infty} t\mathcal{E}_t \geq \frac{\sigma^2}{\tau^2} \left(\sum_a \exp(r(a)/\tau) \right)^2 .$$

Also, to achieve this lower bound, there must be for any $a \in \mathcal{A}$, $\lim_{t\to\infty} N_t(a)/t = \pi^*_{\text{sft}}(a)$.

Note that in Theorem 1, we only assume $\mathcal{E}_t = O(1/t)$, but not that the algorithm achieves (asymptotically) unbiased estimates for each arm. Furthermore, this lower bound also reflects the consistency between the softmax value and the soft indmax policy (3): in order to achieve the lower bound on the mean squared error, the sampling policy must converge to π^*_{sft} asymptotically.

3.2 E2W: an Optimal Sequential Sampling Strategy

Inspired by the lower bound, we propose an optimal algorithm, Empirical Exponential Weight (E2W), for the stochastic softmax bandit problem. The main idea is very intuitive: enforce enough exploration to guarantee good estimation of $\hat{\mathbf{r}}$, and make the policy converge to π^* asymptotically, as suggested by the lower bound. Specifically, at round t, the algorithm selects an action by sampling from the distribution

$$\pi_t(a) = (1 - \lambda_t)\mathbf{f}_\tau(\hat{\mathbf{r}})(a) + \lambda_t \frac{1}{|\mathcal{A}|} . \tag{6}$$

In (6), $\lambda_t = \varepsilon|\mathcal{A}|/\log(t + 1)$ is a decay rate for exploration, with exploration parameter $\varepsilon > 0$. Our next theorem provides an exact convergence rate for E2W.

Theorem 2. *For the softmax stochastic bandit problem, E2W can guarantee,*

$$\lim_{t\to\infty} t\mathcal{E}_t = \frac{\sigma^2}{\tau^2} \left(\sum_a \exp(r(a)/\tau) \right)^2 .$$

Theorem 2 shows that E2W is an asymptotically optimal sequential sampling strategy for estimating the softmax value in stochastic multi-armed bandits. The main contribution of the present paper is the introduction of the softmax bandit algorithm for the implementation of tree policy in MCTS. In our proposed new algorithm, softmax bandit is used as the fundamental tool both for estimating each state's softmax value, and balancing the growth of the search tree.

[2] For prudent readers, we follow the finite horizon bandits setting in [11], where the probability space carries the tuple of random variables $S_T = \{A_0, R_0, \ldots, A_T, R_T\}$. For every time step $t - 1$ the historical observation defines a σ-algebra $\mathcal{F}_{t-1}$ and A_t is $\mathcal{F}_{t-1}$-measurable, the conditional distribution of A_t is our policy at time π_t, and the conditoinal distribution of the reward $R_{A_t} - r(A_t)$ is a martingale difference sequence.

4 Maximum Entropy MCTS

We now describe the main technical contributions of this paper, which combine maximum entropy policy optimization with MCTS. Our proposed method, MENTS (Maximum Entropy for Tree Search), applies a similar algorithmic design as UCT (see Section 2.2) with two innovations: using E2W as the tree policy, and evaluating each search node by softmax values back-propagated from simulations.

4.1 Algorithmic Design

Let $\mathcal{T}$ be a look-ahead search tree built online by the algorithm. Each node $n(s) \in \mathcal{T}$ is labeled by a state s, contains a softmax value estimate $Q_{\text{sft}}(s,a)$, and a visit count $N(s,a)$ for each action a. We use $\mathbf{Q}_{\text{sft}}(s)$ to denote a $|\mathcal{A}|$-dimensional vector with components $Q_{\text{sft}}(s,a)$. Let $N(s) = \sum_a N(s,a)$ and $V_{\text{sft}}(s) = \mathcal{F}_\tau(\mathbf{Q}_{\text{sft}}(s))$. During the in-tree phase of the simulation, the tree policy selects an action according to

$$\pi_t(a|s) = (1-\lambda_s)\mathbf{f}_\tau(\mathbf{Q}_{\text{sft}}(s))(a) + \lambda_s \frac{1}{|\mathcal{A}|} \tag{7}$$

where $\lambda_s = \varepsilon|\mathcal{A}|/\log(\sum_a N(s,a)+1)$. Let $\{s_0, a_0, s_1, a_1, \ldots, s_T\}$ be the state action trajectory in the simulation, where $n(s_T)$ is a leaf node of $\mathcal{T}$. An evaluation function is called on s_T and returns an estimate R [3]. $\mathcal{T}$ is then grown by expanding $n(s_T)$. Its statistics are initialized by $Q_{\text{sft}}(s_T,a) = 0$ and $N(s_T,a) = 0$ for all actions a. For all nodes in the trajectory, we update the visiting counts by $N(s_t,a_t) = N(s_t,a_t) + 1$, and the Q-values using a *softmax backup*,

$$Q_{\text{sft}}(s_t,a_t) = \begin{cases} r(s_t,a_t) + R & t = T-1 \\ r(s_t,a_t) + \mathcal{F}_\tau(\mathbf{Q}_{\text{sft}}(s_{t+1})) & t < T-1 \end{cases} \tag{8}$$

The algorithm MENTS can also be extended to use domain knowledge, such as function approximations learned offline. For instance, suppose that a policy network $\tilde{\pi}(\cdot|s)$ is available. Then the statistics can be initialized by $Q_{\text{sft}}(s_T,a) = \log\tilde{\pi}(a|s_T)$ and $N(s_T,a) = 0$ for all actions a during the expansion. Finally, at each time step t, MENTS proposes the action with the maximum estimated softmax value at the root s_0; i.e. $a_t = \text{argmax}_a Q_{\text{sft}}(s_0,a)$.

4.2 Theoretical Analysis

This section provides the key steps in developing a theoretical analysis of the convergence property for MENTS. We first show that for any node in the search tree, after its subtree has been fully explored, the estimated softmax value will converge to the optimal value at an exponential rate. Recall that in Theorem 1, an optimal sampling algorithm for the softmax stochastic bandit problem must guarantee $\lim_{t\to\infty} N_t(a)/t = \pi^*_{\text{sft}}(a)$ for any action a. Our first result shows that this holds for true in E2W with high probability. This directly comes from the proof of Theorem 2.

Theorem 3. *Consider E2W applied to the stochastic softmax bandit problem. Let $N^*_t(a) = \pi^*_{sft}(a)\cdot t$. Then there exists some constants C and $\tilde{C}$ such that,*

$$\mathbb{P}\left(|N_t(a) - N^*_t(a)| > \frac{Ct}{\log t}\right) \le \tilde{C}|\mathcal{A}|t\exp\left\{-\frac{t}{(\log t)^3}\right\}.$$

In the bandit case, the reward distribution of each arm is assumed to be subgaussian. However, when applying bandit algorithms at the internal nodes of a search tree, the payoff sequence experienced from each action will drift over time, since the sampling probability of the actions in the subtree is changing. The next result shows that even under this drift condition, the softmax value can still be efficiently estimated according to the backup scheme (8).

Theorem 4. *For any node $n(s) \in \mathcal{T}$, define the event,*

$$E_s = \left\{\forall a \in \mathcal{A}, |N(s,a) - N^*(s,a)| < \frac{N^*(s,a)}{2}\right\}$$

[3] We adapt a similar setting to Section 3, where R_t is replaced by the sample from the evaluation function, and the martingale assumption is extended to the the selection policy and the evaluation function on the leaves.

where $N^*(s,a) = \pi^*_{sft}(a|s) \cdot N(s)$. *For* $\epsilon \in [0,1)$, *there exist some constant* C *and* $\tilde{C}$ *such that for sufficiently large* t,

$$\mathbb{P}\left(|V_{sft}(s) - V^*_{sft}(s)| \geq \epsilon | E_s\right) \leq \tilde{C} \exp\left\{-\frac{N(s)\tau^2\epsilon^2}{C\sigma^2}\right\}.$$

Without loss of generality, we assume $Q^*(s,1) \geq Q^*(s,2) \geq \cdots \geq Q^*(s,|\mathcal{A}|)$ for any $n(s) \in \mathcal{T}$, and define $\Delta = Q^*(s,1) - Q^*(s,2)$. Recall that by (3), the gap between the softmax and maximum value is upper bounded by τ times the maximum of entropy. Therefore as long as τ is chosen small enough such that this gap is smaller than Δ, the best action also has the largest softmax value. Finally, as we are interested in the probability that the algorithm fails to find the optimal arm at the root, we prove the following result.

