Distributional Monte-Carlo Planning with Thompson Sampling in Stochastic Environments

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We focus on a class of reinforcement learning algorithms, Monte-Carlo Tree Search (MCTS), in stochastic settings. While recent advancements combining MCTS with deep learning have excelled in deterministic environments, they face challenges in highly stochastic settings, leading to suboptimal action choices and decreased performance. Distributional Reinforcement Learning (RL) addresses these challenges by extending the traditional Bellman equation to consider value distributions instead of a single mean value, showing promising results in Deep Q Learning. In this paper, we bring the concept of Distributional RL to MCTS, focusing on modeling value functions as categorical and particle distributions. Consequently, we propose two novel algorithms: Categorical Thompson Sampling for MCTS (CATS), which uses categorical distributions for Q values, and Particle Thompson Sampling for MCTS (PATS), which models Q values with particle-based distributions. Both algorithms employ Thompson Sampling to handle action selection randomness. Our contributions are threefold: We introduce a distributional framework for Monte-Carlo Planning to model uncertainty in return estimation. We prove the effectiveness of our algorithms by achieving a non-asymptotic problem-dependent upper bound on simple regret of order $O(n^{-1})$, where n is the number of trajectories. We provide empirical evidence demonstrating the efficacy of our approach compared to baselines in both stochastic and deterministic environments.

1 Introduction

Online planning in Markov decision processes (MDPs) involves making real-time decisions based on the current state of the environment. It requires balancing exploration and exploitation while handling uncertainty and partial observability. Monte Carlo Tree Search (MCTS) is a highly effective online planning method for tackling complex MDPs. MCTS has shown impressive performance in various tasks, including traditional board games like Chess and Go, video games, and real-world challenges. Notable successes include advancements in Chess (35) and Go (34; 36; 30), video game strategy (28), robot assembly (16), robot path planning (15; 13), and autonomous driving (24).

Despite these achievements, current MCTS methods are primarily effective in deterministic environments, often overlooking the significant impact of randomness in real-world scenarios. In highly stochastic and partially observable environments, conventional MCTS approaches face substantial challenges due to widespread randomness and limited observability. This leads to compromised value estimates, suboptimal decisions, and diminished overall performance. Therefore, there is a clear need for improved methods capable of navigating the complexities of randomness and partial observability in value estimation.

We now review related works to understand the advancements and limitations in these areas.

Related work In MCTS, value estimation methods and action selection rules are critical factors for algorithm performance. Traditional value estimation methods, such as using empirical average mean

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

for value backup as in the Upper Confidence bounds applied to Trees method (UCT) (21), suffer from underestimation of optimal values while maximum backup suffers from overestimation of optimal values (9). The power mean estimator (12) offers a balanced solution by computing a mean between the average and maximum values. In our approach, we also use power mean for value operator as each V node stores the power mean of empirical means of succeeding Q-value nodes, eliminating the need for V to be modeled as a distribution.

For action selection in MCTS, strategies from Multi-Armed Bandits (MAB) are commonly employed. For instance, UCT extends the UCB1 strategy from bandits to the tree by computing confidence intervals at each step. However, original UCT's performance is hindered by the incorrect choice of logarithmic bonus constant (32). Shah et al. (32) propose an adapted version of UCT incorporating a polynomial bonus term instead of the "logarithmic" bonus term in UCT and show the non-asymtotic convergence of rate $O(n^{-1/2})$, with n is the number of rollout trajectories. On the other hand, our method improves over this rate with theoretical guarantee of $O(n^{-1})$. Although Thompson sampling has been less explored in MCTS, some approaches like those by Bai et al. (1) and Bai et al. (2) incorporate it for exploration. However, these methods lack convergence rate analysis. Furthermore, in the article Bai et al. (1), authors model value functions as a mixture of Normal distributions, which may lack the generality of complex real-world scenarios. Our approach adopts Thompson sampling for action selection but introduces a novelty by modeling the uncertainty of action value estimates over the tree as arbitrary categorical and particle-based distributions. This modification enhances our ability to handle more generality in highly stochastic environments effectively.

Entropy regularization techniques in RL modify value and action selection functions to balance exploration and exploitation, leading to improved value estimation (25; 17; 31; 18). Several works have applied these techniques in MCTS. Maximum Entropy Tree Search (MENTS) (40) emphasizes exploration by integrating MCTS with maximum entropy policy optimization. MENTS aims to maximize cumulative rewards and policy entropy concurrently, regulated by a temperature parameter. Dam et al. (14) extend MENTS by incorporating Relative and Tsallis entropy, leading to the RENTS and TENTS algorithms. However, the effectiveness of MENTS/RENTS/TENTS hinges on the temperature parameter, which may impede convergence. Furthermore, the value estimation converges exponentially to the regularized value not the optimal one. In contrast, Painter et al. (27) utilize a similar action selection approach but employ a maximum backup operator for value estimation. Although their method exhibits exponential decay of simple regret, it heavily relies on the sensitivity of the temperature parameter for Boltzmann Exploration, limiting its practicality.

Distributional Reinforcement Learning (RL) (6; 11; 22) addresses the randomness of the value estimation by introducing a distributional perspective to the traditional Bellman equation. This approach views the value function as a distribution rather than a single mean, providing a comprehensive understanding of uncertainties in rewards and the stochasticity from environments. Through discretization (26), parameterization (6), and quantization (10), it allows for efficient and effective approximation of value distributions, leading to improved performance in various RL tasks. However, these results are only for *learning* not for *planning*.

Outline and contribution In this work, we integrate the distributional approach from reinforcement learning (RL) into the *planning* framework to tackle the challenges of planning in stochastic environments. We focus on modeling value functions as categorical and particle distributions. Consequently, we propose two novel algorithms: Categorical Thompson Sampling for MCTS (CATS) and Particle Thompson Sampling for MCTS (PATS). CATS represents each Q value function as a categorical distribution and uses Thompson Sampling for action selection to manage uncertainty. PATS models each Q value function with a particle-based distribution, using a nuanced Thompson Sampling approach to handle action selection randomness.

Our contributions are threefold:

(i) In section 3, we introduce a distributional framework for *planning* to model uncertainty in return estimation, enhancing the robustness of value estimation in stochastic environments.

(ii) In section 4 Theorem 5 and Theorem 6, we prove the effectiveness of our algorithms by achieving a non-asymptotic problem-dependent upper bound on simple regret of $O(n^{-1})$, which significantly improves upon the current state-of-the-art theoretical analysis of regret, previously established at $O(n^{-1/2})$ by Shah et al. (33).

(iii) In section 5, we provide comprehensive empirical evidence demonstrating the efficacy of our approach compared to baselines, showcasing competitive performance in stochastic settings and the Atari benchmark.

In the next section, we describe the problem setting addressed in this paper.

2 Setting

In our study, We address the dynamics of an agent navigating an infinite-horizon discounted Markov decision process (MDP), defined formally as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$. Here, $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the set of actions, and $\mathcal{R}$ quantifies the Reward function of the MDP ($\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$). Transition dynamics are governed by $\mathcal{P}(\mathcal{S} \times \mathcal{A} \to \mathcal{S})$, with $\gamma \in (0, 1]$ as the discount factor. The agent interacts with the environment via a policy $\pi \in \Pi : \mathcal{S} \to \mathcal{A}$, guiding action selection based on observed states. This yields an action-value function Q^{π}, indicating the expected cumulative discounted reward from a state-action pair under π. The agent seeks the optimal policy maximizing the action-value function, adhering to the Bellman equation (7), given by $Q(s,a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s,a)[\mathcal{R}(s,a,s') + \gamma \max_{a'} Q(s',a')]ds$ for all states s and actions a. Upon acquiring the optimal action-value function, we derive the optimal value function $V(s) \triangleq \max_{a \in \mathcal{A}} Q(s,a)$ for all states s in $\mathcal{S}$.

Monte-Carlo tree search (MCTS) (20; 8) is a planning approach for complex Markov decision processes (MDPs). It employs an iterative approach:

Selection: It begins by selecting an action using a specified strategy, followed by executing this action through Monte Carlo simulation.

Expansion: Subsequently, it assesses the resulting state, either by recursively evaluating if it already exists in the search tree or by inserting it into the tree.

Simulation: Or employing a rollout policy via simulations. This iterative process continues until certain termination criteria are met, allowing traversal through the search tree.

Backpropagation: Finally, the outcomes of the simulations are propagated backward through the chosen nodes to update their statistical metrics.

Simple Regret An MCTS algorithm dynamically gathers trajectories within an MDP starting from an initial state s_0. After processing t trajectories, it provides two outputs:

- $\widehat{a}_t$, a guess for the best action to take at state s_0
- $\widehat{V}_t(s_0)$ an estimator of the optimal value in s_0,

where s_0 is the state at the root node. The algorithm's performance can be assessed by its convergence rate $r(t)$ of the simple regret, formulated as:

$$\mathbb{E}\left[R(s_0,t)\right] = \mathbb{E}\left[V^{\star}(s_0) - \widehat{V}_t\left(s_0\right)\right] \leq r(t),$$

Here, $R(s_0,t) = V^{\star}(s_0) - \widehat{V}_t(s_0)$ is the simple regret of the algorithm at the root node with $V^{\star}(s_0)$ representing the optimal value at state s_0.

In this article, we analyze an MCTS algorithm employing a maximal planning horizon H and a playout policy π_0 with value V_0. We define $\widetilde{V}(s_H) = V_0(s_H)$ recursively as follows: for all $h \leq H-1$,

$$\widetilde{Q}(s_h,a) = r(s_h,a) + \gamma \textstyle\sum_{s_{h+1} \in \mathcal{A}_{s_h}} \mathbb{P}(s_{h+1}|s_h,a)\widetilde{V}(s_{h+1}), \widetilde{V}(s_h) = \max_a \widetilde{Q}(s_h,a), \tag{1}$$

where $r(s_h,a)$ defined formally as the mean intermediate reward at state s_h after taking action a. The primary objective of an MCTS algorithm is to estimate a tied rate $r(t)$ by constructing estimates of $\widetilde{Q}(s_h,a)$ and $\widetilde{V}(s_h)$ to ultimately estimate $\widetilde{Q}(s_0,a)$ and consequently $Q^{\star}(s_0,a)$. In practical implementations of the MCTS algorithm, the maximal depth H can sometimes be set to $+\infty$. However, for theoretical analysis, the maximal depth H is crucial as we will analyze the algorithm that always collects trajectories of length H.

Distributional Reinforcement Learning The mathematical framework used in reinforcement learning is based on the Bellman equation (37), which aims to find an agent to maximize the expected utility Q value. However, the single expected value function cannot encapsulate the stochasticity in the reward function and the dynamic of the environments. Recently, in the article (5), authors shed light on the distributional perspective of the Bellman equation by modeling each Q value function as a distribution instead of a single expected value. The main objective is to study the random return $\mathcal{Q}$ at the state s, action a, and is defined recursively as

$$\mathcal{Q}(s,a) \overset{D}{=} \mathcal{X}(s,a) + \gamma \mathcal{Q}(s^{'},a^{'}), \mathcal{V}(s^{'}) \overset{D}{=} \mathbb{E}_{\pi} \mathcal{Q}(s^{'}, \pi(\cdot|s^{'})), \tag{2}$$

where $\mathcal{X}(s,a)$ is the reward distribution at the state s, action a, $\mathcal{Q}(s,a)$ is the Q value distribution at state s, action a, and $\mathcal{Q}(s^{'},a^{'})$ is the Q value distribution at state $s^{'}$, action $a^{'}$. $s^{'}$ distributed according to $\mathbb{P}(\cdot|s,a)$, $a^{'}$ distributed according to a policy $\pi(\cdot|s^{'})$. $A \overset{D}{=} B$ denotes that two random variables A and B have equal probability laws.

This distributional approach offers a deeper understanding of uncertainty and variability, especially in complex, stochastic systems where traditional expected value representations may fail to capture the true dynamics of the problem. which has been successfully used in Deep Q Learning (5).
Categorical Value Distribution Based on the distributional Bellman equation, In the article (5), authors approximate the Q value distribution $\mathcal{Q}(s,a)$ as a discrete categorical distribution parametrized by $\mathrm{N} \in \mathbb{N}$, which denotes the number of atoms (N+1) at fixed-sized locations. This method effectively divides the Q value function into a set of equally spaced atoms $z_i(s,a) = Q_{min} + i\triangle z : 0 \leq i \leq \mathrm{N}$, where Q_{min} and Q_{max} are respectively the minimum and maximum values at state s, action a. The size of each atom is set as $\triangle z := \frac{Q_{max}-Q_{min}}{\mathrm{N}}$.

This discrete distribution approach is highly expressive and computationally efficient, making it ideal for practical applications. For instance, in the article (5), authors successfully used this representation in Deep Q Learning (C51), showing promising results in several Atari games. In the next section, we demonstrate how to apply this idea to MCTS.

3 Distributional Thompson Sampling in Tree Search

In this section, we introduce two novel distributional approaches for MCTS based on Thompson sampling. The first method represents each Q-value node as a categorical distribution, while the second uses particle-based distributions for greater flexibility. Both methods integrate Thompson sampling for improved exploration and performance.

3.1 Distributional Monte-Carlo Tree Search

We leverage the success of distributional reinforcement learning (4; 3; 6) and apply this concept to MCTS. In MCTS, there are two types of nodes: V-nodes and Q-value nodes. Instead of treating each V value and Q value as a single expected value, we model these functions as distributions.
Based on equation (2), we can derive

$$\mathcal{Q}(s,a) \overset{D}{=} \mathcal{X}(s,a) + \gamma\mathcal{V}(s^{'}), \mathcal{V}(s^{'}) \overset{D}{=} \textstyle\sum_{a^{'}\sim\bar{\pi}(.|s^{'})} \mathcal{Q}(s^{'},a^{'}), \tag{3}$$

with $s^{'} \sim \mathbb{P}(\cdot|s,a)$, where $\bar{\pi}(.|s^{'})$ is formally defined as the tree policy at state s'. We can model any Q distribution with equal law distributed as the sum of the distributions of the next reward and the Q distributions of the next states actions. We further model each V distribution, having equal probability law to the expectation of the chosen policy of the next Q-value distributions (3).

Our method follows the same four basic steps of MCTS but is different in Value Backup and Action selection steps. We introduce two distinct methodologies: categorical-based and particle-based. In the categorical based approach, we parameterize each V value and Q value function in the tree as a categorical distribution. In contrast, in the particle-based approach, we model each value distribution as a set of sampling particles, representing the values observed during the tree planning. We provide a detailed explanation for the value backup and action selection of each method in the next section.