Theorem 5. *Let* a_t *be the action returned by MENTS at iteration* t. *Then for large enough* t *with some constant* C,

$$\mathbb{P}\left(a_t \neq a^*\right) \leq Ct \exp\left\{-\frac{t}{(\log t)^3}\right\}.$$

Remark. MENTS enjoys a fundamentally faster convergence rate than UCT. We highlight two main reasons behind this success result from the innovated algorithmic design. First, MENTS applies E2W as the tree policy during simulations. This assures that the softmax value functions used in MENTS could be effectively estimated in an optimal rate, and the tree policy would converge to the optimal softmax policy π^*_{sft} asymptotically, as suggested by Theorem 1 and Theorem 2. Secondly, Theorem 4 shows that the softmax value can also be efficiently back-propagated in the search tree. Due to these facts, the probability of MENTS failing to identify the best decision at the root decays exponentially, significantly improving the polynomial rate of UCT.

5 Related Work

Maximum entropy policy optimization is a well studied topic in reinforcement learning [5, 6, 12]. The maximum entropy formulation provides a substantial improvement in exploration and robustness, by adopting a smoothed optimization objective and acquiring diverse policy behaviors. The proposed algorithm MENTS is built on the softmax Bellman operator (4), which is used as the value propagation formula in MCTS. To our best knowledge, MENTS is the first algorithm that applies the maximum entropy policy optimization framework for simulation-based planning algorithms.

Several works have been proposed for improving UCT, since it is arguably "over-optimistic" [3] and does not explore sufficiently: UCT can take a long time to discover an optimal branch that initially looked inferior. Previous work has proposed to use flat-UCB, which enforces more exploration, as the tree policy for action selection at internal nodes [3]. Minimizing simple regret in MCTS is discussed in [2, 17]. Instead of using UCB1 as the tree policy at each node, these works employ a hybrid architecture, where a best-arm identification algorithm such as Sequential Halving [7] is applied at the upper levels, while the original UCT is used for the deeper levels of the tree.

Various value back-propagation strategies, particularly back-propagate the maximum estimated value over the children, were originally studied in [4]. It has been shown that the maximum backup is a poor option, since the Monte-Carlo estimation is too noisy when the number of simulations is low, which misguides the algorithm, particularly at the beginning of search. Complex back-propagation strategies in MCTS have been investigated in [9], where a mixture of maximum backup with the well known TD-λ operator [16] is proposed. In contrast to these approaches, MENTS exploits the softmax backup to achieve a faster convergence rate of value estimation.

6 Experiments

We evaluate the proposed algorithm, MENTS, across several different benchmark problems against strong baseline methods. Our first test domain is a *Synthetic Tree* environment. The tree has branching factor (number of actions) k of depth d. At each leaf of the tree, a standard Gaussian distribution is assigned as an evaluation function, that is each time a leaf is visited, the distribution is used to sample a stochastic return. The mean of each Gaussian distribution is determined in the following way: when initializing the environment each edge is assigned a random value, then the mean of the Gaussian

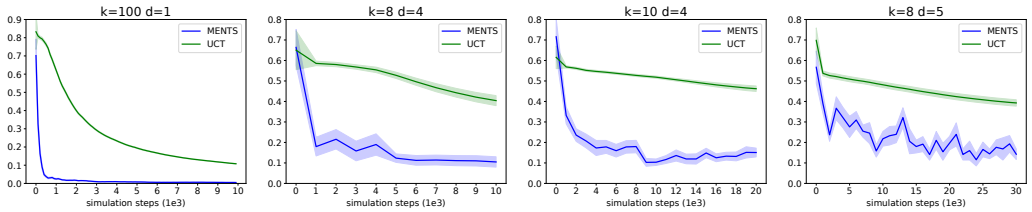

Figure 1: Evaluation of softmax value estimation in the synthetic tree environment. The x-axis shows the number of simulations and y-axis shows the value estimation error. The shaded area shows the standard error. We find that the softmax value can be efficiently estimated by MENTS.

distribution at a leaf is the sum of values along the path from the root to the leaf. This environment is similar to the P-game tree environment [10, 15] used to model two player minimax games, while here we consider the single (max) player version. Finally, we normalize all the means in $[0, 1]$.

We then test MENTS on five Atari games: *BeamRider*, *Breakout*, *Q*bert*, *Seaquest* and *SpaceInvaders*. For each game, we train a vanilla DQN and use it as an evaluation function for the tree search as discussed in the AlphaGo [13, 14]. In particular, the softmax of Q-values is used as the state value estimate, and the Boltzmann distribution over the Q-values is used as the policy network to assign a probability prior for each action when expanding a node. The temperature is set to 0.1. The UCT algorithm adopts the following tree-policy introduced in AlphaGo [14],

$$\text{PUCT}(s, a) = Q(s, a) + cP(s, a)\frac{\sqrt{\sum_b N(s, b)}}{1 + N(s, a)}$$

where $P(s, a)$ is the prior probability. MENTS also applies the same evaluation function. The prior probability is used to initialize the Q_{sft} as discussed in Section 4.1. We note that the DQN is trained using a hard-max target. Training a neural network using softmax targets such as soft Q-learning or PCL might be more suitable for MENTS [5, 12]. However, in the experiments we still use DQN in MENTS to present a fair comparison with UCT, since both algorithms apply the exactly same evaluation function. The details of the experimental setup are provided in the Appendix.

6.1 Results

Value estimation in synthetic tree. As shown in Section 4.2, the main advantage of the softmax value is that it can be efficiently estimated and back-propagated in the search tree. To verify this observation, we compare the value estimation error of MENTS and UCT in both the bandit and tree search setting. For MENTS, the error is measured by the absolute difference between the estimated softmax value $V_{\text{sft}}(s_0)$ and the true softmax state value $V^*_{\text{sft}}(s_0)$ of the root s_0. For UCT, the error is measured by the absolute difference between the Monte Carlo value estimation $V(s_0)$ and the optimal state value $V^*(s_0)$ at the root. We report the results in Figure 1. Each data point is averaged over 5×5 independent experiment (5 runs on 5 randomly initialized environment). In all of the test environments, we observe that MENTS estimates the softmax values efficiently. By comparison, we find that the Monte Carlo estimation used in UCT converges far more slowly to the optimal state value, even in the bandit setting ($d = 1$).

Online planning in synthetic tree. We next compare MENTS with UCT for online planning in the synthetic tree environment. Both algorithms use Monte Carlo simulation with uniform rollout policy as the evaluation function. The error is evaluated by $V^*(s_0) - Q^*(s_0, a_t)$, where a_t is the action proposed by the algorithm at simulation step t, and s_0 is the root of the synthetic tree. The optimal values Q^* and V^* are computed by back-propagating the true values from the leaves when the environment is initialized. Results are reported in Figure 2. As in the previous experiment, each data point is averaged over 5×5 independent experiment (5 runs on 5 randomly initialized environment). UCT converges faster than our method in the bandit environment ($d = 1$). This is because that the main advantage of MENTS is the usage of softmax state values, which can be efficiently estimated and back-propagated in the search tree. In the bandit case such an advantage does not exist. In the tree case ($d > 0$), we find that MENTS significantly outperforms UCT, especially in the large domain. For example, in synthetic tree with $k = 8$ $d = 5$, UCT fails to identify the optimal action at the root in some of the random environments, result in the large regret given the simulation budgets. However,

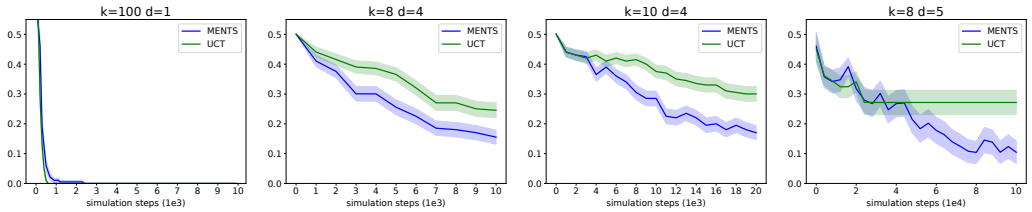

Figure 2: Evaluation of online planning in the synthetic tree environment. The x-axis shows the number of simulations and y-axis shows the planning error. The shaded area shows the standard error. We can observe that MENTS enjoys much smaller error than UCT especially in the large domain.