3.2 Value Backup

In this work, we employ two approaches to represent the Q value distribution.

Categorical distribution: we represent each node in the tree as a categorical distribution. In each Q-value node, we: (1) store the empirical mean value of that Q-value node (same as in UCT), and (2) maintain a categorical distribution of the Q value function. To define a categorical distribution Q function, we require three essential pieces of information:

- The number of atoms $(\mathrm{N}+1)$: We choose a consistent number of atoms $(\mathrm{N}+1)$ that remains the same for all Q distributions along the tree.
- Minimum and maximum values (*min* and *max*): Each node in the tree may have different ranges for its minimum (Q_{min})[1] and maximum (Q_{max}) values, depending on its state/action in the environment. When a new Q-value node is added to the tree, we initially set Q_{min} to 0 (assuming we have scaled the reward range to [0, R]) and initialize Q_{max} to a small

[1]Since reward is scaled in $[0,R]$, Q_{min} is not updated in our setup.

Algorithm 1 CATS

SelectAction (s_h) *(Sec 3.2)*
- **for** $a \in [A]$ **do**
  - $L(s_h, a) \sim \mathrm{Dir}(\alpha^0(s_h, a), \dots, \alpha^N(s_h, a))$
  - $\overline{\phi}(s_h, a) = [z_0(s_h, a), \dots, z_N(s_h, a)]^\top L(s_h, a)$
- $a = \arg\max_a \{\overline{\phi}(s_h, a)\}$
- **return** a

SimulateV (s_h, t) *(Sec 3.2)*
- $a =$`SelectAction` (s_h)
- `SimulateQ` (s_h, a, t)
- $T_{s_h}(t) = T_{s_h}(t) + 1$
- $\widehat{Q}(s_h, a) = \sum_i z_i(s_h, a) p_i(s_h, a)$
- $\widehat{V}(s_h) = \left(\sum_a \frac{T_{s_h,a}(t)}{T_{s_h}(t)} \widehat{Q}^p(s_h, a)\right)^{\frac{1}{p}}$

SimulateQ (s_h, a, t) *(Sec 3.2)*
- $s_{h+1} \sim \mathbb{P}(\cdot|s_h, a), r_t(s_h, a) \sim \mathcal{R}(s_h, a, s_{h+1})$
- **if** *Node* s_{h+1} *not expanded* **then**
  - `Rollout`(s_{h+1})
- **else**
  - `SimulateV` (s_{h+1}, t)
- $T_{s_h,a}(t) = T_{s_h,a}(t) + 1$
- $\overline{Q}_t(s_h, a) = r_t(s_h, a) + \gamma \widehat{V}(s_{h+1})$
- **if** $\overline{Q}_t(s_h, a) \notin [Q_{\min}(s_h, a), Q_{\max}(s_h, a)]$ **then**
  - $Q_{\max}(s_h, a) = \max\{\overline{Q}_t(s_h, a), Q_{\max}(s_h, a)\}$
  - $Q_{\min}(s_h, a) = \min\{\overline{Q}_t(s_h, a), Q_{\min}(s_h, a)\}$
  - $\triangle z = \frac{Q_{max} - Q_{min}}{\mathrm{N}}$
  - $z_i(s_h, a) = Q_{min} + i\triangle z : 0 \le i \le \mathrm{N}$
- Update $p(s_h, a) = [p_0(s_h, a), \dots, p_{\mathrm{N}}(s_h, a)]$

Algorithm 2 PATS

SelectAction (s_h) *(Sec 3.2)*
- **for** $a \in [A]$ **do**
  - $L(s_h, a) \sim \mathrm{Dir}(\alpha(s_h, a))$
  - $\overline{\phi}(s_h, a) = \mathcal{S}(s_h, a)^\top L(s_h, a)$
- $a = \arg\max_a \{\overline{\phi}(s_h, a)\}$
- **return** a

SimulateV (s_h, t) *(Sec 3.2)*
- $a =$`SelectAction` (s_h)
- `SimulateQ` (s_h, a, t)
- $T_s(t) = T_s(t) + 1$
- $\widehat{Q}(s_h, a) = \sum \alpha_t(s_h, a) \overline{Q}_t(s_h, a)$
- $\widehat{V}(s_h) = \left(\sum_a \frac{T_{s_h,a}(t)}{T_{s_h}(t)} \widehat{Q}^p(s, a)\right)^{\frac{1}{p}}$

SimulateQ (s_h, a, t) *(Sec 3.2)*
- $s_{h+1} \sim \mathbb{P}(\cdot|s_h, a), r_t(s_h, a) \sim \mathcal{R}(s_h, a, s_{h+1})$
- **if** *Node* s_{h+1} *not expanded* **then**
  - `Rollout`(s_{h+1})
- **else**
  - `SimulateV` (s_{h+1}, t)
- $T_{s_h,a}(t) = T_{s_h,a}(t) + 1$
- $\overline{Q}_t(s_h, a) = r_t(s_h, a) + \gamma \widehat{V}(s_{h+1})$
- **if** $\overline{Q}_t(s, a) \in \{\mathcal{S}(s_h, a)\}$ **then**
  - $\alpha_t(s_h, a) \mathrel{+}= 1$ //$\alpha_t(s_h, a)$: weight of $\overline{Q}_t(s_h, a)$
- **else**
  - $\mathcal{S}(s_h, a) := (\mathcal{S}(s_h, a), \overline{Q}_t(s_h, a))$
  - $\alpha(s_h, a) := (\alpha(s_h, a), 1)$

Figure 1: Comparing CATS (left) and PATS (right) The main distinction is in the Q value function backup(**SimulateQ**) and action selection function (**SelectAction**); the two methods are identical in other procedures. In CATS, we init $(\alpha^0(s, a), \dots, \alpha^N(s, a)) = (1, \dots, 1)$ and in PATS, $\mathcal{S}(s, a) = (1), \alpha(s, a) = (\varnothing)$ for each s, a.

number, e.g., $Q_{max} = 0.001$. Since the min and max values are unknown, we start with a small range, that will get updated accordingly to the scale of the observed values.

- Probabilistic parameterization: The probability of each atom $(p_i(s, a))$ is determined based on the visitation count ratio. In detail, each atom stores statistical information about the visitation count, and the probability of that atom will be calculated as the visitation count divide with the total visitation count of that Q-value node. When we backpropagate the $r_t(s, a) + \gamma \widehat{V}_t(s')$ value to a specific node, we identify the atom whose value range includes the $r_t(s, a) + \gamma \widehat{V}_t(s')$ value. At this point, we increase its visitation count.

Additionally, as we backpropagate Monte-Carlo Q values over time, we empirically adjust the Q_{min} and Q_{max} values to account for the dynamic range of Q values observed in the tree. This dynamic scaling ensures that the atom locations are effectively rescaled to adapt to the changing conditions. This representation method allows us to encapsulate the knowledge gained through exploration in the form of categorical distributions, which helps in making informed decisions during the tree search.

Paricle based distribution: We represent each Q value distribution as a collection of sampling particles, which encapsulate the observed values during tree planning. Initially, we maintain an empty set of particles for the Q value distribution, denoted as $\mathcal{S}(s, a)$. At time step t, upon receiving an intermediate reward $\overline{Q}_t(s, a) = r_t(s, a) + \gamma \widehat{V}_t(s')$, with $s' \sim \mathbb{P}(\cdot|s, a)$, we add $\overline{Q}_t(s, a)$ to the set $\mathcal{S}(s, a)$ if the particle does not already exist within it. If the particle $\overline{Q}_t(s, a)$ already exists in $\mathcal{S}(s, a)$, we increase the visitation count ratio associated with that particle.

Value function: The Q-value node is crucial in the tree because its representation influences action selection, as detailed in the next section. We now discuss modeling each V-value node. The V-value distribution is based on the expected outcomes of the chosen policy and the subsequent Q-distributions. Thus, the mean of the V-function corresponds to the tree policy's expectation of the means of all succeeding Q-value nodes. The common approach is to use empirical average mean for the value backup, as in UCT (21). However, this approach underestimates the optimal value, while using the maximum value overestimates it (9). The power mean estimator (12) provides a balanced solution, falling between the average and maximum values. In our methods, each V node stores the power mean of the empirical means of all succeeding Q-value nodes, eliminating the need to model V as a distribution.

$$\widehat{V}(s) = \left(\sum_a \frac{T_{s,a}(n)}{T_s(n)} \widehat{Q}^p(s,a) \right)^{\frac{1}{p}}, p \geq 1,$$

where $T_s(n), T_{s,a}(n)$ are the number of visitations at s and s, a at timestep n respectively. Next, we show how to select actions in the tree based on the categorical distribution of Q-value nodes.

3.3 Action Selection

Thompson sampling has shown promising results in real bandit scenarios due to the randomness of action selection. Taking advantage of the established categorical based distribution and particle based distribution, we use the Thompson sampling method for action selection. We maintain a Dirichlet distribution of parameter of the Q value distribution. We denote the Dirichlet distribution of parameters $(\alpha^0, \alpha^1, \dots, \alpha^N)$ by $\mathrm{Dir}(\alpha^0, \alpha^1, \dots, \alpha^N)$, whose density function is given by $\frac{\Gamma(\sum_{i=0}^N \alpha^i)}{\Pi_{i=0}^N \Gamma(\alpha^i)} \Pi_{i=0}^N x_i^{\alpha^i - 1}$ for $(x_0, \dots, x_N) \in [0,1]^{N+1}$ such that $\sum_{i=0}^N x_i = 1$.

Categorical distribution: The probability mass function of the discrete categorical distribution at each Q-value node at state s, action a: $p(s,a) = [p_0(s,a), p_1(s,a), \dots, p_N(s,a)]$, where $p_i(s,a)$ represents the probability of selecting the i-th atom $z_i(s,a)$, $N+1$ is the number of atoms. We maintain a Dirichlet distribution $\mathrm{Dir}(\alpha^0(s,a), \alpha^1(s,a), \dots, \alpha^N(s,a))$ as the prior for the Q-value node at state s, action a. At each time step t we sample $L_t(s,a) \sim \mathrm{Dir}(\alpha^0(s,a), \alpha^1(s,a), \dots, \alpha^N(s,a))$ and compute $\overline{\phi}_t(s,a) = [z_0(s,a), z_1(s,a), \dots, z_N(s,a)]^\top L_t(s,a)$. Then, the action a_t is selected as follows:

$$a_t = \arg\max_a \left\{ \overline{\phi}_t(s,a) \right\}$$

After taking action a_t and get an intermediate reward $\overline{Q}_t(s,a_t) = r_t(s,a_t) + \gamma \widehat{V}_t(s')$. The posterior is also a Dirichlet: $\mathrm{Dir}(\alpha^0(s,a), \dots, \alpha^t(s,a)+1, \dots, \alpha^N(s,a))$ with the intermediate reward at time step t: $\overline{Q}_t(s,a_t)$ is in the range of the atom $z_t(s,a)$. We denote this mechanism as Categorical Thompson sampling for Tree Search (CATS) method.

Paricle based distribution: In the particle-based approach, the prior Dirichlet distribution of the Q-value node at state s, action a is $\mathrm{Dir}(\alpha(s,a))$, with $\alpha(s,a)$ is initiated as $[1]$. Considering each Q value distribution at state s, action a has a set of particle $\{\overline{Q}_t(s,a)\}$ with the corresponding weighted $\alpha(s,a) = \{\alpha^t(s,a)\}$ At each time step t we also sample $L_t(s,a) \sim \mathrm{Dir}(\alpha(s,a))$ and compute $\overline{\phi}_t(s,a) = [1, \overline{Q}_0(s,a), \overline{Q}_1(s,a), \dots, \overline{Q}_N(s,a)]^\top L_t(s,a)$. Then the action a_t is chosen as

$$a_t = \arg\max_a \left\{ \overline{\phi}_t(s,a) \right\}.$$

After taking action a_t and get an intermediate reward $\overline{Q}_t(s,a_t) = r_t(s,a_t) + \gamma \widehat{V}_t(s')$. We update $\alpha^t(s,a) = \alpha^t(s,a) + 1$ if $\overline{Q}_t(s,a_t)$ is in the set $\{\overline{Q}_t(s,a)\}$. If not, we add $\overline{Q}_t(s,a_t)$ to the set $\{\overline{Q}_t(s,a)\}$ and add 1 to the set $\{\alpha^t(s,a)\} = \{\alpha^t(s,a), 1\}$.

We call this method as Paricle Thompson sampling for Tree Search (PATS) method. Detailed pseudocode and a comparison of CATS and PATS can be seen in Fig 1. The two methods are identical in all procedures except for the Q value function backup (**SimulateQ**) and the action selection function (**SelectAction**).

Remark 1. *CATS and PATS both use similar action selection strategies within a bandit setting, specifically referring to Multinomial Thompson Sampling and Non-Parametric Thompson Sampling, respectively (29). While CATS action selection heavily depends strictly on Thompson Sampling by maintaining parameters of posterior Q-value distribution, PATS is not based on the posterior sampling in the strict sense. At each step, it computes an average of the observed rewards with random weight and is a Non-Parametric approach. Furthermore, CATS maintains a fixed set of atoms, whereas in PATS, the number of particles increases depending on the observed Q values.*

In the next section, we provide a theoretical analysis of the convergence of simple regret for CATS and PATS.

Algorithm 3 CATS in Non-stationary bandits

```
Require: K arms; n: number of plays;
N + 1 support size of categorical distributions
Init (α_a^0, ..., α_a^N) = (1, ..., 1) for each a ∈ [K]
Main ()
  for t = 0,1,2,..., n do
    for a ∈ [A] do
      L_{a,t} ~ Dir(α_a^0, ..., α_a^N)
      φ̄_{a,t} = [0, R(t)/N, 2R(t)/N, ..., R(t)]^⊤ L_t
    a = arg max_a {φ̄_{a,t}}
    Pull arm a and observe reward
    R_{a,t} = mR(t)/N where m ∈ {0, 1, ... N}
    Update α_a^m = α_a^m + 1
```

Algorithm 4 PATS in Non-stationary bandits

```
Require: K arms; n: number of plays;
Init α_a = (1); S_a = (1) for each a ∈ [K]
Main ()
  for t = 0,1,2,..., n do
    for a ∈ [A] do
      L_{a,t} ~ Dir(α_a)
      φ̄_{a,t} = S_a^⊤ L_{a,t}
    a = arg max_a {φ̄_{a,t}}
    Pull arm a and observe reward R_{a,t}
    if R_{a,t} ∈ {S_a} then
      α_a^t += 1 // α_a^t : weight of R_{a,t}
    else
      S_a := (S_a, R_{a,t})
      α_a := (α_a, 1)
```

Figure 2: Comparing CATS (left) and PATS (right) in Non-stationary bandits.