Table 1: Performance comparison of Atari games playing.

Agent	*BeamRider*	*Breakout*	*Q*bert*	*Seaquest*	*SpaceInvaders*
DQN	19280	345	14558	1142	625
UCT	21952	367	16010	1129	656
MENTS	18576	**386**	**18336**	1161	**1503**

MENTS can continuously make progress towards the optimal solution in all random environments, confirming MENTS scales with larger tree depth.

Online planning in Atari 2600 games. Finally, we compare MENTS and UCT using Atari games. At each time step we use 500 simulations to generate a move. Results are provided in Table 1, where we highlight scores where MENTS significantly outperforms the baselines. Scores obtained by DQN are also provided. In *Breakout*, *Q*bert* and *SpaceInvaders*, MENTS significantly outperforms UCT as well as the DQN agent. In *BeamRider* and *Seaquest* all algorithms performs similarly, since the search algorithms only use the DQN as the evaluation function and only 500 simulations are applied to generate a move. We can expect better performance when a larger simulation budget is used.

7 Conclusion

We propose a new online planning algorithm, Maximum Entropy for Tree Search (MENTS), for large scale sequential decision making. The main idea of MENTS is to augment MCTS with maximum entropy policy optimization, evaluating each node in the search tree using softmax values back-proagated from simulations. We contribute two new observations that are essential to establishing the effectiveness of MENTS: first, we study *stochastic softmax bandits* for single-step decision making and show that softmax values can be estimated at an optimal convergence rate in terms of mean squared error; second, the softmax values can be efficiently back-propagated from simulations in the search. We prove that the probability of MENTS failing to identify the best decision at the root decays exponentially, which fundamentally improves the worst case efficiency of UCT. Empirically, MENTS exhibits a significant improvement over UCT in both synthetic tree environments and Atari game playing.

Acknowledgement

The authors wish to thank Csaba Szepesvari for useful discussions, and the anonymous reviewers for their valuable advice. Part of the work is performed when the first two authors were interns at BorealisAI. This research was supported by NSERC, the Natural Sciences and Engineering Research Council of Canada, and AMII, the Alberta Machine Intelligence Institute.

References

[1] Peter J Bickel and Kjell A Doksum. *Mathematical statistics: basic ideas and selected topics, volume I*, volume 117. CRC Press, 2015.

[2] Tristan Cazenave. Sequential halving applied to trees. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):102–105, 2015.

[3] Pierre-Arnaud Coquelin and Rémi Munos. Bandit algorithms for tree search. In *Uncertainty in Artificial Intelligence*, 2007.

[4] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.

[5] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[7] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 1238–1246, 2013.

[8] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2-3):193–208, 2002.

[9] Piyush Khandelwal, Elad Liebman, Scott Niekum, and Peter Stone. On the analysis of complex backup strategies in monte carlo tree search. In *International Conference on Machine Learning*, pages 1319–1328, 2016.

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

[11] Tor Lattimore and Csaba Szepesvári. Bandit algorithms. 2018.

[12] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785, 2017.

[13] 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. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

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

[15] Stephen JJ Smith and Dana S Nau. An analysis of forward pruning. In *AAAI*, 1994.

[16] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[17] David Tolpin and Solomon Eyal Shimony. Mcts based on simple regret. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[18] JM Wainwright. Mathematical statistics, chapter 2. *STAT 210B lecture note*, 2015.

[19] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[20] Chenjun Xiao, Jincheng Mei, and Martin Müller. Memory-augmented monte carlo tree search. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A Experimental Details

We provide the experiment details in this section.

Value estimation in synthetic tree. For all settings, we use $\tau = 0.01$ for the softmax value. The exploration parameters for both MENTS and UCT are tuned from $\{0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0\}$.

Online planning in synthetic tree. The exploration parameters for MENTS and UCT are tuned from $\{0.1, 0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0\}$. The temperature parameter τ of MENTS is tuned from $\{0.5, 0.1, 0.05, 0.01, 0.005\}$.

Online planning in Atari 2600 games. The exploration parameter for both algorithms are tuned from $\{5.0, 2.0, 1.0, 0.5, 0.1\}$ The temperature parameter τ of MENTS is tuned from $\{0.1, 0.05, 0.01\}$. The results is averaged over ten environment restarts.

In games such as *BeamRider*, one test game will take thousands of environment steps. Therefore, we only test the algorithms within 10,000 environment steps. The search algorithms are used every 10 steps. For the other steps the agent will use the DQN to select action.

B Proofs for softmax stochastic bandit

We first introduce a Lemma that approximates the exponential function of empirical estimator using delta method [1]. This Lemma will be used for both lower bound and upper bound analysis.

Lemma 1. *Let $X_1, \ldots, X_n$ be i.i.d. random variables, such that $\mathbb{E}[X_i] = \mu$ and $\mathbb{V}[X_i] = \sigma^2 < \infty$, $\bar{X}_n = \sum_{i=1}^n X_i/n$. The first two moment of $\exp\left(\bar{X}_n/\tau\right)$ could be approximated by,*

$$\mathbb{E}\left[\exp\left(\frac{\bar{X}_n}{\tau}\right)\right] = e^{\mu/\tau} + \frac{\sigma^2}{2n}\left(\frac{e^{\mu/\tau}}{\tau^2}\right) + R\left(n\right) \tag{9}$$

$$\mathbb{V}\left[\exp\left(\frac{\bar{X}_n}{\tau}\right)\right] = \frac{\sigma^2}{n}\left(\frac{e^{\mu/\tau}}{\tau}\right)^2 + R'\left(n\right) \tag{10}$$

where $|R(n)| \leq O\left(n^{-2}\right), |R'(n)| \leq O\left(n^{-2}\right)$.

Proof. By Taylor's expansion,

$$\exp\left(\frac{\bar{X}_n}{\tau}\right) = e^{\mu/\tau} + \frac{e^{\mu/\tau}}{\tau}\left(\bar{X}_n - \mu\right) + \frac{e^{\mu/\tau}}{2\tau^2}\left(\bar{X}_n - \mu\right)^2 + \frac{e^{\xi/\tau}}{6\tau^3}\left(\bar{X}_n - \mu\right)^3$$

for some ξ between μ and $\bar{X}_n$. Taking the expectation on both sides,

$$\mathbb{E}\left[\exp\left(\frac{\bar{X}_n}{\tau}\right)\right] = e^{\mu/\tau} + 0 + \frac{e^{\mu/\tau}}{2\tau^2}\mathbb{V}\left[\bar{X}_n\right] + \frac{e^{\xi/\tau}}{6\tau^3}\mathbb{E}\left[\left(\bar{X}_n - \mu\right)^3\right].$$

Let $R(n) = \frac{e^{\xi/\tau}}{6\tau^3}\mathbb{E}\left[\left(\bar{X}_n - \mu\right)^3\right]$. By Lemma 5.3.1 of [1], $|R(n)| \leq O(n^{-2})$, which gives Eq. (9).

Furthermore, note that

$$\begin{aligned}\left(\mathbb{E}\left[\exp\left(\frac{\bar{X}_n}{\tau}\right)\right]\right)^2 &= \left(e^{\mu/\tau} + \frac{\sigma^2}{2n}\left(\frac{e^{\mu/\tau}}{\tau^2}\right) + R(n)\right)^2 \\ &= e^{2\mu/\tau} + \frac{\sigma^2}{n}\left(\frac{e^{\mu/\tau}}{\tau}\right)^2 + \frac{C_1}{n^2} \\ &\quad + C_2 R(n) + C_3\frac{R(n)}{n}\end{aligned}$$

for some constant C_1, C_2, C_3. On the other hand, following the same idea of deriving Eq. (9),

$$\mathbb{E}\left[\left(\exp\left(\frac{\bar{X}_n}{\tau}\right)\right)^2\right] = e^{2\mu/\tau} + \frac{2\sigma^2}{n}\left(\frac{e^{\mu/\tau}}{\tau}\right)^2 + \tilde{R}(n)$$

where $|\tilde{R}(n)| \leq O(n^{-2})$. The proof of Eq. (10) ends by taking the difference of the above two equations. □

B.1 Proof of Theorem 1

We consider the learning problem in a Bayesian setting. In the stochastic bandit problem, we assume the expected reward of each action $r(a)$ is independently sampled from a Gaussian prior $\mathcal{N}(0, \sigma_0^2)$. At time step t, for any action a, a reward $X_{a,t}$ is sampled from $\mathcal{N}(r(A_t), \sigma^2)$, independently to all the previous observations. The learner chooses an action A_t according to some policy and observe $X_t = X_{A_t,t}$. Without loss of generality, we assume that $\sigma^2 = 1$ and $\tau = 1$. Our goal is to prove

$$\lim_{t\to\infty} \mathbb{E}\left[t\left(U - \hat{U}_t\right)^2 - \frac{\sigma^2}{\tau^2}\left(\sum\nolimits_a e^{r(a)/\tau}\right)^2\right] \geq 0\,,$$

where the expectation is taken on the randomness of the algorithm, the expected rewards $\mathbf{r}$, and the observation $X_{a,i}$ given $\mathbf{r}$. Therefore the existence of $\mathbf{r}$ that provides the lower bound is guaranteed since $\mathbf{r}$ satisfies the property in expectation.

We define $\tilde{U}_t$ to be the posterior mean of U, i.e. the conditional expectation of U given the observations $X_{a,t}$. Thus, $\mathbb{E}\left[\left(U - \hat{U}_t\right)^2 - \left(U - \tilde{U}_t\right)^2\right] \geq 0$. The benefit of considering $\tilde{U}_t$ is that $\tilde{U}_t$ can further be decomposed into the Bayes estimator of each action, even without the assumption that $\hat{U}_t$ is decomposable or $\hat{U}_t$ has (asymptotic) unbiased estimator for each arm.

We next introduce two technical lemmas that are useful to prove the lower bound. The first result shows that for an algorithm that performs well on all possible environments, it must pull each arm at least in $\Omega(\log t)$ in t rounds. Note that unlike in the regret analysis for stochastic multi-armed bandits, where one only cares about how many times the suboptimal arms are pulled, the $\Omega(\log t)$ lower bound on $N_t(a)$ for suboptimal arms is not strong enough to provides a tight lower bound of $\mathcal{E}_t$.

Lemma 2. *For any algorithm $\mathcal{A}$ such that $\mathcal{E}_t = O(\frac{1}{t})$, it holds that $N_t(a) = \Omega(\log t)$ for any arm a.*

In the Bayesian learning setting defined above, since $\exp(X_{a,t})$ has a log-normal distribution with a Gaussian prior, its posterior estimation is still log-normal. The second result studies the concentration rate of the posterior estimation.

Lemma 3. *Let $\Phi(a) = \frac{\sum_{i=1}^{N_t(a)} X_{a,i} + 1/2}{\tau_0 + N_t(a)}$ be the posterior estimation of $r(a)$ and define $\Delta(a) = e^{r(a)} - e^{\Phi(a)}$. We have*

$$\mathbb{E}\left[\Delta(a) | N_t(a), \mathbf{r}\right] = O\left(\frac{1}{N_t(a)}\right)$$
$$\mathbb{E}\left[\Delta(a)^2 \,\middle|\, N_t(a), \mathbf{r}\right] = e^{2r(a)}\left(\frac{N_t(a)}{(N_t(a)+\sigma_0)^2} + O\left(\frac{1}{N_t^2(a)}\right)\right).$$

Now we are ready to present the proof of the lower bound.

Proof of Theorem 1. By the tower rule and the fact that $\tilde{U}$ is the minimizer of the mean squared error,

$$\mathbb{E}\left[t\left(U - \hat{U}_t\right)^2\right] \geq \mathbb{E}\left[t\left(U - \tilde{U}_t\right)^2\right] = \mathbb{E}\left[\mathbb{E}\left[t\left(U - \tilde{U}_t\right)^2 \,\middle|\, \mathbf{r}\right]\right],$$

It then suffices to prove that

$$\lim_{t\to\infty} \mathbb{E}\left[t\left(U - \tilde{U}_t\right)^2 \,\middle|\, \mathbf{r}\right] \geq \left(\sum\nolimits_a e^{r(a)}\right)^2$$

for any $\mathbf{r}$. The rest of the proof is always conditioned on $\mathbf{r}$. Let $\mathbf{X}_{a,t} = X_{a,1}, \ldots, X_{a,N_t(a)}$ be the observations of action a up to time step t. We can decompose $\tilde{U}$ by

$$\tilde{U}_t = \mathbb{E}\left[U \mid \mathbf{X}_{j,t}, j \in \{1,\ldots,K\}\right] = \sum_{j=1}^{K} \mathbb{E}\left[e^{r(j)} \,\middle|\, \mathbf{X}_{j,t}, j \in \{1,\ldots,K\}\right] = \sum_{j=1}^{K} \mathbb{E}\left[e^{r(j)} \,\middle|\, \mathbf{X}_{j,t}\right].$$

Therefore, the Bayesian estimator of U is

$$\tilde{U}_t = \sum_j \exp\left(\frac{\sum_{i=1}^{N_t(j)} X_{j,i} + 1/2}{\tau_0 + N_t(j)}\right).$$

It remains to bound $\left(U - \tilde{U}_t\right)^2$ conditioned on $\mathbf{r}$. Note that

$$\left(U - \tilde{U}_t\right)^2 = \left(\sum_j e^{r(j)} - \exp\left(\frac{\sum_{k=1}^{N_t(j)} X_{j,k} + 1/2}{\tau_0 + N_t(j)}\right)\right)^2 = \sum_j \Delta_j^2 + \sum_{i \neq j} \Delta_j \Delta_i,$$

where $\Delta_j = e^{r(j)} - \exp(\frac{\sum_{k=1}^{N_t(j)} X_{j,k} + 1/2}{\tau_0 + N_t(j)})$. Finally, define $P_t(j) = N_t(j)/t$ and let $\tau_0 \to 0$. By Lemma 3, we have

$$\begin{aligned}
\lim_{t\to\infty} t\mathbb{E}\left[\left(U - \tilde{U}_t\right)^2 \mid \mathbf{r}\right] &= \lim_{t\to\infty} t\mathbb{E}\left[\mathbb{E}\left[\left(U - \tilde{U}_t\right)^2 \,\middle|\, N_t(1), \ldots, N_t(k), \mathbf{r}\right]\right] \\
&= \lim_{t\to\infty} \mathbb{E}\left[\sum_j \frac{e^{2r(j)} + O\left(\frac{1}{N_t(j)}\right)}{P_t(j)}\right] \\
&\geq \left(\sum\nolimits_a e^{r(a)}\right)^2
\end{aligned}$$

where the last inequality follows by Cauchy-Schwarz inequality and Lemma 2. Note that for the inequality to hold there must be for all action $k \in [K]$, $N_t(k) = N_t^*(k)$.