4 Theoretical analysis

Planning in MCTS involves making a sequence of decisions along the tree, where each internal node functions as a non-stationary bandit, with the empirical mean drifting due to the action selection strategy. Therefore, we first study the non-stationary multi-armed bandit settings using the action selections of CATS and PATS, examining the concentration properties of the power mean backup for each arm relative to the optimal arm. We then apply these results to MCTS.

4.1 Non-stationary multi-armed bandit

We consider a class of non-stationary multi-armed bandit (MAB) problems with $K \geq 1$ arms. Let $R_{a,t}$ denote the random reward obtained by playing arm $a \in [K]$ at the time step t bounded in $[0, R]$. We consider $\widehat{\mu}_{a,n} = \frac{1}{n}\sum_{t=1}^{n} R_{a,t}$ as the average rewards collected at arm a after n plays. We first define:

Definition 1. *A sequence of estimators $(\widehat{V}_n)_{n\geq 1}$ is concentrated and convergent towards some limit V if the following two properties hold:*

(A) Concentration: For all $n \geq 1$, for all $\varepsilon > 0$, $\exists c > 0$ that $\mathbb{P}\left(|\widehat{V}_n - V| > \varepsilon\right) \leq cn^{-1}\varepsilon^{-1}$.

(B) Convergence: $\lim_{n\to\infty} \mathbb{E}[\widehat{V}_n] = V$.

In that case, we write $\underset{n\to\infty}{plim}\, \widehat{V}_n = V$.

We assume that the reward sequence $\{R_{a,t}\}, t \geq 1$ is a non-stationary process satisfying the convergence and concentration properties from Definition 1, by making the following assumption:

Assumption 1. *Consider K arms that for $a \in [K]$, let $(\widehat{\mu}_{a,n})_{n\geq 1}$ be a sequence of estimator satisfying*

$$\underset{n\to\infty}{plim}\, \widehat{\mu}_{a,n} = \mu_a.$$

The action selection of CATS and PATS follows closely as in Section 3.3 and pseudocode are shown in Fig. 2. Let us define $\widehat{\mu}_n(p) = \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \widehat{\mu}^p_{a,T_a(n)}\right)^{\frac{1}{p}}$ as the power mean value backup operator after n rounds. Here $1 \leq p < \infty$ is a constant. We denote $T_a(n)$ is the number of visitations of the arm a.

We define $\mu_\star = \max_{a\in[K]}\{\mu_a\}$ and assume that $\mu_\star$ is unique. Then, we establish the concentration and convergence properties of the power mean backup operator $\widehat{\mu}_n(p)$ towards the optimal value $\mu_\star$, as shown in Theorem 1 and Theorem 2, respectively for CATS and PATS.

Theorem 1. *For $a \in [K]$, let $(\widehat{\mu}_{a,n})_{n\geq 1}$ be a sequence of estimator satisfying $\underset{n\to\infty}{plim}\, \widehat{\mu}_{a,n} = \mu_a$ and let $\mu_\star = \max_a\{\mu_a\}$. Assume that all the estimators are bounded in $[0, R]$. We consider a bandit algorithm that selects each arm according to CATS once in each round $n \geq K$. Then, $\underset{n\to\infty}{plim}\, \widehat{\mu}_n(p) = \mu_\star$.*

Theorem 2. *For $a \in [K]$, let $(\widehat{\mu}_{a,n})_{n\geq 1}$ be a sequence of estimator satisfying $\underset{n\to\infty}{plim}\, \widehat{\mu}_{a,n} = \mu_a$ and let $\mu_\star = \max_a\{\mu_a\}$. Assume that all the estimators are bounded in $[0, R]$. We consider a bandit algorithm that selects each arm according to PATS once in each round $n \geq K$. Then, $\underset{n\to\infty}{plim}\, \widehat{\mu}_n(p) = \mu_\star$.*

Detailed proofs of the two Theorems can be found in the appendix. Based upon these results we analyse the concentration properties for any internal node and convergence of the simple regret in the MCTS in the next section.

4.2 Monte-Carlo Tree Search

Before presenting the main results (Theorem 3 Theorem 4), we first show an important Lemma

Lemma 1. *Let $(\widehat{V}_{m,n})_{n\geq 1}$, $m \in [M]$, be a sequence of estimator satisfying $\underset{n\to\infty}{plim}\,\widehat{V}_{m,n} = V_m$. Assume that there exists a constant $L > 0$ such that $L = supremum\{\widehat{V}_{m,n}\}_{n\geq 1}$. Let R_i be an iid sequence with mean μ and S_i be an iid sequence from a distribution $p = (p_1, \ldots, p_M)$ supported on $\{1, \ldots, M\}$. Introducing the random variables $N^n_m = \#|\{i \leq n : S_i = s_m\}|$, we define the sequence of estimator*

$$\widehat{Q}_n = \frac{1}{n}\sum_{i=1}^{n} R_i + \gamma \sum_{m=1}^{M} \frac{N^n_m}{n} \widehat{V}_{m,N^n_m}.$$

Then $\underset{n\to\infty}{plim}\,\widehat{Q}_n = \mu + \sum_{m=1}^{M} p_m V_m$.

The significance of Lemma 1 lies in demonstrating the concentration and convergence of an estimated Q value, conditioned on the concentration and convergence of a child V-value node. Here, $\widehat{V}_{\cdot,n}$ represents the value estimation at time step n, and R_i denotes an intermediate reward received by taking a specific action at a particular state.

Next, we first start with Theorem 3 to show the convergence and concentration of any V-Node and Q-node in the tree for CATS.

Theorem 3. *When we apply the CATS algorithm, we have*

(i) For any node s_h at the depth h^{th} in the tree, $\underset{n\to\infty}{plim}\,\widehat{Q}_n(s_h, a_k) = \widetilde{Q}(s_h, a_k)$.

(ii) For any node s_h at the depth h^{th} in the tree, $\underset{n\to\infty}{plim}\,\widehat{V}_n(s_h) = \widetilde{V}(s_h)$.

We can derive a similar result for PATS as shown in Theorem 4.

Theorem 4. *When we apply the PATS algorithm, we have*

(i) For any node s_h at the depth h^{th} in the tree, $\underset{n\to\infty}{plim}\,\widehat{Q}_n(s_h, a_k) = \widetilde{Q}(s_h, a_k)$.

(ii) For any node s_h at the depth h^{th} in the tree, $\underset{n\to\infty}{plim}\,\widehat{V}_n(s_h) = \widetilde{V}(s_h)$.

The results of Theorems 4 and 4 demonstrate that, at any node in the tree, both the V-value and Q-value nodes are convergent and concentrated. These results are applicable to any power mean backup operator of V-value nodes with $p \in [1, +\infty)$. Finally, we show important results in Theorem 5, and Theorem 6, since they show the convergence of simple regret of CATS and PATS, respectively.

Theorem 5. ***(Convergence of Simple Regret of CATS)** We have at the root node s_0,*

$$\left|\mathbb{E}\left[V^\star(s_0) - \widehat{V}_n(s_0)\right]\right| \leq O(n^{-1}).$$

Theorem 6. ***(Convergence of Simple Regret of PATS)** We have at the root node s_0,*

$$\left|\mathbb{E}\left[V^\star(s_0) - \widehat{V}_n(s_0)\right]\right| \leq O(n^{-1}).$$

Remark 2. *These results demonstrate that both CATS and PATS share the same convergence rate for value estimation at the root node of $\mathcal{O}(n^{-1})$, which improves over the rate $\mathcal{O}(n^{-1/2})$ of Fixed-Depth-MCTS (33). Furthermore, Our finding more broadly applies to the power mean estimator with $p \in [1, +\infty)$.*

5 Experiments

We compare our methods with UCT (21), Fixed-Depth-MCTS (33), MENTS (40), RENTS, TENTS (14), BTS (27) and DNG (1) in a stochastic setting (*SyntheticTree*) to highlight the benefits of CATS and PATS in stochastic environments. Additionally, we test on 17 Atari games, comparing our algorithms with DQN (base network without planning) and other non-distributional planning methods (Power-UCT (12), MENTS (40), TENTS (14)) to demonstrate CATS and PATS' competitiveness and put results in Appendix. In all settings, we use 100 atoms for CATS, and set the discount factor γ to 0.99 for Atari, and γ to 1 for *SyntheticTree*.

SyntheticTree: We evaluate CATS and PATS using the synthetic tree toy problem (14). This problem involves a tree with depth d and branching factor k. Each tree edge has a random value between 0 and 1. Returns at the leaf nodes are simulated using Gaussian distributions with means equal to the sum of edge values from the root to the leaf, and a standard deviation of 0.5. Means are normalized between 0 and 1. An agent traverses the tree from the root, aiming to find the leaf node with the highest mean value. Internal nodes give zero reward, while leaf nodes provide a reward sampled from their Gaussian distribution. We introduce stochasticity into the environment by altering the transition probabilities: there is a 50% chance of moving to the intended node and a 50% chance of moving to a different node with equal probability. We conduct 25 experiments on five trees with five runs each, covering all combinations of branching factors $k = \{2, 4, 6, 8, 10, 12, 14, 16, 100, 200\}$ and depths $d = \{1, 2, 3, 4\}$. We compute the value estimation error at the root node. Fig. 3 shows the convergence of the value estimations of CATS and PATS at the root node in the Synthetic Tree environment which shows they archives faster convergence compared to other methods.

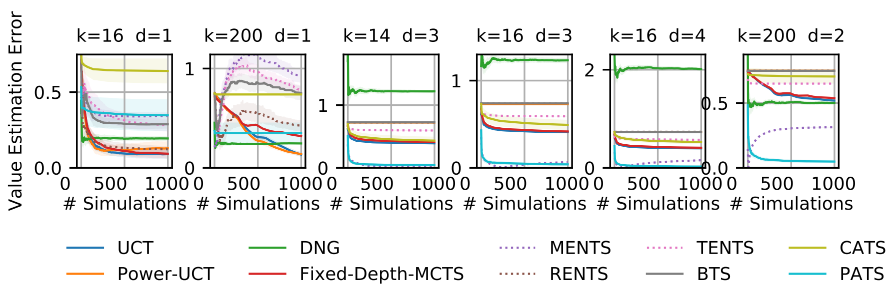

Figure 3: Performance of CATS and PATS in SyntheticTree.

6 Conclusion

To conclude, our work introduces Categorical Thompson Sampling for MCTS (CATS) and Particle Thompson Sampling for MCTS (PATS), distributional planning approaches specifically designed to tackle complexities arising from stochasticity. CATS uses a categorical distribution, while PATS uses a particle-based distribution to represent and model the uncertainty inherent in return outcomes. We also propose exploration strategies based on Thompson Sampling that leverage this distributional modeling. Our methods come with a rigorous theoretical convergence guarantee, achieving a simple regret polynomial decay of the order $O(n^{-1})$, which improves over the $O(n^{-1/2})$ rate of the fixed version of UCT (32). Empirical findings conclusively demonstrate the effectiveness of our approach in stochastic environments.

References

[1] A. Bai, F. Wu, and X. Chen. Bayesian mixture modelling and inference based thompson sampling in monte-carlo tree search. *Advances in neural information processing systems*, 26, 2013.

[2] A. Bai, F. Wu, Z. Zhang, and X. Chen. Thompson sampling based monte-carlo planning in pomdps. *the International Conference on Automated Planning and Scheduling*, 24(1), 2014.

[3] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pages 1471–1479, 2016.

[4] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[5] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, 2016.

[6] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 449–458. JMLR. org, 2017.

[7] R. Bellman. The theory of dynamic programming. Technical report, Rand corp santa monica ca, 1954.

[8] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[9] R. Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*. Springer, 2006.

[10] W. Dabney, G. Ostrovski, D. Silver, and R. Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pages 1096–1105. PMLR, 2018.

[11] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] T. Dam, P. Klink, C. D'Eramo, J. Peters, and J. Pajarinen. Generalized mean estimation in monte-carlo tree search. *arXiv preprint arXiv:1911.00384*, 2019.

[13] T. Dam, G. Chalvatzaki, J. Peters, and J. Pajarinen. Monte-carlo robot path planning. *IEEE Robotics and Automation Letters*, 7(4):11213–11220, 2022.

[14] T. Q. Dam, C. D'Eramo, J. Peters, and J. Pajarinen. Convex regularization in monte-carlo tree search. In *International Conference on Machine Learning*, pages 2365–2375. PMLR, 2021.

[15] S. Eiffert, H. Kong, N. Pirmarzdashti, and S. Sukkarieh. Path planning in dynamic environments using generative rnns and monte carlo tree search. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10263–10269. IEEE, 2020.

[16] N. Funk, G. Chalvatzaki, B. Belousov, and J. Peters. Learn2assemble with structured representations and search for robotic architectural construction. In *Conference on Robot Learning*, pages 1401–1411. PMLR, 2022.

[17] M. Geist, B. Scherrer, and O. Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169, 2019.

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

[19] J. Honda and A. Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *COLT*, pages 67–79. Citeseer, 2010.

[20] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, page 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 354045375X. doi: 10.1007/11871842_29. URL `https://doi.org/10.1007/11871842_29`.

[21] L. Kocsis, C. Szepesvári, and J. Willemson. Improved monte-carlo search. *Univ. Tartu, Estonia, Tech. Rep*, 1, 2006.

[22] B. Mavrin, H. Yao, L. Kong, K. Wu, and Y. Yu. Distributional reinforcement learning for efficient exploration. In *International conference on machine learning*, pages 4424–4434. PMLR, 2019.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[24] S. Mo, X. Pei, and C. Wu. Safe reinforcement learning for autonomous vehicle using monte carlo tree search. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6766–6773, 2021.

[25] G. Neu, A. Jonsson, and V. Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[26] T. Nguyen-Tang, S. Gupta, and S. Venkatesh. Distributional reinforcement learning via moment matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9144–9152, 2021.

[27] M. Painter, M. Baioumy, N. Hawes, and B. Lacerda. Monte carlo tree search with boltzmann exploration. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] D. Perez, S. Samothrakis, and S. Lucas. Knowledge-based fast evolutionary mcts for general video game playing. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2014.

[29] C. Riou and J. Honda. Bandit algorithms based on thompson sampling for bounded reward distributions. In *Algorithmic Learning Theory*, pages 777–826. PMLR, 2020.

[30] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[31] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[32] D. Shah, Q. Xie, and Z. Xu. Non-asymptotic analysis of monte carlo tree search. In *Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, pages 31–32, 2020.

[33] D. Shah, Q. Xie, and Z. Xu. Nonasymptotic analysis of monte carlo tree search. *Operation Research*, 70(6):3234–3260, 2022.

[34] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, Jan. 2016. doi: 10.1038/nature16961.

[35] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[36] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, Oct. 2017. URL `http://dx.doi.org/10.1038/nature24270`.

[37] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[38] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[39] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

[40] C. Xiao, R. Huang, J. Mei, D. Schuurmans, and M. Müller. Maximum entropy monte-carlo planning. In *Advances in Neural Information Processing Systems*, pages 9516–9524, 2019.

A Outline

- Notations will be described in Section B.
- Supporting Lemmas are presented in Section C.
- The Convergence of CATS and PATS in Non-stationary multi-armed bandits is shown in Section D.
- Section E presents the concentration and convergence guarantee of CATS and PATS in MCTS.
- Section F discusses about Limitations and possible improvements.
- Experimental setup is provided in Section G.
- Additional Experimental results are shown in Section H.

B Notations

Table 1: List of all notations for Non-stationary Multi-arms bandit.

Notation	Type	Description
K	$\mathbb{N}$	Number of arms
$T_a(t)$	$\mathbb{N}$	Number of visitations at arm a after t timesteps
μ_a	$\mathbb{R}$	mean value of arm a
$a_\star$	$\mathcal{A}$	optimal action
$\mu_\star$	$\mathbb{R}$	mean value of an optimal arm. We assume it is unique.
$\widehat{\mu}_n(p)$	$\mathbb{R}$	power mean estimator, with a constant $p \in [1, +\infty)$
$\widehat{\mu}_{a,n}$	$\mathbb{R}$	mean estimator of arm a after n visitations

C Supporting Lemmas

We start with a result of the following lemma which plays an important role in the analysis of our MCTS algorithm.

Lemma 1. *For $m \in [M]$, let $(\widehat{V}_{m,n})_{n \geq 1}$ be a sequence of estimator satisfying $\underset{n\to\infty}{plim}\, \widehat{V}_{m,n} = V_m$. Assume that there exists a constant $L > 0$ such that $L =$ supremum$\{\widehat{V}_{m,n}\}_{n\geq 1}$. Let R_i be an iid sequence with mean μ and S_i be an iid sequence from a distribution $p = (p_1, \dots, p_M)$ supported on $\{1, \dots, M\}$. Introducing the random variables $N_m^n = \#|\{i \leq n : S_i = s_m\}|$, we define the sequence of estimator*

$$\widehat{Q}_n = \frac{1}{n}\sum_{i=1}^{n} R_i + \gamma \sum_{m=1}^{M} \frac{N_m^n}{n} \widehat{V}_{m,N_m^n}.$$

Then there exists some constant c' (which depends on p_i (i=1,2,...,M), γ, μ) such that

$$\underset{n\to\infty}{plim}\, \widehat{Q}_n = \mu + \sum_{m=1}^{M} p_m V_m.$$

Proof. Let $p = (p_1, p_2, ...p_M), p \in \triangle^M$ where $\triangle^M = \{x \in \mathbb{R}^M : \sum_{i=1}^M R_i = 1, R_i \geq 0\}$ is the $(M-1)$-dimensional simplex. Let us study a random vector $\widehat{p}_n = (\frac{N_1^n}{n}, \frac{N_2^n}{n}, ..., \frac{N_M^n}{n})$. Let us define

Table 2: List of all notations for Monte-Carlo Tree Search.

Notation	Type	Description
γ	$\mathbb{R}$	Discount factor
N	$\mathbb{N}$	Number of atoms
s_h	$\mathcal{S}$	state at depth h
$\widehat{V}_t(s)$	$\mathbb{R}$	Estimated Value function at state s after t visitations
$T_s(t)$	$\mathbb{N}$	Number of visitations at state s after t timesteps
$T_{s,a}(t)$	$\mathbb{N}$	Number of visitations at (s,a) after t timesteps
$T_{s,a}^{s'}(t)$	$\mathbb{N}$	Number of visitations at (s,a) that goes to s' after t timesteps
$\widehat{Q}_t(s,a)$	$\mathbb{R}$	Estimated Q Value function at state s action a after t visitations
$Q_{\min}(s,a)$	$\mathbb{R}$	Minimum value for the Q value distribution at state s, action a
$Q_{\max}(s,a)$	$\mathbb{R}$	Maximum value for the Q value distribution at state s, action a
$\mathcal{R}(s,a)$		Reward distribution at state s action a
$\mathcal{V}(s)$		Value distribution at state s
$\mathcal{Q}(s,a)$		Q Value distribution at state s action a
$p_i(s,a)$	$\mathbb{R}$	Probability of the i_{th} atom at the Q Value distribution at state s action a
$\triangle z$	$\mathbb{R}$	Size of each atom
$z_i(s,a)$	$\mathbb{R}$	value of the atom i^{th} at state s, action a.
$\overline{Q}_t(s,a)$	$\mathbb{R}$	intermediate Q value at time t at (s,a)

$V = (V_1, V_2, ...V_M)$. Let $\widehat{R}_n = \frac{1}{n}\sum_{i=1}^n R_i, \widehat{V}_n = (\widehat{V}_{1,N_1^n}, \widehat{V}_{2,N_2^n}, ..., \widehat{V}_{M,N_M^n}), \sum_{i=1}^M N_i^n = n$, N_i^n is the number of times that population i was observed. We have $\widehat{Q}_n = \widehat{R}_n + \gamma \left\langle \widehat{p}_n, \widehat{V}_n \right\rangle$. Therefore,

$$\mathbb{P}\bigg(\widehat{Q}_n - \big(\mu + \gamma \langle p, V\rangle\big) \geq \epsilon\bigg) \leq \mathbb{P}\bigg(\widehat{R}_n - \mu \geq \frac{1}{2}\epsilon\bigg) + \mathbb{P}\bigg(\gamma\left\langle \widehat{p}_n, \widehat{V}_n\right\rangle - \gamma\langle p, Y\rangle \geq \frac{1}{2}\epsilon\bigg)$$
$$\leq \exp\{-2n\frac{\epsilon^2}{4}\} + \underbrace{\mathbb{P}\bigg(\left\langle \widehat{p}_n, \widehat{V}_n\right\rangle - \langle p, Y\rangle \geq \frac{1}{2\gamma}\epsilon\bigg)}_{\text{A}}.$$

To upper bound A, let us consider $\left\langle \widehat{p}_n, \widehat{V}\right\rangle - \langle p, V\rangle = \left\langle (\widehat{p}_n - p), \widehat{V}_n\right\rangle + \left\langle p, (\widehat{V} - V)\right\rangle$. Then,

$$A \leq \underbrace{\mathbb{P}\bigg(\left\langle (\widehat{p}_n - p), \widehat{V}_n\right\rangle \geq \frac{1}{4\gamma}\epsilon\bigg)}_{A_1} + \underbrace{\mathbb{P}\bigg(\left\langle p, (\widehat{V}_n - V)\right\rangle \geq \frac{1}{4\gamma}\epsilon\bigg)}_{A_2}.$$

By applying a Hölder inequality to $\widehat{p}_n - p$ and $\widehat{V}$, we obtain

$$\Big\langle (\widehat{p}_n - p), \widehat{V}_n \Big\rangle \leq \parallel \widehat{p}_n - p \parallel_1 \parallel \widehat{V}_n \parallel_\infty = \parallel \widehat{p}_n - p \parallel_1 L,$$

with L is the supremum of $\widehat{V}$. Then we can derive

$$\begin{aligned} A_1 &= \mathbb{P}\bigg(\Big\langle (\widehat{p}_n - p), \widehat{V}_n \Big\rangle \geq \frac{1}{4\gamma}\epsilon \bigg) \leq \mathbb{P}\bigg(\parallel \widehat{p}_n - p \parallel_1 L \geq \frac{1}{4\gamma}\epsilon \bigg) \\ &= \mathbb{P}\bigg(\parallel \widehat{p}_n - p \parallel_1 \geq \frac{1}{4\gamma L}\epsilon \bigg). \end{aligned}$$

According to (39), we have for any $M \geq 2$ and $\delta \in [0, 1]$

$$\mathbb{P}\bigg(\parallel \widehat{p}_n - p \parallel_1 \geq \sqrt{\frac{2M \ln(2/\delta)}{n}} \bigg) \leq \delta.$$

Define $\epsilon = \sqrt{\frac{2M \ln(2/\delta)}{n}}$, therefore $\delta = 2\exp\{\frac{-n\epsilon^2}{2M}\}$, we have

$$\mathbb{P}\bigg(\parallel \widehat{p}_n - p \parallel_1 \geq \epsilon \bigg) \leq 2\exp\{\frac{-n\epsilon^2}{2M}\}.$$

Therefore,

$$A_1 \leq \mathbb{P}\bigg(\parallel \widehat{p}_n - p \parallel_1 \geq \epsilon \bigg) \leq 2\exp\{\frac{-n\epsilon^2}{32M\gamma^2 L^2}\}.$$

We also have

$$\begin{aligned} A_2 &= \mathbb{P}\bigg(\sum_{m=1}^{M} p_m (\widehat{V}_{m,N_m^n} - V_m) \geq \frac{1}{4\gamma}\epsilon \bigg) \\ &\leq \sum_{m=1}^{M} \mathbb{E}\bigg[\mathbb{P}\bigg(\frac{1}{N_m^n} \sum_{t=1}^{N_m^n} V_{m,t} - V_m \geq \frac{1}{4\gamma p_m}\epsilon \Big| N_m^n \bigg) \bigg] \\ &\leq \sum_{m=1}^{M} \mathbb{E}\bigg[c(N_m^n)^{-1} (\frac{\epsilon}{4\gamma p_m})^{-1} \bigg]. \end{aligned}$$

Let us define an event $\mathcal{E} = \bigg\{ N_m^n \geq \frac{np_m}{2} \bigg\}$. Therefore,

$$\begin{aligned} A_2 &\leq \sum_{m=1}^{M} \mathbb{E}\bigg[c(\frac{np_m}{2})^{-1} (\frac{\epsilon}{4\gamma p_m})^{-1} \bigg] \\ &+ \sum_{m=1}^{M} \mathbb{E}\bigg[\mathbb{P}(N_m^n < \frac{np_m}{2}) \bigg] = \sum_{m=1}^{M} (c2^{1+2}\gamma^1 p_m^{-1+1}) n^{-1}\epsilon^{-1} \\ &+ \sum_{m=1}^{M} \mathbb{E}\bigg[\mathbb{P}(N_m^n - p_m n \leq -\frac{p_m n}{2}) \bigg] \\ &\leq \sum_{m=1}^{M} (c2^3\gamma) n^{-1}\epsilon^{-1} + \sum_{m=1}^{M} \exp\bigg\{ -2n(\frac{p_m n}{2})^2 \bigg\} \end{aligned}$$

We consider $p_m > 0$ only since if $p_m = 0, p_m(\widehat{V}_{m,N_m^n} - V_m) = 0$, and has been eliminated. Therefore,

$$A \leq A_1 + A_2 \leq 2\exp\{\frac{-n\epsilon^2}{32M\gamma^2 L^2}\} + \sum_{m=1}^{M} (c2^3\gamma) n^{-1}\epsilon^{-1} + \sum_{m=1}^{M} \exp\bigg\{ -2n(\frac{p_m n}{2})^2 \bigg\}.$$

That leads to

$$\mathbb{P}\bigg(\widehat{Q}_n - \big(\mu + \gamma \langle p, V\rangle\big) \geq \epsilon\bigg) \leq \exp\{-2n\frac{\epsilon^2}{4}\}$$
$$+ 2\exp\{\frac{-n\epsilon^2}{32M\gamma^2L^2}\} + \sum_{m=1}^{M}(c2^3\gamma)n^{-1}\epsilon^{-1} + \sum_{m=1}^{M}\exp\bigg\{-2n(\frac{p_m n}{2})^2\bigg\} \leq c^{'}n^{-1}\epsilon^{-1},$$

with $c^{'} > 0$ depends on c, M, p_i. So that

$$\mathbb{P}\bigg(\widehat{Q}_n - \big(\mu + \gamma \langle p, V\rangle\big) \geq \epsilon\bigg) \leq c^{'}n^{-1}\epsilon^{-1},$$

By following the same steps, we can derive

$$\mathbb{P}\bigg(\widehat{Q}_n - \big(\mu + \gamma \langle p, V\rangle\big) \leq -\epsilon\bigg) \leq c^{'}n^{-1}\epsilon^{-1}.$$

Therefore, with $n \geq 1, \epsilon > 0$,

$$\mathbb{P}\bigg(\Big|\widehat{Q}_n - \big(\mu + \gamma \langle p, V\rangle\big)\Big| \geq \epsilon\bigg) \leq c^{'}n^{-1}\epsilon^{-1}.$$

Furthermore,

$$\begin{aligned}\widehat{Q}_n - \big(\mu + \gamma \langle p, V\rangle\big) &= (\widehat{R}_n - \mu) + \bigg(\gamma\Big\langle \widehat{p}_n, \widehat{V}_n\Big\rangle - \gamma\langle p, Y\rangle\bigg)\\ &= (\widehat{R}_n - \mu) + \gamma\bigg(\Big\langle(\widehat{p}_n - p), \widehat{V}_n\Big\rangle + \Big\langle p, (\widehat{V} - V)\Big\rangle\bigg)\end{aligned}$$

Therefore,

$$\Rightarrow \Big|\mathbb{E}[\widehat{Q}_n] - \big(\mu + \gamma \langle p, V\rangle\big)\Big| \leq \Big|\mathbb{E}[(\widehat{R}_n - \mu)]\Big| + \gamma\bigg(|\mathbb{E}[\widehat{p}_n - p]|\,\Big|\widehat{V}_n\Big| + p\Big|\mathbb{E}[\widehat{V} - V]\Big|\bigg)$$
$$\Rightarrow \Big|\mathbb{E}[\widehat{Q}_n] - \big(\mu + \gamma \langle p, V\rangle\big)\Big| \leq \Big|\mathbb{E}[(\widehat{R}_n - \mu)]\Big| + \gamma\bigg(L\,|\mathbb{E}[\widehat{p}_n - p]| + p\Big|\mathbb{E}[\widehat{V} - V]\Big|\bigg)$$

Also because $\lim_{n\to\infty}\mathbb{E}[\widehat{V}_{m,n}] = V_m$, $\lim_{n\to\infty}\frac{\widehat{N}^n_m}{n} = p_m$, and $\mathbb{E}[(\widehat{R}_n - \mu)] = 0$ so that,

$$\lim_{n\to\infty}\mathbb{E}[\widehat{Q}_n] = \mu + \gamma\sum_{m=1}^{M}p_m V_m.$$

That mean

$$\operatorname*{plim}_{n\to\infty}\widehat{Q}_n = \mu + \gamma\sum_{m=1}^{M}p_m V_m,$$

which concludes the proof. $\square$

Results from Lemma 1 is important as it shows the concentration for the Q value estimation given the concentration of V value of the children nodes.