For the general case, where $\sigma, \tau \neq 1$, we can simply scale the reward by τ, then the variance of $X_{j,k}$ is $\frac{\sigma^2}{\tau^2}$. The proof still holds and we obtain the following inequality,

$$\lim_{t\to\infty} t\mathbb{E}\left[\left(U - \tilde{U}_t\right)^2 \mid \mathbf{r}\right] \geq \frac{\sigma^2}{\tau^2}\left(\sum_a \bar{\pi}(a) e^{r(a)/\tau}\right)^2.$$

□

B.2 Concentration of $N_t(a)$ in Bandit (Theorem 3)

Define $\tilde{N}_t(a) = \sum_s \pi_s(a)$, where π_s is the policy followed by E2W at time step s. By Theorem 2.3 in [18] or [19], we have the following concentration result.

$$\mathbb{P}\left(|N_t(a) - \tilde{N}_t(a)| > \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2}{2\sum_{s=1}^t \sigma_s^2}\right) \leq 2\exp\left(-\frac{2\epsilon^2}{t}\right),$$

where $\sigma_s^2 \leq 1/4$ is the variance of Benoulli distribution with $p = \pi_s(k)$ at time step s. Denote the event

$$\widetilde{E}_\epsilon = \{\forall a \in \mathcal{A}, |\tilde{N}_t(a) - N_t(a)| < \epsilon\}.$$

Thus we have

$$\mathbb{P}\left(\widetilde{E}_\epsilon^c\right) \leq 2|\mathcal{A}|\exp\left(-\frac{2\epsilon^2}{t}\right).$$

It remains to bound $\mathbb{P}\left(|\tilde{N}_t(a) - N_t^*(a)| \geq \epsilon\right)$. To prove Theorem 3, we first introduce two technical lemmas, which prove the accuracy of our estimate on the reward and connect the convergence of the reward estimation to the convergence of policy.

Lemma 4. *For the stochastic softmax bandit problem, E2W can guarantee that, for $t \geq 4$,*

$$\mathbb{P}\left(\|\mathbf{r} - \hat{\mathbf{r}}_t\|_\infty \geq \frac{2\sigma}{\log(2+t)}\right) \leq 4|\mathcal{A}| \exp\left(-\frac{t}{(\log(2+t))^3}\right).$$

Lemma 5. *Given two soft indmax policies, $\pi^{(1)} = \mathbf{f}_\tau(\mathbf{r}^{(1)})$ and $\pi^{(2)} = \mathbf{f}_\tau(\mathbf{r}^{(2)})$, we have*

$$\left\|\pi^{(1)} - \pi^{(2)}\right\|_\infty \leq \left(1 + \frac{1}{\tau}\right) \left\|\mathbf{r}^{(1)} - \mathbf{r}^{(2)}\right\|_\infty$$

Proof of Theorem 3

Proof. We denote the following event,

$$E_{\mathbf{r}_t} = \left\{\|\mathbf{r} - \hat{\mathbf{r}}_t\|_\infty < \frac{2\sigma}{\log(2+t)}\right\}.$$

For any time step s and action a, by the definition of $\pi_s(a)$ we have,

$$|\pi_s(a) - \pi^*(a)| \leq |\hat{\pi}_s(a) - \pi^*(a)| + \lambda_s.$$

Thus, to bound $|\tilde{N}_t(a) - N_t^*(a)|$, conditioned on the event $\cap_{i=1}^t E_{\mathbf{r}_t}$ and for $t \geq 4$ there is,

$$\begin{aligned}
|\tilde{N}_t(a) - N_t^*(a)| \leq & \sum_{s=1}^t |\hat{\pi}_s(a) - \pi^*(a)| + \sum_{s=1}^t \lambda_s \\
\leq & \left(1 + \frac{1}{\tau}\right) \sum_{s=1}^t \|\hat{\mathbf{r}}_s - \mathbf{r}\|_\infty + \sum_{s=1}^t \lambda_s && \text{(by Lemma 5)} \\
\leq & \left(1 + \frac{1}{\tau}\right) \sum_{s=1}^t \frac{2\sigma}{\log(2+s)} + \sum_{s=1}^t \lambda_s && \text{(by Lemma 4)} \\
\leq & \left(1 + \frac{1}{\tau}\right) \int_{s=0}^t \frac{2\sigma}{\log(2+s)} \mathrm{d}s + \int_{s=0}^t \frac{|\mathcal{A}|}{\log(1+s)} \mathrm{d}s \\
\leq & \frac{Ct}{\log t},
\end{aligned}$$

for some constant C depending on $|\mathcal{A}|$, σ and τ. Finally,

$$\begin{aligned}
\mathbb{P}\left(|\tilde{N}_t(a) - N_t^*(a)| \geq \frac{Ct}{\log t}\right) \leq & \sum_{i=1}^t \mathbb{P}\left(E_{\mathbf{r}_t}^c\right) = \sum_{i=1}^t 4|\mathcal{A}| \exp\left(-\frac{t}{(\log(2+t))^3}\right) \\
\leq & 4|\mathcal{A}| t \exp\left(-\frac{t}{(\log(2+t))^3}\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \mathbb{P}\left(|N_t(a) - N_t^*(a)| \geq (1+C)\frac{t}{\log t}\right) \\
\leq & \mathbb{P}\left(|\tilde{N}_t(k) - N_t^*(k)| \geq \frac{Ct}{\log t}\right) + \mathbb{P}\left(|N_t(k) - \tilde{N}_t(k)| > \frac{t}{\log t}\right) \\
\leq & 4|\mathcal{A}| t \exp\left(-\frac{t}{\log(2+t)^3}\right) + 2|\mathcal{A}| \exp\left(-\frac{2t}{\log(2+t)^2}\right) \\
\leq & O\left(t \exp\left(-\frac{t}{(\log t)^3}\right)\right)
\end{aligned}$$

□

B.3 Proof of Theorem 2

Proof of Theorem 2. Let $\delta_t = Ct/\log t$ with some constant C. Define the following set

$$\mathcal{G}_t = \left\{ s \middle| s \in 1:t, \lceil N_t^*(a) + \delta_t \rceil \geq s \geq \lfloor N_t^*(a) - \delta_t \rfloor \right\},$$

and its complementary set $\mathcal{G}_t^{\mathsf{c}} = \{1, 2, \ldots, t\} \setminus \mathcal{G}_t$.

By Theorem 3, $\forall a \in \{1, \ldots, K\}$, with probability at least $1 - O\left(t \exp\left(-\frac{t}{(\log t)^3}\right)\right)$, $N_t(a) \in \mathcal{G}_t$. By law of total expectation and Lemma 1,

$$\begin{aligned}
\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\right] &= \sum_{s=1}^{t} \mathbb{P}\left(N_t(a) = s\right) \mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right) \middle| N_t(a) = s\right] \\
&= \sum_{s=1}^{t} \mathbb{P}\left(N_t(a) = s\right) \left(e^{r(a)/\tau} + \frac{\sigma^2}{2s}\left(\frac{e^{r(a)/\tau}}{\tau^2}\right)\right) + \sum_{s=1}^{t} \mathbb{P}\left(N_t(a) = s\right) O\left(s^{-2}\right) \\
&= \sum_{s=1}^{t} \mathbb{P}\left(N_t(a) = s\right) \left(\frac{\sigma^2}{2s}\left(\frac{e^{r(a)/\tau}}{\tau^2}\right) + O\left(s^{-2}\right)\right) + e^{r(a)/\tau}
\end{aligned} \tag{11}$$

We divide the summation in two parts. For $s \in \mathcal{G}_t^{\mathsf{c}}$, by Theorem 3,

$$\sum\nolimits_{s \in \mathcal{G}_t^{\mathsf{c}}} \mathbb{P}\left(N_t(a) = s\right) \cdot \left(\frac{\sigma^2}{2s}\left(\frac{e^{r(a)/\tau}}{\tau^2}\right) + O\left(s^{-2}\right)\right) \leq O\left(\frac{1}{t}\right) \tag{12}$$

For $s \in \mathcal{G}_t$,

$$\sum\nolimits_{s \in \mathcal{G}_t} \mathbb{P}\left(N_t(a) = s\right) \cdot \left(\frac{\sigma^2}{2s}\left(\frac{e^{r(a)/\tau}}{\tau^2}\right) + O\left(s^{-2}\right)\right) \leq O\left(\left(N_t^*(a) - \delta_t\right)^{-1}\right) \tag{13}$$

Combine the above together,

$$\begin{aligned}
t\left(U - \mathbb{E}\left[U_t\right]\right)^2 &= t\left(\sum_a \mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\right] - \exp\left(\frac{r_t(a)}{\tau}\right)\right)^2 \\
&= t\left(\sum_a O\left(\frac{1}{t}\right) + O\left(\left(N_t^*(a) - \delta_t\right)^{-1}\right)\right)^2.
\end{aligned}$$

Thus,

$$\lim_{t \to \infty} t\left(U^* - \mathbb{E}\left[U_t\right]\right)^2 = 0,$$

i.e. U_t is a consistent estimate for U^*.