Lemma 2. *Let consider non-negative variables $x, y \in \mathbb{R}^+$, and a constant m that $0 \leq m \leq 1$. Then*

$$(x + y)^m \leq x^m + y^m.$$

Proof. With $y = 0$, or $x = 0$, the inequality (2) becomes correct. Let consider the case where $x > 0, y > 0$, the inequality (2) can be written as

$$(\frac{x}{y} + 1)^m \leq \left(\frac{x}{y}\right)^m + 1.$$

Let us define a function

$$f(t) = (t + 1)^m - t^m - 1, (t > 0).$$

We can see that

$$f^{'}(t) = m(t+1)^{m-1} - mt^{m-1} = m\left((t+1)^{m-1} - t^{m-1}\right) \leq 0 \text{ with } m \in [0,1], t > 0,$$

because $g(x) = x^{m-1}$ is a decreasing function with $m \in [0,1], x > 0$. Therefore,

$$f(t) \leq f(0) = 0 \text{ with } t > 0.$$

So that,

$$(t+1)^m - t^m - 1 \leq 0, (t > 0).$$

with $t = \frac{x}{y} \geq 0$, we can derive the inequality (2). □

We use Minkowski's inequality as shown below

Lemma 3. ***(Minkowski's inequality)*** *Given $p \geq 1, \{x_i, y_i\} \in \mathbb{R}, i = 1, 2, ..., n$, then we have the following inequality*

$$\left(\sum_i (|x_i + y_i|)^p\right)^{\frac{1}{p}} \leq \left(\sum_i (|x_i|)^p\right)^{\frac{1}{p}} + \left(\sum_i (|y_i|)^p\right)^{\frac{1}{p}}.$$

Proof. This is a basic result. □

Lemma 4. ***(Markov's inequality)*** *If X is a nonnegative random variable and $a > 0$, then the probability that X is at least a is at most the expectation of X divided by a:*

$$\mathbf{Pr}(X > a) \leq \frac{\mathbb{E}[X]}{a}.$$

Proof. This is a well-known result. □

D Convergence of CATS and PATS in Non-stationary multi-armed bandits

We note that in an MCTS tree, each node is considered a non-stationary multi-armed bandit where the average mean drifts due to the given action selection strategy. Therefore, we first study the convergence of CATS and PATS in non-stationary multi-armed bandits where the action selection is Thompson sampling, with the power mean backup operator at the root node. Detailed descriptions of the CATS and PATS in Non-stationary multi-armed bandits settings can be found in the main article in the Theoretical Analysis section.

We first establish the convergence and concentration properties for the power mean backup operator in non-stationary bandits, detailed in Theorem 1 for CATS and Theorem 2 for PATS.

To achieve these results, we demonstrate that the expected payoff of the power mean backup operator decays polynomially at a rate of $O(\frac{\log n}{n})$. This is supported by Lemma 7 for CATS and Lemma 8 for PATS. Critical to this analysis are Lemma 5 and Lemma 6, which establish an upper bound of $\log(n)$ for the expected number of suboptimal arm pulls.

We introduce some important definitions. F_a^n represents the empirical cumulative distribution function of arm a after n visitations, and F_a represents the cumulative distribution function of arm a. We employ the following distance measure: If P and Q are two distributions characterized by parameters $p = (p_0, p_1, \cdots, p_N)$ and $q = (q_0, q_1, \cdots, q_N)$ respectively, then the distance is defined as

$$d(P, Q) := \parallel p - q \parallel_\infty = \sup_{i \in [0,N]} |p_i - q_i|$$

This represents the L^∞ distance between p and q in $\mathbb{R}^{N+1}$. We also denotes $\text{KL}(P \parallel Q)$ as the Kullback–Leibler divergence between P and Q, and denote $\mathcal{K}_{\text{inf}}(F_a, \mu_\star) = \inf_{G:\mathbb{E}[G]>\mu_\star} \text{KL}(F_a \parallel G)$. In addition, we denote $\mathcal{K}_{\text{inf}}^{(N)}(F_a, \mu_\star) = \inf\left\{\text{KL}(F_a \parallel G) \middle| \text{the support of G} \in \left\{0, \frac{R}{N}, \frac{2R}{N}, \cdots, R\right\}, \mathbb{E}[G] > \mu_\star\right\}$.

We see that the definition of $\mathcal{K}_{\text{inf}}(F_a, \mu_\star)$ and $\mathcal{K}_{\text{inf}}^{(N)}(F_a, \mu_\star)$ is only difference in the support set.

We denote the true parameter of arm a by $p_a = (p_a^0, p_a^1, \dots, p_a^N)$ with $p_a^i = \mathbf{Pr}_{X\sim F_a}[X = \frac{i}{N}]$. We denote the parameter of the posterior distribution of arm a as $\alpha_a = (\alpha_a^0, \alpha_a^1, \dots, \alpha_a^N)$. Since each arm a is non-stationary, we also denote the parameter of arm a after n visitations by $p_a(n) = (p_a^0(n), p_a^1(n), \dots, p_a^N(n))$ with $p_a^i(n) = \mathbf{Pr}_{X\sim F_a^n}[X = \frac{i}{N}]$. The parameter of the posterior distribution of arm a denoted as $\alpha_a(n) = (\alpha_a^0(n), \alpha_a^1(n), \dots, \alpha_a^N(n))$ We first show the results of an important Lemma 5. The proof follows closely to the Proof of Proposition 7 (29). The only difference is that in our settings, we study non-stationary bandits.

Lemma 5. *Consider Categorical Thompson Sampling(CATS) strategy applied to a non-stationary problem where the pay-off sequence satisfies Assumption 1. Let $T_a(n)$ denote the number of plays of arm a up to timestep n.*

If a is the index of a suboptimal arm, Then for any $\epsilon_0, \epsilon_1 \geq 0$, each sub-optimal arm a is played in expectation at most

$$\mathbb{E}[T_a(n)] \leq \frac{(1+\epsilon_0)\log n}{\mathcal{K}_{inf}^{(N)}(F_a, \mu_\star) - \epsilon_1} + o(\log n) + O(1),$$

Proof. We have $\overline{\phi}_{a,t} = [0, \frac{R}{N}, \frac{2R}{N}, \cdots, R]^\top L_{a,t}$, with $L_{a,t} \sim \mathrm{Dir}(\alpha_a^0(t), \dots, \alpha_a^N(t))$.

To analyze the expectation associated with selecting a suboptimal arm a, we decompose it into two components:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}(I(t)=a)\right] =& \underbrace{\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}(I(t)=a), \overline{\phi}_{a,t} \geq \mu_* - \epsilon_1, d(\widehat{F}_{I(t)}, F_{I(t)}) \leq \epsilon_2)\right]}_{A1} \\ &+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}(I(t)=a), \overline{\phi}_{a,t} < \mu_* - \epsilon_1, d(\widehat{F}_{I(t)}, F_{I(t)}) > \epsilon_2)\right]}_{A2}\end{aligned}$$

<u>We first find an upper bound for A_1:</u>

$$A1 = \sum_{t=1}^{n}\sum_{m=1}^{n} \mathbb{1}\left(I(t)=a, \overline{\theta}_k(t) \geq \mu_\star - \epsilon_1; \| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right)$$

We see that if the event

$$\left\{I(t)=a, \overline{\theta}_k(t) \geq \mu_\star - \epsilon_1; \| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right\}$$

occurs at step t for a certain $m \in [1, n]$, then $T_k(t') > T_k(t) = m$ for any $t' > t$. Therefore, for any $m \in [n]$

$$\sum_{t=1}^{n} \mathbb{1}\left(I(t)=a, \overline{\theta}_k(t) \geq \mu_\star - \epsilon_1; \| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right) \leq 1$$

We can bound for any $m_0 \in [n]$

$$\begin{aligned}A1 \leq& m_0 + \sum_{t=1}^{n}\sum_{m=m_0}^{n} \mathbb{E}\left[\mathbb{1}\left(I(t)=a, \overline{\theta}_k(t) \geq \mu_\star - \epsilon_1; \| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right)\right] \\ \leq& m_0 + \sum_{t=1}^{n}\sum_{m=m_0}^{n} \mathbf{Pr}\left(\overline{\theta}_k(t) \geq \mu_\star - \epsilon_1; \| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right) \\ \leq& m_0 + \sum_{t=1}^{n}\sum_{m=m_0}^{n} \mathbf{Pr}\left(\overline{\theta}_k(t) \geq \mu_\star - \epsilon_1 \middle| \| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right) \\ &\times \mathbf{Pr}\left(\| \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \|_\infty \leq \epsilon_2, T_k(t)=m\right)\end{aligned} \tag{4}$$

By applying results of Lemma 13 Appendix F (29), we have

$$\begin{aligned}&\mathbf{Pr}\left(\overline{\theta}_k(t) \geq \mu_\star - \epsilon_1 \middle| \alpha_a, T_k(t) = m\right)\\ &\leq C(m+N+1)^{N/2} \exp\{-(m+N+1)\mathrm{KL}(P_{\alpha_a(t)} \parallel P^*_{\mu_\star-\epsilon_1})\}\end{aligned}$$

where $P^*_{\mu_\star-\epsilon_1} = \arg\min_{x:u^\top x \geq \mu_\star-\epsilon_1} \mathrm{KL}(P_{\alpha_a} \parallel x)$ and $P_{\alpha_a(t)} = \frac{1}{n+N+1}\alpha_a(t)$. And by definition $\mathrm{KL}(P_{\alpha_a(t)} \parallel P^*_{\mu_\star-\epsilon_1}) = \mathcal{K}_{\text{inf}}(P_{\alpha_a(t)}, \mu_\star - \epsilon_1)$, therefore

$$\begin{aligned}&\mathbf{Pr}\left(\overline{\theta}_k(t) \geq \mu_\star - \epsilon_1 \middle| \alpha_a(t), T_k(t) = m\right)\\ &\leq C(m+N+1)^{N/2} \exp\{-(m+N+1)\mathcal{K}_{\text{inf}}(P_{\alpha_a(t)}, \mu_\star - \epsilon_1)\},\end{aligned}$$

where $C = \frac{\exp\{1/12\}}{\Gamma(N+1)}\left(\frac{1}{\sqrt{2\pi}}\right)^N$. On the other hand, $\mathcal{K}_{\text{inf}}(x, \mu_\star - \epsilon_1)$ is continuous in $x \in [0,1]^{N+1}$ on the probability simplex with respect to the L^∞ distance from ((19), Theorem 7) and Lemma 18 in Appendix H (29). Therefore, for any $\epsilon_3 > 0$, there exists $\epsilon_2 > 0$ and constant $C' > 0$ such that

$$\begin{aligned}&\mathbf{Pr}\left(\overline{\theta}_k(t) \geq \mu_\star - \epsilon_1 \middle| \parallel \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \parallel_\infty \leq \epsilon_2, T_k(t) = m\right)\\ &\leq C' \exp\{-(m+N+1)(\mathcal{K}_{\text{inf}}(p_a, \mu_\star - \epsilon_1) - \epsilon_3)\}\end{aligned}$$

And because $\mathbf{Pr}\left(\parallel \frac{\alpha_a(t)}{T_k(t)+N+1} - p_a(t) \parallel_\infty \leq \epsilon_2, T_k(t) = m\right) \leq 1$. Therefore,

$$\begin{aligned}A1 &\leq m_0 + C_1' \sum_{t=1}^{n} \exp\{-(m+N+1)(\mathcal{K}_{\text{inf}}(p_a, \mu_\star - \epsilon_1) - \epsilon_3)\}\\ &\leq m_0 + C_1' T \exp\{-(m+N+1)(\mathcal{K}_{\text{inf}}(p_a, \mu_\star - \epsilon_1) - \epsilon_3)\}\end{aligned} \tag{5}$$

Choosing $m_0 = \frac{\log n}{\mathcal{K}_{\text{inf}}(p_a, \mu_\star-\epsilon_1)-\epsilon_3} - N - 1$, we have

$$A1 \leq \frac{\log n}{\mathcal{K}_{\text{inf}}(p_a, \mu_\star - \epsilon_1) - \epsilon_3} - N - 1 + C_1'$$

Furthermore, as from ((19), Theorem 7), it is proven that $\mu \to \mathcal{K}_{\text{inf}}(F, \mu)$ is continuous for $\mu < 1$, when we scale reward from [0,1] to $[0, R]$ therefore μ from [0,1] to $[0, R]$. We have $\mu \to \mathcal{K}_{\text{inf}}(F, \mu)$ is continuous for $\mu < R$. Therefore, $\forall \epsilon_4 > 0, \exists \epsilon_1 > 0$, such that

$$\begin{aligned}&|\mathcal{K}_{\text{inf}}(p_a, \mu^* - \epsilon_1) - \mathcal{K}_{\text{inf}}(p_a, \mu^*)| \leq \epsilon_4\\ \Rightarrow\ &\mathcal{K}_{\text{inf}}(p_a, \mu^* - \epsilon_1) - \epsilon_3 \geq \mathcal{K}_{\text{inf}}(p_a, \mu^*) - \epsilon_3 - \epsilon_4\end{aligned}$$

Therefore, $\forall \epsilon_0 > 0$

$$A1 \leq \frac{(\epsilon_0 + 1)\log n}{\mathcal{K}_{\text{inf}}(p_a, \mu_\star)} - N - 1 + C_1'$$

Also According to Proposition 8 (29), for any $\epsilon_0 > 0$ we have

$$A2 \leq O(1) \tag{6}$$

Combining inequality (5) and inequality (6) leads us to

$$\mathbb{E}[T_a(n)] \leq \frac{(1+\epsilon_0)\log n}{\mathcal{K}^{(N)}_{\text{inf}}(F_a, \mu_\star)} + o(\log n) + O(1).$$

Therefore which concludes the proof. □

Lemma 6. *Consider Particle Thompson Sampling(PATS) strategy applied to a non-stationary problem where the pay-off sequence satisfies Assumption 1. Then for any $\epsilon_0 \geq 0$. Let $T_a(n)$ denote the number of plays of arm a up to timestep n. Then if a is the index of a suboptimal arm, then each sub-optimal arm a is played in expectation at most*

$$\mathbb{E}[T_a(n)] \leq \frac{\log n}{\mathcal{K}_{\textit{inf}}(F_a, \mu_\star) - \epsilon_0} + o(\log n) + O(1).$$

Proof. In this Theorem, we use the Levy distance. Recall that the Levy distance between two cumulative distribution functions F and G on $[0,1]$ is defined as

$$D_L(F,G) = \inf\{\epsilon > 0 : \forall x \in [0,1], F(x-\epsilon) - \epsilon \le G(x) \le F(x+\epsilon) + \epsilon\}.$$

The proof follows the same steps as in Lemma 5. We also can derive

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}(I(t)=a)\right] =& \underbrace{\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}(I(t)=a), \overline{\phi}_{a,t} \ge \mu_* - \epsilon_1, D_L(\widehat{F}_{I(t)}, F_{I(t)}) \le \epsilon_2)\right]}_{B1} \\ &+ \underbrace{\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}(I(t)=a), \overline{\phi}_{a,t} < \mu_* - \epsilon_1, D_L(\widehat{F}_{I(t)}, F_{I(t)}) > \epsilon_2)\right]}_{B2}\end{aligned}$$

We can use the same ways of derivations as in Lemma 5, equation (4) to have the same bound

$$\begin{aligned}B1 \le m_0 + \sum_{t=1}^{n}\sum_{m=m_0}^{n} \mathbf{Pr}\left(\overline{\theta}_k(t) \ge \mu_\star - \epsilon_1 \middle| D_L\left(\widehat{F}_a(t), F_a(t)\right) \le \epsilon_2, T_k(t) = m\right) \\ \times \mathbf{Pr}\left(D_L\left(\widehat{F}_a(t), F_a(t)\right) \le \epsilon_2, T_k(t) = m\right) \qquad (7)\end{aligned}$$

According to Lemma 15 in Appendix G.1 (29) on conditional probabilities, for any $\nu \in (0,1)$ we have

$$\begin{aligned}&\mathbf{Pr}\left(\overline{\theta}_k(t) \ge \mu_\star - \epsilon_1 \middle| D_L\left(\widehat{F}_a(t), F_a(t)\right) \le \epsilon_2, T_k(t) = m\right) \\ &\le \frac{1}{\nu}\exp\left\{-n\left(\mathcal{K}_{\text{inf}}(\widehat{F}_a(t), \mu_\star - \epsilon_1) - \nu\frac{\mu_\star - \epsilon_1}{1-(\mu_\star - \epsilon_1)}\right)\right\}\end{aligned}$$

Because $\mathcal{K}_{\text{inf}}(F,\mu)$ is continuous in F with respect to the Levy distance from (19), Theorem 7, for any $\epsilon_3 > 0$ there exists $\epsilon_2 > 0$ such that

$$D_L(\widehat{F}_a(t), F_a) \le \epsilon_2 \Rightarrow \left|\mathcal{K}_{\text{inf}}(\widehat{F}_a(t), \mu_\star - \epsilon_1) - \mathcal{K}_{\text{inf}}(F_a, \mu_\star - \epsilon_1)\right| \le \epsilon_3$$

Therefore, $\forall\nu \in (0,1)$ and for any $\epsilon_5 > 0$, there exists $\epsilon_1, \epsilon_2 > 0$ such that

$$\begin{aligned}&\mathbf{Pr}\left(\overline{\theta}_k(t) \ge \mu_\star - \epsilon_1 \middle| D_L\left(\widehat{F}_a(t), F_a(t)\right) \le \epsilon_2, T_k(t) = m\right) \\ &\le \frac{1}{\nu}\left(-m\left(\mathcal{K}_{\text{inf}}(F_a, \mu_\star - \epsilon_1) - \epsilon_3 - \nu\frac{\mu_\star - \epsilon_1}{1-(\mu_\star - \epsilon_1)}\right)\right) \\ &\overset{\text{(Theorem 6 (19))}}{\le} \frac{1}{\nu}\left(-m\left(\mathcal{K}_{\text{inf}}(F_a, \mu_\star)\frac{\epsilon_1}{1-\mu_\star} - \epsilon_3 - \nu\frac{\mu_\star - \epsilon_1}{1-(\mu_\star - \epsilon_1)}\right)\right)\end{aligned}$$

This implies that $\forall\epsilon_0 > 0$, there exists $\nu \in (0,1), \epsilon_1 > 0$ and $\epsilon_2 > 0$ such that

$$\mathbf{Pr}\left(\overline{\theta}_k(t) \ge \mu_\star - \epsilon_1 \middle| D_L\left(\widehat{F}_a(t), F_a(t)\right) \le \epsilon_2, T_k(t) = m\right) \le \frac{1}{\nu}\exp\left\{-m(\mathcal{K}_{\text{inf}}(F_a, \mu_\star) - \epsilon_0)\right\}$$

Therefore, according to inequality (7) and the fact that

$$\mathbf{Pr}\left(D_L\left(\widehat{F}_a(t), F_a(t)\right) \le \epsilon_2, T_k(t) = m\right) \le 1$$

we have

$$\begin{aligned}B1 &\le m_0 + \sum_{t=1}^{n} \frac{1}{\nu}\exp\left\{-m(\mathcal{K}_{\text{inf}}(F_a, \mu_\star) - \epsilon_0)\right\} \\ &\le m_0 + \frac{1}{\nu} T \exp\left\{-m_0(\mathcal{K}_{\text{inf}}(F_a, \mu_\star) - \epsilon_0)\right\}\end{aligned}$$

Choose $m_0 = \frac{\log n}{\mathcal{K}_{\text{inf}}(F_a,\mu_\star)-\epsilon_0}$ we have

$$B1 \leq \frac{\log n}{\mathcal{K}_{\text{inf}}(F_a,\mu_\star)-\epsilon_0} + \frac{1}{\nu}$$

Also According to Proposition 10 (29), for any $\epsilon_0 > 0$ we have

$$B2 \leq O(1)$$

That leads us to

$$\mathbb{E}[T_a(n)] \leq \frac{\log n}{\mathcal{K}_{\text{inf}}(F_a,\mu_\star)-\epsilon_0} + o(\log n) + O(1),$$

which concludes the proof. □

Lemma 7. *Consider Categorical Thompson Sampling(CATS) strategy applied to a non-stationary problem where the pay-off sequence satisfies Assumption 1. Let us define the power mean estimator* $\widehat{\mu}_n(p)$ *as* $\widehat{\mu}_n(p) = \left(\sum_{a=1}^{K} \frac{T_a(n)}{n}\widehat{\mu}^p_{a,T_a(n)}\right)^{\frac{1}{p}}$*, and* $\delta_{\star,n} = \mu_\star - \mu_{\star,n}$ *For any* $p \geq 1, \epsilon_0 > 0$*, we have*

$$|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_\star| \leq |\delta_{\star,n}| + \frac{R}{n}\sum_{a=1,a\neq a_*}^{K}\left\{\frac{(1+\epsilon_0)\log n}{\mathcal{K}^{(N)}(F_a,\mu^\star)} + o(\log n) + O(1)\right\}$$

Proof. We observe that

$$|\widehat{\mu}_n(p) - \mu_\star| \leq |\widehat{\mu}_n(p) - \mu_{\star,n}| + |\mu_\star - \mu_{\star,n}| = |\widehat{\mu}_n(p) - \mu_{\star,n}| + |\delta_{\star,n}|$$

Furthermore,

$$\widehat{\mu}_{a,T_a(n)} \leq \mu_{a,n} + \left|\widehat{\mu}_{a,T_a(n)} - \mu_{a,n}\right|. \tag{8}$$

Since $\mu_{\star,n} = \max_{a\in[K]}\{\mu_{a,n}\}$, we have

$$\widehat{\mu}_n(p) - \mu_{\star,n} = \widehat{\mu}_n(p) - \sum_{a=1}^{K} T_a(n)\mu_{\star,n} \leq \left(\sum_{a=1}^{K}\frac{T_a(n)}{n}\left(\widehat{\mu}_{a,T_a(n)}\right)^p\right)^{\frac{1}{p}} - \left(\sum_{a=1}^{K}\frac{T_a(n)}{n}\left(\mu_{a,n}\right)^p\right)^{\frac{1}{p}}$$

$$= \frac{\left(\sum_{a=1}^{K} T_a(n)\left(\widehat{\mu}_{a,T_a(n)}\right)^p\right)^{\frac{1}{p}} - \left(\sum_{a=1}^{K} T_a(n)\left(\mu_{a,n}\right)^p\right)^{\frac{1}{p}}}{n^{\frac{1}{p}}}$$

Applying Minkowski's inequality from Lemma 3, and the result of (8), we have

$$\widehat{\mu}_n(p) - \mu_{\star,n} \leq \frac{\left(\sum_{a=1}^{K} T_a(n)\left(\mu_a + \left|\widehat{\mu}_{a,T_a(n)} - \mu_{a,n}\right|\right)^p\right)^{\frac{1}{p}} - \left(\sum_{a=1}^{K} T_a(n)\left(\mu_{a,n}\right)^p\right)^{\frac{1}{p}}}{n^{\frac{1}{p}}}$$

$$\leq \frac{\left(\sum_{a=1}^{K} T_a(n)\left(\left|\widehat{\mu}_{a,T_a(n)} - \mu_{a,n}\right|\right)^p\right)^{\frac{1}{p}}}{n^{\frac{1}{p}}}$$

On the other hand,

$$\mu_{\star,n} - \widehat{\mu}_n(p) = \frac{n\mu_{\star,n} - n\widehat{\mu}_n(p)}{n} = \frac{n\mu_{\star,n} - \left(\sum_{a=1}^{K} T_a(n)\mu_{a,n}\right) + \sum_{a=1}^{K} T_a(n)\mu_{a,n} - n\widehat{\mu}_n(p)}{n}$$

$$= \frac{\sum_{a=1,a\neq a_*}^{K} T_a(n)\left|\mu_{\star,n} - \mu_{a,n}\right| + \sum_{a=1}^{K} T_a(n)\mu_{a,n} - n\widehat{\mu}_n(p)}{n}$$

$$\leq R\sum_{a=1,a\neq a_*}^{K}\frac{T_a(n)}{n} + \sum_{a=1}^{K}\frac{T_a(n)}{n}\mu_{a,n} - \widehat{\mu}_n(p) \tag{9}$$

Because power mean is an increasing function of p, so that

$$\sum_{a=1}^{K} \frac{T_a(n)}{n} \mu_{a,n} \leq \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \left(\mu_{a,n} \right)^p \right)^{1/p}.$$

Furthermore, we observe that

$$\mu_{a,n} \leq \widehat{\mu}_{a,T_a(n)} + \left| \widehat{\mu}_{a,T_a(n)} - \mu_{a,n} \right|.$$

So that, from equation (9) we have

$$\begin{aligned}
\mu_{\star,n} - \widehat{\mu}_n(p) &\leq R \sum_{a=1, a\neq a_*}^{K} \frac{T_a(n)}{n} + \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \left(\mu_{a,n} \right)^p \right)^{1/p} - \widehat{\mu}_n(p) \\
&\leq R \sum_{a=1, a\neq a_*}^{K} \frac{T_a(n)}{n} \\
&+ \frac{\left(\sum_{a=1}^{K} T_a(n) \left(\widehat{\mu}_{a,T_a(n)} + \left| \widehat{\mu}_{a,T_a(n)} - \mu_{a,n} \right| \right)^p \right)^{\frac{1}{p}} - \left(\sum_{a=1}^{K} T_a(n) \left(\widehat{\mu}_{a,T_a(n)} \right)^p \right)^{\frac{1}{p}}}{n^{\frac{1}{p}}} \\
&\overset{\text{(Minkovski's inequality)}}{\leq} R \sum_{a=1, a\neq a_*}^{K} \frac{T_a(n)}{n} + \frac{\left(\sum_{a=1}^{K} T_a(n) \left(\left| \widehat{\mu}_{a,T_a(n)} - \mu_{a,n} \right| \right)^p \right)^{\frac{1}{p}}}{n^{\frac{1}{p}}} \\
&\overset{\text{(Properties of } L^p \text{ norm)}}{\leq} R \sum_{a=1, a\neq a_*}^{K} \frac{T_a(n)}{n} + \frac{\left(\sum_{a=1}^{K} T_a(n) \left(\left| \widehat{\mu}_{a,T_a(n)} - \mu_{a,n} \right| \right) \right)}{n^{\frac{1}{p}}} \\
&= R \sum_{a=1, a\neq a_*}^{K} \frac{T_a(n)}{n} + \frac{\sum_{a=1}^{K} \left(\left| \sum_{t}^{T_a(n)} R_{a,t} - T_a(n)\mu_{a,n} \right| \right)}{n^{\frac{1}{p}}}
\end{aligned}$$

Therefore

$$\begin{aligned}
\left| \mathbb{E}[\widehat{\mu}_n(p) - \mu_{\star,n}] \right| &\leq R \sum_{a=1, a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n} + \frac{\mathbb{E}\left[\left(\left| \sum_{a=1}^{K} \sum_{t}^{T_a(n)} R_{a,t} - T_a(n)\mu_{a,n} \right| \right) \right]}{n^{\frac{1}{p}}} \\
&= R \sum_{a=1, a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n}
\end{aligned}$$

Please note that because we study non-stationary bandits, $\mathbb{E}[\sum_t^n R_{a,t}] = n\mu_{a,n}$, therefore,

$$\frac{\mathbb{E}\left[\left(\left| \sum_{a=1}^{K} \sum_{t}^{T_a(n)} R_{a,t} - T_a(n)\mu_{a,n} \right| \right) \right]}{n^{\frac{1}{p}}} = 0$$