To bound $\mathcal{E}_t$, it remains to bound the variance of U_t since it is unbiased. By the law of total variance,

$$\mathbb{V}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\right] = \mathbb{E}\left[\mathbb{V}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right) \middle| N_t(a)\right]\right] + \mathbb{V}\left[\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right) \middle| N_t(a)\right]\right] \tag{14}$$

Note that by Lemma 1, the first term is

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{V}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right) \middle| N_t(a)\right]\right] \\
&= \sum_{s=1}^{t} \mathbb{P}\left(N_t(a) = s\right) \mathbb{V}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right) \middle| N_t(a) = s\right] \\
&= \sum_{s=1}^{t} \mathbb{P}\left(N_t(a) = s\right) \left(\frac{\sigma^2}{s}\left(\frac{e^{r(a)/\tau}}{\tau}\right)^2 + O\left(s^{-\frac{3}{2}}\right)\right)
\end{aligned}$$

Using the same idea in Eq. (12) and Eq. (13), we consider the summation in two parts. For $s \in \mathcal{G}_t^{\mathsf{c}}$,

$$\sum\nolimits_{s\in\mathcal{G}_t^{\mathsf{c}}} \mathbb{P}\left(N_t(a)=s\right)\cdot\left(\frac{\sigma^2}{s}\left(\frac{e^{r(a)/\tau}}{\tau}\right)^2+O\left(s^{-\frac{3}{2}}\right)\right)\leq O\left(\frac{1}{t}\right)$$

For $s \in \mathcal{G}_t$,

$$\sum\nolimits_{s\in\mathcal{G}_t} \mathbb{P}\left(N_t(a)=s\right)\cdot\left(\frac{\sigma^2}{s}\left(\frac{e^{r(a)/\tau}}{\tau}\right)^2+O\left(s^{-\frac{3}{2}}\right)\right)\leq \frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{N_t^*(a)-\delta_t}+O\left(\left(N_t^*(a)-\delta_t\right)^{-\frac{3}{2}}\right)$$

Put these together we have,

$$\mathbb{E}\left[\mathbb{V}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\middle|N_t(a)\right]\right]\leq O\left(\frac{1}{t}\right)+\frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{N_t^*(a)-\delta_t}+O\left(\left(N_t^*(a)-\delta_t\right)^{-\frac{3}{2}}\right) \tag{15}$$

For the second term of Eq. (14) we have,

$$\mathbb{V}\left[\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\middle|N_t(a)\right]\right]=\mathbb{E}\left[\left(\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\middle|N_t(a)\right]\right)^2\right]-\left(\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\right]\right)^2$$

For the first term, by Lemma 1,

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\middle|N_t(a)\right]\right)^2\right]\\
&=\sum_{s=1}^{t}\mathbb{P}\left(N_t(a)=s\right)\left(\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\middle|N_t(a)\right]\right)^2\\
&=\sum_{s=1}^{t}\mathbb{P}\left(N_t(a)=s\right)\left(e^{2r(a)/\tau}+\frac{\sigma^2}{s}\left(\frac{e^{r(a)/\tau}}{\tau}\right)^2\right)+O\left(s^{-3/2}\right)\\
&\leq e^{2r(a)/\tau}+O\left(\frac{1}{t}\right)+\frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{N_t^*(a)-\delta_t}+O\left(\left(N_t^*(a)-\delta_t\right)^{-\frac{3}{2}}\right)
\end{aligned}$$

where the last inequality follows by the same idea of proving (15). For the second term, combining Eqs. (11) to (13),

$$\left(\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\right]\right)^2=\exp\left(\frac{2r(a)}{\tau}\right)+O\left(\frac{1}{t}\right)+O\left(\left(N_t^*(a)-\delta_t\right)^{-1}\right)$$

Then we have,

$$\mathbb{V}\left[\mathbb{E}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\middle|N_t(a)\right]\right]\leq O\left(\frac{1}{t}\right)+\frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{N_t^*(a)-\delta_t}+O\left(\left(N_t^*(a)-\delta_t\right)^{-1}\right) \tag{16}$$

Note that

$$\begin{aligned}
\lim_{t\to\infty} t\cdot\frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{N_t^*(a)-\delta_t}&=\lim_{t\to\infty}\frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{\pi^*(a)-\frac{\delta_t}{t}}\\
&=\frac{\sigma^2}{\tau^2}\cdot\frac{e^{r(a)/\tau}}{\bar{\pi}(a)}\cdot\left(\sum_a\bar{\pi}(a)\exp(r(a)/\tau)\right)
\end{aligned} \tag{17}$$

Combine Eq. (15), Eq. (16) and Eq. (17) together,

$$\begin{aligned}
&\lim_{t\to\infty} t\mathbb{V}\left[\hat{U}_t\right] \\
&= \lim_{t\to\infty} t\left(\sum_a \bar{\pi}^2(a)\mathbb{V}\left[\exp\left(\frac{\hat{r}_t(a)}{\tau}\right)\right]\right) \\
&\le \lim_{t\to\infty} t\sum_a \bar{\pi}^2(a)\left(O\left(\frac{1}{t}\right) + \frac{\sigma^2}{\tau^2}\cdot\frac{e^{2r(a)/\tau}}{N_t^*(a)-\delta_t}\right) \\
&\quad + t\sum_a \bar{\pi}^2(a) O\left(\left(N_t^*(a)-\delta_t\right)^{-1}\right) \\
&= \frac{\sigma^2}{\tau^2}\left(\sum_a \bar{\pi}(a)e^{r(a)/\tau}\right)^2
\end{aligned}$$

which ends the proof. □

B.4 Technical Lemmas

Proof of Lemma 2. Consider two gaussian environments ν_1 and ν_2 with unit variance. The vector of means of the reward per arm in ν_1 is $(r(1),\ldots,r(K))$ and $(r(1)+2\epsilon, r(2),\ldots,r(K))$ in ν_2. Define

$$U_1 = \sum\nolimits_{i=1}^{K} e^{r_i}, \quad U_2 = e^{r_1+2\epsilon} + \sum\nolimits_{i=2}^{K} e^{r_i}$$

Let $\mathbb{P}_1$ and $\mathbb{P}_2$ be the distribution induced by ν_1 and ν_2 respectively. Denote the event,

$$E = \left\{|\hat{U}_t - U_1| > e^{r_1}\epsilon\right\},$$

By definition, the error $\mathcal{E}_{t,\nu_1}$ under ν_1 satisfies

$$\mathcal{E}_{t,\nu_1} \ge \mathbb{P}_1(E)\,\mathbb{E}\left[(U_1-\hat{U}_t)^2 \mid E\right] \ge \mathbb{P}_1(E)\,e^{2r_1}\epsilon^2,$$

and the error $\mathcal{E}_{t,\nu_2}$ under ν_2 satisfies

$$\mathcal{E}_{t,\nu_2} \ge \mathbb{P}_2(E^c)\,\mathbb{E}\left[(U_2-\hat{U}_t)^2 \mid E^c\right] \ge \mathbb{P}_2(E^c)\,e^{2r_1}\epsilon^2.$$