According to Lemma 5, we have

$$\left| \mathbb{E}[\widehat{\mu}_n(p) - \mu_{\star,n}] \right| \leq R \sum_{a=1, a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n} \leq \frac{R}{n} \sum_{a=1, a\neq a_*}^{K} \left\{ \frac{(1+\epsilon_0)\log n}{\mathcal{K}^{(N)}(F_a, \mu^\star)} + o(\log n) + O(1) \right\},$$

which concludes the proof. □

Lemma 8. *Consider Particle Thompson Sampling(PATS) strategy applied to a non-stationary problem where the pay-off sequence satisfies Assumption 1. Let us define the power mean estimator* $\widehat{\mu}_n(p)$ *as* $\widehat{\mu}_n(p) = \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \widehat{\mu}_{a,T_a(n)}^p \right)^{\frac{1}{p}}$*, and* $\delta_{\star,n} = \mu_\star - \mu_{\star,n}$ *For any* $p \geq 1, \epsilon_0 > 0$*, we have*

$$\left| \mathbb{E}[\widehat{\mu}_n(p)] - \mu_\star \right| \leq |\delta_{\star,n}| + \frac{R}{n} \sum_{a=1, a\neq a_*}^{K} \left\{ \frac{\log n}{\mathcal{K}_{inf}(F_a, \mu^\star) - \epsilon_0} + o(\log n) + O(1) \right\}$$

Proof. Similar to Lemma 7, we can derive

$$|\mathbb{E}[\widehat{\mu}_n(p) - \mu_{\star,n}]| \leq |\delta_{\star,n}| + R \sum_{a=1,a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n}.$$

And according to Lemma 6, we have

$$|\mathbb{E}[\widehat{\mu}_n(p) - \mu_{\star,n}]| \leq R \sum_{a=1,a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n} \leq \frac{R}{n} \sum_{a=1,a\neq a_*}^{K} \left\{ \frac{\log n}{\mathcal{K}_{\inf}(F_a, \mu^\star) - \epsilon_0} + o(\log n) + O(1) \right\},$$

which concludes the proof. □

Theorem 1. *For $a \in [K]$, let $(\widehat{\mu}_{a,n})_{n\geq 1}$ be a sequence of estimator satisfying $\underset{n\to\infty}{plim}\, \widehat{\mu}_{a,n} = \mu_a$ and let $\mu_\star = \max_a\{\mu_a\}$. Assume that all the estimators are bounded in $[0, R]$. We consider a bandit algorithm that selects each arm according to CATS once in each round $n \geq K$.*

Then, for all $p \in [1, \infty)$, the sequence of estimators

$$\widehat{\mu}_n(p) = \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \widehat{\mu}^p_{a,T_a(n)} \right)^{\frac{1}{p}},$$

where $T_a(n) = \sum_{t=1}^{n-1} \mathbb{1}(a_t = a)$ is the number of selections of a prior to round n satisfies

$$\underset{n\to\infty}{plim}\, \widehat{\mu}_n(p) = \mu_\star.$$

Proof. We first prove that $\lim_{n\to\infty} \mathbb{E}[\widehat{\mu}_n(p)] = \mu_*$. According to the result of Lemma 7, we have

$$\begin{aligned}
|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_\star| &\leq |\delta_{\star,n}| + R \sum_{a=1,a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n} \\
&\leq |\delta_{\star,n}| + \frac{R}{n} \sum_{a=1,a\neq a_*}^{K} \left\{ \frac{(1+\epsilon_0)\log n}{\mathcal{K}^{(N)}(F_a, \mu^\star)} + o(\log n) + O(1) \right\}
\end{aligned}$$

with $\delta_{\star,n} = \mu_\star - \mu_{\star,n}$, and because $\lim_{n\to\infty} \mu_{*,n} = \mu_\star$, we can concludes that

$$\lim_{n\to\infty} \mathbb{E}[\widehat{\mu}_n(p)] = \mu_*.$$

Second, we prove that

$$\forall n \geq 1, \forall \varepsilon > 0, \exists c > 0 \text{ that } \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \leq cn^{-1}\varepsilon^{-1}.$$

We observe that

$$\begin{aligned}
&|\widehat{\mu}_n(p) - \mu_\star| \leq |\widehat{\mu}_n(p) - \mu_{\star,n}| + |\mu_\star - \mu_{\star,n}| = |\widehat{\mu}_n(p) - \mu_{\star,n}| + |\delta_{\star,n}| \\
\Longrightarrow &\mathbb{P}(|\widehat{\mu}_n(p) - \mu_\star| \geq \epsilon) \leq \mathbb{P}(|\widehat{\mu}_n(p) - \mu_{\star,n}| \geq \epsilon/2) + \mathbb{P}(|\delta_{\star,n}| \geq \epsilon/2).
\end{aligned}$$

Because $\lim_{n\to n} |\delta_{\star,n}| = 0$, therefore, $\exists N_0 > 0$ such that $\forall n \geq N_0$, we have $|\delta_{\star,n}| < \epsilon/2$ that means

$$\forall n > N_0, \mathbb{P}(|\delta_{\star,n}| \geq \epsilon/2) = 0.$$

Next, according to Lemma 7,

$$|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_{\star,n}| \leq \frac{R}{n} \sum_{a=1,a\neq a_*}^{K} \left\{ \frac{(1+\epsilon_0)\log n}{\mathcal{K}^{(N)}(F_a, \mu^\star)} + o(\log n) + O(1) \right\} = O(n^{-1}),$$

that leads to

$$\mathbb{P}(|\widehat{\mu}_n(p) - \mu_{\star,n}| \geq \epsilon/2) \leq \frac{|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_{\star,n}|}{\epsilon/2} = \frac{O(n^{-1})}{\epsilon/2}.$$

Therefore, $\exists c > 0$ such that

$$\mathbb{P}(|\widehat{\mu}_n(p) - \mu_{\star,n}| \geq \epsilon/2) \leq cn^{-1}\epsilon^{-1},$$

which means

$$\forall n \geq N_0, \forall \varepsilon > 0, \exists c > 0 \text{ that } \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \leq cn^{-1}\varepsilon^{-1}.$$

Now we see that $|\widehat{\mu}_n(p) - \mu_\star| \leq R$. With $\epsilon \geq R$, we have $|\widehat{\mu}_n(p) - \mu_\star| > \epsilon \Leftrightarrow |\widehat{\mu}_n(p) - \mu_\star| > R$, therefore the inequality holds as

$$\mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) = 0 \leq cn^{-1}\varepsilon^{-1}.$$

with $0 < \epsilon < R, 1 \leq n < N_0 \Rightarrow n\epsilon < RN_0 \Rightarrow n^{-1}\varepsilon^{-1} > 1/RN_0$. Therefore

$$\forall C > 1/RN_0 \Rightarrow \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \leq 1 < Cn^{-1}\varepsilon^{-1},$$

which means

$$\forall n \geq 1, \forall \varepsilon > 0, \exists C > 0 \text{ that } \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \leq Cn^{-1}\varepsilon^{-1}.$$

That concludes the proof. $\square$

Theorem 2. *For $a \in [K]$, let $(\widehat{\mu}_{a,n})_{n\geq 1}$ be a sequence of estimator satisfying $\underset{n\to\infty}{plim}\,\widehat{\mu}_{a,n} = \mu_a$ and let $\mu_\star = \max_a\{\mu_a\}$. Assume that all the estimators are bounded in $[0, R]$. We consider a bandit algorithm that selects each arm according to PATS once in each round $n \geq K$.*

Then, for all $p \in [1, \infty)$, the sequence of estimators

$$\widehat{\mu}_n(p) = \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \widehat{\mu}^p_{a,T_a(n)}\right)^{\frac{1}{p}},$$

where $T_a(n) = \sum_{t=1}^{n-1} \mathbb{1}(a_t = a)$ is the number of selections of a prior to round n satisfies

$$\underset{n\to\infty}{plim}\,\widehat{\mu}_n(p) = \mu_\star.$$

Proof. The proof follows the same steps as Theorem 1. We first prove that $\lim_{n\to\infty} \mathbb{E}[\widehat{\mu}_n(p)] = \mu_*$. According to the result of Lemma 8, we have

$$\begin{aligned}
|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_\star| &\leq |\delta_{\star,n}| + R \sum_{a=1, a\neq a_*}^{K} \frac{\mathbb{E}[T_a(n)]}{n} \\
&\leq |\delta_{\star,n}| + \frac{R}{n} \sum_{a=1, a\neq a_*}^{K} \left\{ \frac{\log n}{\mathcal{K}_{\text{inf}}(F_a, \mu^\star) - \epsilon_0} + o(\log n) + O(1) \right\}
\end{aligned}$$

with $\delta_{\star,n} = \mu_\star - \mu_{\star,n}$, and because $\lim_{n\to\infty} \mu_{*,n} = \mu_\star$, we can concludes that

$$\lim_{n\to\infty} \mathbb{E}[\widehat{\mu}_n(p)] = \mu_*.$$

Second, we prove that

$$\forall n \geq 1, \forall \varepsilon > 0, \exists c > 0 \text{ that } \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \leq cn^{-1}\varepsilon^{-1}.$$

We observe that

$$\begin{aligned}
&|\widehat{\mu}_n(p) - \mu_\star| \leq |\widehat{\mu}_n(p) - \mu_{\star,n}| + |\mu_\star - \mu_{\star,n}| = |\widehat{\mu}_n(p) - \mu_{\star,n}| + |\delta_{\star,n}| \\
\Longrightarrow &\mathbb{P}(|\widehat{\mu}_n(p) - \mu_\star| \geq \epsilon) \leq \mathbb{P}(|\widehat{\mu}_n(p) - \mu_{\star,n}| \geq \epsilon/2) + \mathbb{P}(|\delta_{\star,n}| \geq \epsilon/2).
\end{aligned}$$

Because $\lim_{n\to n} |\delta_{\star,n}| = 0$, therefore, $\exists N_0 > 0$ such that $\forall n \geq N_0$, we have $|\delta_{\star,n}| < \epsilon/2$ that means

$$\forall n > N_0, \mathbb{P}(|\delta_{\star,n}| \geq \epsilon/2) = 0.$$

Next, according to Lemma 8,

$$|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_{\star,n}| \le \frac{R}{n} \sum_{a=1, a\neq a_*}^{K} \left\{ \frac{\log n}{\mathcal{K}_{\text{inf}}(F_a, \mu^\star) - \epsilon_0} + o(\log n) + O(1) \right\} = O(n^{-1}),$$

that leads to

$$\mathbb{P}(|\widehat{\mu}_n(p) - \mu_{\star,n}| \ge \epsilon/2) \le \frac{|\mathbb{E}[\widehat{\mu}_n(p)] - \mu_{\star,n}|}{\epsilon/2} = \frac{O(n^{-1})}{\epsilon/2}.$$

Therefore, $\exists c > 0$ such that

$$\mathbb{P}(|\widehat{\mu}_n(p) - \mu_{\star,n}| \ge \epsilon/2) \le cn^{-1}\epsilon^{-1},$$

which means

$$\forall n \ge N_0, \forall \varepsilon > 0, \exists c > 0 \text{ that } \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \le cn^{-1}\varepsilon^{-1}.$$

Now we see that $|\widehat{\mu}_n(p) - \mu_\star| \le R$. With $\epsilon \ge R$, we have $|\widehat{\mu}_n(p) - \mu_\star| > \epsilon \Leftrightarrow |\widehat{\mu}_n(p) - \mu_\star| > R$, therefore the inequality holds as

$$\mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) = 0 \le cn^{-1}\varepsilon^{-1}.$$

with $0 < \epsilon < R, 1 \le n < N_0 \Rightarrow n\epsilon < RN_0 \Rightarrow n^{-1}\varepsilon^{-1} > 1/RN_0$. Therefore

$$\forall C > 1/RN_0 \Rightarrow \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \le 1 < Cn^{-1}\varepsilon^{-1},$$

which means

$$\forall n \ge 1, \forall \varepsilon > 0, \exists C > 0 \text{ that } \mathbb{P}\left(|\widehat{\mu}_n(p) - \mu_\star| > \varepsilon\right) \le Cn^{-1}\varepsilon^{-1}.$$

That concludes the proof. □

E Convergence of CATS and PATS in Monte-Carlo Tree Search

Based upon the results of CATS and PATS using power mean as the value backup operator on the described non-stationary multi-armed bandit problem, we derive theoretical results for CATS in an MCTS tree.

We derive Theorem 3 for CATS and Theorem 4 for PATS, which show concentration and convergence for any internal node in the tree. These proofs utilize induction, leveraging the results of Lemma 7 for CATS and Lemma 8 for PATS, and Lemma 5 for CATS and Lemma 6 for PATS. Additionally, we use Lemma 1, which demonstrates the concentration and convergence of an estimated Q-value based on the child V-value node, applying it recursively throughout the tree.

Our main results, Theorem 5 for CATS and Theorem 5 for PATS, show that the simple regret converges non-asymptotically at a rate of $O(n^{-1})$.

Theorem 3. *When we apply the CATS algorithm, we have*

(i) For any node s_h at the depth h^{th} in the tree,

$$\underset{n\to\infty}{plim}\, \widehat{Q}_n(s_h, a_k) = \widetilde{Q}(s_h, a_k).$$

(ii) For any node s_h at the depth h^{th} in the tree,

$$\underset{n\to\infty}{plim}\, \widehat{V}_n(s_h) = \widetilde{V}(s_h).$$

Proof. We will prove this by induction on the depth D of the tree. If the tree only has depth (1). The state at the root node is s_0, let us assume that at time step t, after taking action a_k, the MCTS tree gets an intermediate reward $r_t(s_0, a_k)$ and traverses to the next state s_1. Let us assume that $R(s_0, a_k)$ is the mean of the intermediate reward at state s_0, after taking action a_k. We recall the definition of $\widetilde{Q}(s_0, a_k)$, with π_0 is the rollout policy to estimate the newly added node at the leaf,

$$\widetilde{Q}(s_0, a_k) = R(s_0, a_k) + \gamma \sum_{s_1 \in \mathcal{A}_{s_0}} \mathbb{P}(s_1|s_0, a_k)\widetilde{V}(s_1)$$

where $\widetilde{V}(s_1)$ is the value of the policy π_0 at state s_1, $\mathcal{A}_{s_0}$ is the set of feasible actions at state s_0, $|\mathcal{A}_{s_0}| = M$, $\mathbb{P}(s_1|s_0, a_k)$ is the probability transition of taking action a_k at state s_0 to state s_1. From ((1)), we have

$$\widehat{Q}_n(s_0, a_k) = \frac{1}{n}\sum_{t=1}^{n} r_t(s_0, a_k) + \gamma \sum_{s_1 \sim \tau(s_0, a_k)} \frac{T^{s_1}_{s_0,a_k}(n)}{n} \widehat{V}_{T^{s_1}_{s_0,a_k}(n)}(s_1)$$

(i) is a direct result of Lemma 1 with X_t is the intermediate reward $r_t(s_0, a_k)$ at time t, $p = (p_1, p_2, ...p_M) \sim \mathbb{P}(\cdot|s_0, a_k)$, where $\mathbb{P}(\cdot|s_0, a_k)$ is the probability transition dynamic of taking action a_k at state s_0. For $m \in [M]$, each $(\widehat{V}_{m,t})_{t\geq 1}$ at time step t is the deterministic initial Value function $\widetilde{V}(s_1)$. We have

$$\operatorname*{plim}_{n\to\infty} \widehat{V}_{m,n}(s_1) = \widetilde{V}(s_1), \text{ with } s_1 \in \{s_m\}, m = 1, 2, 3...M, \text{ where } s_m \sim \tau(\cdot|s_0, a_k)$$

(ii) Direct results from Theorem 1. In detail, we have from (i),

$$\operatorname*{plim}_{n\to\infty} \widehat{Q}_n(s_0, a_k) = \widetilde{Q}(s_0, a_k), \text{ with } a_k \in \mathcal{A}_{s_0}$$

Because by definition:

$$\widetilde{V}(s_0) = \max_{a_k \in \mathcal{A}_{s_0}} \widetilde{Q}(s_0, a_k)$$

$$\widehat{V}_n(s_0) = \left(\sum_{a \in \mathcal{A}_{s_0}} \frac{T_{s_0,a}(n)}{n} \left(\widehat{Q}_{T_{s_0,a}(n)}(s_0, a)\right)^p\right)^{\frac{1}{p}} \text{ for some } p \in [1, +\infty)$$

Then we have

$$\operatorname*{plim}_{n\to\infty} \widehat{V}_n(s_0) = \widetilde{V}(s_0)$$

that concludes for (ii)

Let us assume that with the tree of depth D, the theorem holds for all its children.

Now let's consider the tree with depth $(D + 1)$. When we take one action at the root node at the state s_0, it comes to a subtree with depth (D). According to the induction assumption, the results hold for any internal node in the tree after we take the first action. We have $s_1 \sim \tau(s_0, a_k)$. By the definition, $\widetilde{V}(s_H) = V_0(s_H)$ and, for all $h \leq H - 1$,

$$\begin{aligned}
\widetilde{Q}(s_h, a) &= R(s_h, a) + \gamma \sum_{s_{h+1} \in \mathcal{A}_s} \mathbb{P}(s_{h+1}|s_h, a)\widetilde{V}(s_{h+1}) \\
\widetilde{V}(s_h) &= \max_a \widetilde{Q}(s_h, a)
\end{aligned}$$

By the assumption of the induction the root node of a subtree with depth (D) at state s_1 we have

$$\operatorname*{plim}_{n\to\infty} \widehat{V}_n(s_1) = \widetilde{V}(s_1)$$

(i) Let's apply Lemma 1 with $\{X_t\}$ is the intermediate reward $\{r_t(s_0, a_k)\}$, $p = (p_1, p_2, ...p_M) \sim \mathbb{P}(\cdot|s_0, a_k)$. For $m \in [M]$, each $(\widehat{V}_{m,t})_{t\geq 1}$ at time step t is the empirical Value function $\widehat{V}_t(s_1)$. We will have

$$\operatorname*{plim}_{n\to\infty} \widehat{Q}_n(s_0, a_k) = \widetilde{Q}(s_0, a_k), \text{ with } a_k \in \mathcal{A}_{s_0}$$

(ii) follows the results of Theorem 1 as at the root node s_0 of depth $D + 1$, with

$$\widetilde{V}(s_0) = \max_{a_k \in \mathcal{A}_{s_0}} \widetilde{Q}(s_0, a_k)$$

$$\widehat{V}_n(s_0) = \left(\sum_{a \in \mathcal{A}_s} \frac{T_{s_0,a}(n)}{n} \left(\widehat{Q}_{T_{s_0,a}(n)}(s_0, a)\right)^p\right)^{\frac{1}{p}} \text{ for some } p \in [1, +\infty)$$

And because

$$\operatorname*{plim}_{n\to\infty} \widehat{Q}_n(s_0, a_k) = \widetilde{Q}(s_0, a_k), \text{ with } a_k \in \mathcal{A}_{s_0}$$

Then, we have

$$\operatorname*{plim}_{n\to\infty} \widehat{V}_n(s_0) = \widetilde{V}(s_0).$$

that concludes for (ii)

The results of Theorem 3 hold for any node in the tree with the tree of depth $(D+1)$. By induction, we can conclude the proof. □

Similarly we can derive the following Theorem

Theorem 4. *When we apply the PATS algorithm, we have*

(i) For any node s_h at the depth h^{th} in the tree,

$$\operatorname*{\mathit{plim}}_{n\to\infty} \widehat{Q}_n(s_h, a_k) = \widetilde{Q}(s_h, a_k).$$

(ii) For any node s_h at the depth h^{th} in the tree,

$$\operatorname*{\mathit{plim}}_{n\to\infty} \widehat{V}_n(s_h) = \widetilde{V}(s_h).$$

Proof. The proof follows the same steps as Theorem 3 by applying the results of Lemma 1 and Theorem 2. □

Theorem 5. ***(Convergence of Expected Payoff of CATS)*** *We have at the root node s_0,*

$$\mathbb{E}\left[\left|\widehat{V}_n\left(s_0\right) - V^\star(s_0)\right|\right] \le O(n^{-1}).$$

Proof. We prove the result by induction and use the results of Theorem 3 to prove this Theorem. Let us assume that the depth of the tree is $D = 1$, as the results of Lemma 7, we have

$$\left|\mathbb{E}[\widehat{V}_n(s_0)] - V^\star(s_0)\right| \le |\delta_{\star,n}| + O(\frac{\log n}{n}) = |\delta_{\star,n}| + O(n^{-1}).$$

And because the tree only have the depth $D = 1$, we have $|\delta_{\star,n}| = 0$, so that the result holds at the depth $D = 1$. Let us assume that we have the result of the tree at the depth D. Now when the depth of the tree is $D + 1$, at the root node s_0, the conditions of Assumption 1 hold as the results of Theorem 3 then we have

$$\left|\mathbb{E}[\widehat{V}_n(s_0)] - V^\star(s_0)\right| \overset{\text{(Lemma 7)}}{\le} |\delta_{\star,n}| + O(\frac{\log n}{n}) = |\delta_{\star,n}| + O(n^{-1}),$$

where the bias

$$|\delta_{\star,n}| = \left|\mathbb{E}[\widehat{Q}_n(s_0, a_\star)] - Q^\star(s_0, a_\star)\right| \overset{\text{(contraction)}}{\le} \gamma \parallel \mathbb{E}[\widehat{V}_n^{(1)}] - V^\star \parallel_\infty \overset{\text{(by induction)}}{\le} \gamma O(n^{-1}).$$

Therefore,

$$\left|\mathbb{E}[\widehat{V}_n(s_0)] - V^\star(s_0)\right| \le O(n^{-1}),$$

that concludes the proof. □

Next, we present the results of Theorem 6. The proof follows the same steps as Theorem 5.

Theorem 6. ***(Convergence of Expected Payoff of PATS)*** *We have at the root node s_0,*

$$\mathbb{E}\left[\left|\widehat{V}_n\left(s_0\right) - V^\star(s_0)\right|\right] \le O(n^{-1}).$$

F Limitations

Computational Demands: The CATS distributional Monte Carlo Tree Search (MCTS) faces challenges in managing computational demands while maintaining and updating probability distributions, leading to a slightly increased complexity.
Fixed precision: The PATS set of particles can increase in size if the observed value are different. We prevent this in the implementation by fixing the float precision.
Number of atoms: Our approach's performance is slightly influenced by hyperparameters, with the number of atoms being a critical factor. Suboptimal choices may affect performance.

G Experimental setup

All the experiments were done on 8 Intel Xeon Gold 6130 (Skylake), x86_64, 2.10GHz, 2 CPUs/node, 16 cores/CPU. Whenever feasible, we opted for open-source implementations of algorithms and environments.

Parameters selection We search the number of atoms from {10,20,...,100} and choose the results with best performances. We set the discount factor $\gamma = .99$ for MDPs, and $\gamma = .95$ for POMDPs. For UCT, we use the exploration constant $C = \sqrt{2} \times (R_{\max} - R_{\min})$.

Atari hyperparameters We run CATS in Atari with 10 random seeds, where each seed with 512 samples and collect the average score. We found that only 512 simulations were necessary due to the utilization of a pretrained neural network. We run CATS with 100 atoms. The temperature parameter τ of MENTS and TENTS is tuned from {0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. The selected parameter τ are shown in Table 4. The exploration constant ϵ for MENTS and TENTS are set to 0.01. For Power-UCT, we select the power mean $p = 2$.

Atari

Table 3: Average scores in Atari with 512 samples (10 seeds) $\pm$ 2 times std.

	CATS	PATS	UCT	DQN	Power-UCT	TENTS	MENTS
Phoenix	3290.00 ± 1599.52	3619.00 ± 891.72	2450.00 ± 786.22	340.0 ± 0.00	560.00 ± 0.00	4423.00 ± 642.38	3098.30 ± 919.65
MsPacman	2058.00 ± 243.93	2232.00 ± 896.29	1792.00 ± 62.85	1930.00 ± 224.83	1982.00 ± 473.45	1922.00 ± 416.91	2018.30 ± 316.98
Alien	1765.0 ± 801.03	1724.00 ± 649.63	1900.00 ± 00.00	1094.00 ± 122.83	1748.00 ± 120.21	1613.00 ± 296.96	1508.60 ± 322.58
SpaceInvaders	826.0 ± 194.76	791.0 ± 332.52	525.00 ± 00.00	525.00 ± 0.00	672.00 ± 148.42	742.50 ± 193.53	832.55 ± 211.95
BeamRider	1952.00 ± 500.04	1848.0 ± 320.29	1889.60 ± 171.09	1952.00 ± 0.00	1577.60 ± 112.47	3013.00 ± 778.89	2822.18 ± 697.31
Asterix	6040.00 ± 1560.89	5495.00 ± 3106.64	5380.00 ± 1464.05	6220.00 ± 156.80	5540.00 ± 863.39	5180.00 ± 528.19	5576.00 ± 1397.91
Robotank	11.50 ± 2.11	11.9 ± 1.51	12.2 ± 1.04	10.20 ± 0.39	11.00 ± 1.55	12.10 ± 1.47	11.59 ± 1.36
Seaquest	3170.00 ± 787.61	3288.0 ± 889.41	3564.00 ± 86.83	2304.00 ± 531.31	2704.00 ± 318.93	2928.00 ± 801.11	3312.40 ± 390.77
Solaris	1062.0 ± 519.21	1196.00 ± 524.45	392.00 ± 198.61	1112.00 ± 521.53	452.00 ± 153.19	1168.00 ± 516.33	1118.20 ± 513.00
Asteroids	930.00 ± 100.12	953.00 ± 107.05	5380.00 ± 1464.05	860.00 ± 48.89	930.00 ± 54.66	1518.00 ± 121.48	1414.70 ± 261.59
Enduro	142.40 ± 31.21	131.10 ± 17.16	127.00 ± 10.07	133.60 ± 8.73	134.00 ± 6.69	115.40 ± 18.82	128.79 ± 16.26
Atlantis	35890.00 ± 1914.28	36180.0 ± 2592.70	34300.00 ± 00.00	34480.00 ± 119.76	35420.00 ± 1494.63	36280.00 ± 1476.24	36277.00 ± 1811.53
Hero	3006.50 ± 9.16	3020.50 ± 27.24	3011.50 ± 17.04	3005.00 ± 9.53	2998.00 ± 35.16	3008.00 ± 0.00	3044.55 ± 181.04
Frostbite	1582.00 ± 1041.37	1580.00 ± 1127.23	1900.00 ± 00.00	2407.00 ± 116.76	1754.00 ± 651.38	2357.00 ± 398.45	2388.20 ± 320.37
WizardOfWor	670.0 ± 192.09	590.00 ± 359.02	200.00 ± 00.00	530.00 ± 92.63	640.00 ± 134.53	1210.00 ± 183.52	1211.00 ± 314.30
Breakout	315.00 ± 85.80	302.10 ± 70.47	271.8 ± 54.63	288.10 ± 53.01	289.00 ± 44.46	337.00 ± 15.91	309.03 ± 35.13

Atari environments (4) provide diverse video game-inspired scenarios commonly used in reinforcement learning research. These environments offer challenges based on classic Atari 2600 games (23; 38; 6). To explore enhanced exploration in deep reinforcement learning, we employ a Deep Q-Network pre-trained following the experimental setup outlined in (23). This pre-trained network initializes action-values for each node, combined with a Monte-Carlo Tree Search method similar to the AlphaGo one. Here, P_{prior} represents the Boltzmann distribution derived from the action-values $Q(s, .)$ computed by the network. The results in Table 3 show that CATS and PATS outperform UCT, DQN, Power-UCT, TENTS and MENTS in most of the games. For example, CATS is significant better than other methods in *Breakout, Enduro*, while PATS is significant better than other methods in *MsPacman, Solaris*. Our intention in this experiment is not to assert exceptional superiority, but rather to emphasize that CATS and PATS actually work in complicated Atari benchmark.

Table 4: The hyperparameter τ (temperature) for MENTS and TENTS in Atari.

	MENTS	TENTS
Phoenix	0.07	0.6
MsPacman	0.09	0.03
Alien	0.1	0.03
SpaceInvaders	0.02	0.06
BeamRider	0.02	0.03
Asterix	0.02	0.1
Robotank	0.01	0.05
Seaquest	0.02	0.03
Solaris	0.03	0.06
Asteroids	0.08	0.2
Qbert	0.02	0.4
Enduro	0.02	0.1
Atlantis	0.08	0.03
Hero	0.4	0.03
Frostbite	0.01	0.02
WizardOfWor	0.1	0.01
Breakout	0.02	0.04

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

(i) **Claims**

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes] ,

Justification: We discuss the problem of planning in stochastic environments and we present a method to tackle problem with clear contributions.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

(ii) **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitation in Section 6

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

(iii) **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide the main theorems in the main paper and proofs in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

(iv) **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Code and reproducibility steps are provided in supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

(v) **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Full code is available in supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

(vi) **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental setting is detailed in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

(vii) **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide error bars for the plots. For Atari, we report the standard deviation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

(viii) **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the details about the computer resources used (CPU and number of cores).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

(ix) **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conducted in the paper conforms the Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

(x) **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The research conducted in the paper has no societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

(xi) **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The research proposed in this paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

(xii) **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: We do not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

(xiii) **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The provided code is well documented.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

(xiv) **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

(xv) **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.