Therefore, under the assumption that the algorithm suffers $O(\frac{1}{t})$ error in both environments,

$$\begin{aligned}
O(\frac{1}{t}) &= \mathcal{E}_{t,P_1} + \mathcal{E}_{t,P_2} \ge \mathbb{P}_1(E)\,e^{2r_1}\epsilon^2 + \mathbb{P}_2(E^c)\,e^{2r_1}\epsilon^2 \\
&= e^{2r_1}\epsilon^2\left(\mathbb{P}_1(E) + \mathbb{P}_2(E^c)\right) \ge \frac{1}{2}e^{2r_1}\epsilon^2 e^{-2N_t(k)\epsilon^2}.
\end{aligned}$$

where the last inequality follows by Pinsker's inequality and Divergence decomposition Lemma [11]. Therefore $N_t(k) = \Omega(\log(t))$. □

Proof of Lemma 3. Define

$$\Gamma(a) = \Phi(a) - r(a) = \frac{N_t(a)}{N_t(a)+\tau_0}(\hat{r}(a)-r(a)) + \frac{1/2-\tau_0 r(a)}{\tau_0+N_t(a)}.$$

By the fact that the variance of $X_{a,t}$ given $\mathbf{r}$ is 1,

$$\mathbb{E}\left[\Gamma(a) \mid N_t(a), \mathbf{r}\right] = \frac{1/2-\tau_0 r(a)}{\tau_0+N_t(a)}.$$

$$\mathbb{E}\left[\Gamma(a)^2 \,\middle|\, N_t(a), \mathbf{r}\right] = \frac{\sigma^2 N_t(a)}{(N_t(a)+\tau_0)^2} + O\left(\frac{1}{N_t^2(a)}\right),$$

Then we have

$$\begin{aligned}\mathbb{E}\left[\Delta(a)|N_t(a),\mathbf{r}\right] &= e^{r(a)} - \mathbb{E}\left[e^{\Phi(a)}|N_t(a),\mathbf{r}\right] \\ &= e^{r(a)}\left(1-\mathbb{E}\left[e^{\Gamma(a)}|N_t(a),\mathbf{r}\right]\right) = O\left(\frac{1}{N_t(a)}\right)\end{aligned}$$

Similarly,

$$\mathbb{E}\left[\Delta(a)^2 \,\middle|\, N_t(a),\mathbf{r}\right] = e^{2r(a)}\left(\frac{N_t(j)}{(N_t(j)+\sigma_0)^2} + O\left(\frac{1}{N_t^2(j)}\right)\right).$$

□

Proof of Lemma 4. By the choice of $\lambda_s = \frac{|\mathcal{A}|}{\log(1+s)}$, it follows that for all a and $t \geq 4$,

$$\begin{aligned}\tilde{N}_t(a) &= \sum\nolimits_{s=1}^{t} \pi_s(a) \geq \sum\nolimits_{s=1}^{t} \frac{1}{\log(1+s)} \\ &\geq \sum\nolimits_{s=1}^{t} \frac{1}{\log(1+s)} - \frac{s/(s+1)}{(\log(1+s))^2} \\ &\geq \int_1^{1+t} \frac{1}{\log(1+s)} - \frac{s/(s+1)}{(\log(1+s))^2} ds \\ &= \frac{1+t}{\log(2+t)} - \frac{1}{\log 2} \\ &\geq \frac{t}{2\log(2+t)}\end{aligned}$$

Conditioned on the event $\widetilde{E}_\epsilon$ where we set $\epsilon = \frac{t}{4\log(2+t)}$, it follows that $N_t(a) \geq \frac{t}{4\log(2+t)}$. Then, for any action a by the definition of sub-gaussian,

$$\begin{aligned}&\mathbb{P}\left(|r(a)-\hat{r}_t(a)| > \sqrt{\frac{8\sigma^2\log(\frac{2}{\delta})\log(2+t)}{t}}\right) \\ \leq &\mathbb{P}\left(|r(a)-\hat{r}_t(a)| > \sqrt{\frac{2\sigma^2\log(\frac{2}{\delta})}{N_t(a)}}\right) \leq \delta.\end{aligned}$$

Let δ satisfy that $\log(2/\delta) = \frac{t}{(\log(2+t))^3}$,

$$\mathbb{P}\left(|r(a)-\hat{r}_t(a)| > \frac{2\sigma}{\log(2+t)}\right) \leq 2\exp\left(-\frac{t}{(\log(2+t))^3}\right)$$

Therefore for $t \geq 2$

$$\begin{aligned}&\mathbb{P}\left(\left\|\mathbf{r}_t - \hat{\mathbf{r}}_t\right\|_\infty \geq \frac{2\sigma}{\log(2+t)}\right) \\ \leq &\mathbb{P}\left(\left\|\mathbf{r}_t - \hat{\mathbf{r}}_t\right\|_\infty \geq \frac{2\sigma}{\log(2+t)} \,\middle|\, \widetilde{E}_\epsilon\right) + \mathbb{P}\left(\widetilde{E}_\epsilon^c\right) \\ \leq &\sum_k \mathbb{P}\left(|r(a)-\hat{r}_t(a)| > \frac{2\sigma}{\log(2+t)} \,\middle|\, \widetilde{E}_\epsilon\right) + \mathbb{P}\left(\widetilde{E}_\epsilon^c\right) \\ \leq &2|\mathcal{A}|\exp\left(-\frac{t}{(\log(2+t))^3}\right) + 2|\mathcal{A}|\exp\left(-\frac{t}{2(\log(t+2))^2}\right) \\ \leq &4|\mathcal{A}|\exp\left(-\frac{t}{(\log(2+t))^3}\right)\end{aligned}$$

□

Proof of Lemma 5. Note that

$$\begin{aligned}\left\|\pi^{(1)}-\pi^{(2)}\right\|_\infty &\le \left\|\log\pi^{(1)}-\log\pi^{(2)}\right\|_\infty \\ &\le \frac{1}{\tau}\left\|\mathbf{r}^{(1)}-\mathbf{r}^{(2)}\right\|_\infty + \left|\mathcal{F}_\tau(\mathbf{r}^{(1)})-\mathcal{F}_\tau(\mathbf{r}^{(2)})\right|\end{aligned}$$

The proof ends by using the fact $\left|\mathcal{F}_\tau(\mathbf{r}^{(1)})-\mathcal{F}_\tau(\mathbf{r}^{(2)})\right| \le \left\|\mathbf{r}^{(1)}-\mathbf{r}^{(2)}\right\|_\infty$, which follows Lemma 8 of [12]. □

C Proofs for Tree

This section contains the detailed proof for theorems in the tree setting, in particular, Theorem 4 and Theorem 5.

C.1 Proof of Theorem 4

Proof. We prove this using induction on the depth D of tree. For the base case (D=0), the result directly follows by the fact ν is sub-gaussian. Now, at some internal node $n(s) \in \mathcal{T}$, assume the result holds for all its children, we prove the result still holds.

For any state s, we define $\mathrm{EV}(s) = \exp(V_{\text{sft}}(s)/\tau)$ and $\mathrm{EV}^*(s) = \exp(V^*_{\text{sft}}(s)/\tau)$. Note that

$$\begin{aligned}\mathrm{EV}-\mathrm{EV}^* \ge \epsilon\mathrm{EV}^* &\Leftrightarrow V \ge \tau\log(1+\epsilon)+V^* \\ \mathrm{EV}^*-\mathrm{EV} \ge \epsilon\mathrm{EV}^* &\Leftrightarrow V \le \tau\log(1-\epsilon)+V^*\end{aligned}$$

Therefore it is equivalent to prove for any node in tree,

$$\mathbb{P}\left(|\mathrm{EV}(s)-\mathrm{EV}^*(s)| \ge \epsilon\mathrm{EV}^*(s)|E_s\right) \le \tilde{C}\exp\left\{-\frac{\epsilon^2 N(s)}{C\sigma^2}\right\}$$

for some constant C and $\tilde{C}$. Note that by the definition of U we have

$$\mathrm{EV}(s) = \sum_a \exp(Q_{\text{sft}}(s,a)/\tau) = \sum_a \exp\{(r(s,a)+V_{\text{sft}}(s_a))/\tau\}$$

where s_a is the state reached by taking action a at state s. Since the reward is deterministic and bounded which only affects the scale, we can then only consider the convergence of $V_{\text{sft}}(s_a)$. Consider a decompose vector α such that $\sum_a \alpha_a \mathrm{EV}^*(s_a) = \epsilon\mathrm{EV}^*(s)$.

$$\begin{aligned}\mathbb{P}\left(|\mathrm{EV}(s)-\mathrm{EV}^*(s)| \ge \epsilon\mathrm{EV}^*(s)\,|\,E_s\right) &\le \sum_a \mathbb{P}\left(|\mathrm{EV}(s_a)-\mathrm{EV}^*(s_a)| \ge \alpha_a\mathrm{EV}^*(s_a)\,|\,E_s\right) \\ &\le \sum_a \tilde{C}_a \exp\left(-\frac{\alpha_a^2 N(s)\pi^*_{\text{sft}}(a|s)}{2C_a\sigma^2}\right),\end{aligned}$$

where the last inequality is by the induction hypothesis. Let $\alpha_a^2 \pi^*_{\text{sft}}(a|s) = M$ where $\sqrt{M} = \frac{\epsilon \text{EV}^*(s)}{\sum_a \text{EV}^*(s_a)/\sqrt{\pi^*_{\text{sft}}(a|s)}}$. One can verify that $\sum_a \alpha_a \text{EV}^*(s_a) = \epsilon \text{EV}^*(s)$. Therefore,

$$\begin{aligned}
\mathbb{P}\left(|\text{EV}(s) - \text{EV}^*(s)| \geq \epsilon \text{EV}^*(s)\right) \leq & \sum_a \tilde{C}_a \exp\left(-\frac{N(s)}{2C_a\sigma^2}\left(\frac{\epsilon \text{EV}^*(s)}{\sum_a \text{EV}^*(s_a)/\sqrt{\pi^*_{\text{sft}}(a|s)}}\right)^2\right) \\
\leq & |\mathcal{A}|\tilde{C}\exp\left(-\frac{\epsilon^2 N(s)}{2C\sigma^2}\frac{\text{EV}^*(s)^2}{\left(\sum_a \sqrt{\text{EV}^*(s)\text{EV}^*(s_a)}\right)^2}\right) \\
\leq & |\mathcal{A}|\tilde{C}\exp\left(-\frac{\epsilon^2 N(s)}{2C\sigma^2}\frac{\text{EV}^*(s)}{\left(\sum_a \sqrt{\text{EV}^*(s_a)}\right)^2}\right) \\
\leq & |\mathcal{A}|\tilde{C}\exp\left(-\frac{1}{|\mathcal{A}|}\frac{\epsilon^2 N(s)}{2C\sigma^2}\right) \\
\leq & \tilde{C}_1 \exp\left(-\frac{\epsilon^2 N(s)}{\tilde{C}_2\sigma^2}\right).
\end{aligned}$$

Picking $\tilde{C} = \max\{\tilde{C}_1, \tilde{C}_2\}$ leads to the conclusion. □

C.2 Proof of Theorem 5

Proof. Let a^* be the action with largest softmax value and s be the root state. Moreover, let $U(s_a) = \exp\left(Q_{\text{sft}}(s,a)/\tau\right)$ and $U^*(s_a) = \exp\left(Q^*_{\text{sft}}(s,a)/\tau\right)$. The event E_s is defined as in Theorem 4. The probability that MENT selects an sub-optimal arm at s is,

$$\begin{aligned}
\mathbb{P}\left(\exists a \in \mathcal{A}, U(s_a) > U(s_{a^*})\right) \leq & \mathbb{P}\left(\exists a \in \mathcal{A}, U(s_a) > U(s_{a^*}) \,|\, E_s\right) + \mathbb{P}\left(E_s^c\right) \\
\leq & \sum_a \mathbb{P}\left(U(s_a) > U(s_{a^*}) \,|\, E_s\right) + \mathbb{P}\left(E_s^c\right).
\end{aligned}$$

Since we can upper bound $\mathbb{P}\left(E_s^c\right)$ by Theorem 3, it remains to bound $\mathbb{P}\left(U(s_a) > U(s_{a^*}) \,|\, E_s\right)$.

$$\begin{aligned}
& \mathbb{P}\left(U(s_a) > U(s_{a^*}) \,|\, E_s\right) \\
= & \mathbb{P}\left(U(s_a) - U(s_{a^*}) - U^*(s_a) + U^*(s_{a^*}) > U^*(s_{a^*}) - U^*(s_a) \,|\, E_s\right) \\
\leq & \mathbb{P}\left(|U(s_{a^*}) - U^*(s_{a^*})| > \alpha U^*(s_{a^*}) \,|\, E_s\right) + \mathbb{P}\left(|U(s_a) - U^*(s_a)| > \beta U^*(s_a) \,|\, E_s\right) \\
\leq & \tilde{C}_{a^*} \exp\left\{-\frac{N^*(s,a^*)\alpha^2}{2C_{a^*}\sigma^2}\right\} + \tilde{C}_a \exp\left\{-\frac{N^*(s,a)\beta^2}{2C_a\sigma^2}\right\}
\end{aligned}$$

where $\alpha U^*(s_{a^*}) + \beta U^*(s_a) = U^*(s_{a^*}) - U^*(s_a)$. The last inequality follows by Theorem 4, since $U(s_a) - U^*(s_a) = \exp(r(s,a))\left(\exp\left(V_{\text{sft}}(s')\right) - \exp\left(V^*_{\text{sft}}(s')\right)\right)$, where s' is the state of the child of $n(s)$ taking action a. Recall that for any action a, $N^*(s,a) = t \cdot \pi^*_{\text{sft}}(a|s)$. We can choose α and β similarly as in the proof,

$$\begin{aligned}
\alpha = & \frac{(U^*(s_{a^*}) - U^*(s_a))/\sqrt{\pi^*_{\text{sft}}(a^*|s)}}{U^*(s,a)/\sqrt{\pi^*_{\text{sft}}(a|s)} + U^*(s,a^*)/\sqrt{\pi^*_{\text{sft}}(a^*|s)}} \\
\beta = & \frac{(U^*(s_{a^*}) - U^*(s_a))/\sqrt{\pi^*_{\text{sft}}(a|s)}}{U^*(s,a)/\sqrt{\pi^*_{\text{sft}}(a|s)} + U^*(s,a^*)/\sqrt{\pi^*_{\text{sft}}(a^*|s)}}.
\end{aligned}$$

Then, there exists some constant C_a and C'_a such that

$$\mathbb{P}\left(U(s_a) > U(s_{a^*}) \,|\, E_s\right) \leq C'_a \exp\left(-\frac{t}{2C_a\sigma^2}\frac{(U^*(s_{a^*}) - U^*(s_a))^2}{U^*(s)(\sqrt{U^*(s,a)} + \sqrt{U^*(s,a^*)})^2}\right).$$

We can omit the terms depending on U^* since they only affect the scale (we can switch to a new constant C'_a.) Finally, by Theorem 3,

$$\begin{aligned}
\mathbb{P}\left(\exists a \in \mathcal{A}, U(s_a) > U(s_{a^*})\right) &\leq \sum_a \mathbb{P}\left(U(s_a) > U(s_{a^*}) \mid E_s\right) + \mathbb{P}\left(E_s^c\right) \\
&\leq \sum_a C'_a \exp\left\{-\frac{t}{2C_a\sigma^2}\right\} + C't\exp\left\{-\frac{t}{(\log t)^3}\right\} \\
&\leq Ct\exp\left\{-\frac{t}{(\log t)^3}\right\}
\end{aligned}$$

for some constant C not depending on t. □