DISTRIBUTIONAL MONTE-CARLO TREE SEARCH WITH THOMPSON SAMPLING IN STOCHASTIC ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We focus on a class of reinforcement learning algorithms, Monte-Carlo Tree Search (MCTS), in stochastic settings. MCTS has excelled in deterministic domains but can struggle in highly stochastic scenarios where transition randomness and partial observability lead to underexploration and suboptimal value estimates. To address these challenges, we integrate *distributional* Reinforcement Learning (RL) with Thompson Sampling and an optimistic exploration bonus, resulting in two novel *distributional MCTS* algorithms: `CATSO` (Categorical Thompson Sampling with Optimistic Bonus) and `PATSO` (Particle Thompson Sampling with Optimistic Bonus). In both methods, each Q-node in the search tree maintains a distribution of returns—via either a fixed set of categorical atoms (`CATSO`) or a dynamic set of particles (`PATSO`). We then employ Thompson Sampling plus a polynomial optimism bonus to drive exploration in stochastic environments. Theoretically, we show that both algorithms attain a non-asymptotic, problem-dependent simple regret bound of $\mathcal{O}(n^{-1/2})$, with n as the number of visited trajectories. Empirical evaluations confirm that our distributional approach significantly improves performance over existing baselines, demonstrating its potential for robust online planning under uncertainty.

1 INTRODUCTION

Online planning in Markov Decision Processes (MDPs) involves making real-time decisions based on the current state of the environment, balancing exploration and exploitation under uncertainty and partial observability. Monte-Carlo Tree Search (MCTS) is one of the most powerful online planning methods for tackling complex MDPs. It has demonstrated impressive performance in various settings, including board games such as Chess and Go (Silver et al., 2017a; 2016; 2017b; Schrittwieser et al., 2020), video game strategy (Perez et al., 2014), robot assembly (Funk et al., 2022), robot path planning (Eiffert et al., 2020; Dam et al., 2022), and autonomous driving (Mo et al., 2021).

Despite these achievements, most standard MCTS methods primarily excel in *deterministic* or near-deterministic environments and often overlook the significant impact of stochasticity common in real-world scenarios. In highly random or partially observable settings, classical MCTS can suffer from inaccurate value estimates and suboptimal decisions, ultimately degrading overall performance. Hence, methods that handle stochasticity in MCTS are urgently needed for robust real-world deployment.

Value Estimation and Action Selection in MCTS. A crucial aspect of MCTS is *value estimation*. Classic approaches either use the empirical average for backup (as in UCT (Kocsis et al., 2006)), which can systematically *underestimate* optimal values, or use a maximum operator, which tends to *overestimate* them (Coulom, 2006). The *power mean* estimator (Dam et al., 2019) provides a balanced solution by computing a mean between the average and maximum values. In our approach, we adopt this principle at each V-node, storing the power mean of child Q-nodes' empirical means to avoid modeling V as a distribution.

For *action selection*, many MCTS algorithms borrow exploration ideas from Multi-Armed Bandits. UCT (Kocsis et al., 2006) extends the UCB1 confidence-bound rule but can underperform if its bonus constant is tuned incorrectly (Shah et al., 2020). One way to improve exploration is

through *Thompson Sampling* (TS), which randomly samples from a posterior over Q-values to balance exploration and exploitation. However, purely sampling from the posterior can lead to insufficient exploration if the posterior underestimates the potential of less-visited actions. To address this, several works propose *augmenting* Thompson sampling with an explicit optimism or confidence bonus—what may be referred to as *Optimistic Thompson Sampling* (May et al., 2012), *Bayes-UCB* (Kaufmann et al., 2012), or *PSRL + Optimism* (Osband et al., 2013; 2018). Some of these methods have also been explored in MCTS contexts, for instance via variance-based exploration (Bai et al., 2013; 2014) and Bayesian model adaptation. These ideas share a common motivation: *if the posterior is temporarily miscalibrated, an optimism bonus helps ensure that rarely tried actions are not prematurely discarded*.

Entropy Regularization. Another approach to balancing exploration and exploitation is *entropy regularization*. Maximum Entropy Tree Search (MENTS) (Xiao et al., 2019) incorporates a maximum-entropy policy objective, while RENTS and TENTS (Dam et al., 2021) extend this with Relative and Tsallis entropy. However, these methods rely on a temperature parameter that may impede convergence, and their estimated values converge to a *regularized* rather than *true* optimal value. Boltzmann-based approaches (Painter et al., 2024) can achieve exponential decay of simple regret but again hinge on carefully tuned temperature parameters.

Distributional Reinforcement Learning. On the RL side, *Distributional RL* (Bellemare et al., 2017; Dabney et al., 2018b; Mavrin et al., 2019) extends the classical Bellman equation to model full *value distributions*, thus capturing randomness in rewards and transitions more faithfully. Techniques like the C51 algorithm (Bellemare et al., 2016) represent Q-values via a discrete set of *atoms*, while quantile-based methods (Dabney et al., 2018a) approximate the distribution more flexibly. Nevertheless, most distributional approaches focus on *learning* rather than *planning*, leaving open questions about how best to incorporate distributional estimates into MCTS.

Contributions and Outline. In this paper, we bridge this gap by introducing *distributional* MCTS with *Thompson Sampling* and a *polynomial optimism bonus*. Specifically:

- We propose two *distributional* MCTS algorithms, `CATSO` (Categorical Thompson Sampling with Optimistic Bonus) and `PATSO` (Particle Thompson Sampling with Optimistic Bonus), each capturing Q-node distributions differently. `CATSO` discretizes Q-values into a fixed set of atoms, whereas `PATSO` grows a set of sampled particles. Both methods also integrate a UCB-style polynomial bonus to ensure that underexplored actions receive additional optimism, following the principle of *Optimistic Thompson Sampling* (May et al., 2012; Bai et al., 2013).
- We show theoretically that both approaches achieve a non-asymptotic *simple regret* rate of $O(n^{-1/2})$ with n as the number of visited trajectories, matching the known existing bounds of Fixed-depth-MCTS (Shah et al., 2022).
- We prove that our algorithms produce near-optimal policies for Wasserstein Distributionally Robust MDPs, achieving sample complexity bounds of $O\left(\left[\frac{R_{\max}^3}{\varepsilon(1-\gamma)^3}\log\left(\frac{H|\mathcal{S}|^2|\mathcal{A}|}{\delta}\right)\right]^{2H}\right)$ for learning (ε,δ)-robust policies.
- Empirical evaluations on synthetic trees and Atari tasks confirm the viability of our distributional and optimism-based strategies.

Overall, our work provides a new way to handle stochastic environments in MCTS through the synergy of distributional value representations, Thompson sampling, and optimism. The subsequent sections detail the algorithms, theoretical analysis, and experimental validation.

2 Setting

We study an infinite-horizon discounted MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ with state space $\mathcal{S}$, action set $\mathcal{A}$, transition distribution $\mathcal{P}(\cdot \mid s, a)$, discount factor $\gamma \in (0, 1]$, and reward function $\mathcal{R}(s, a, s') \in [0, R_{\max}]$. The *optimal* action-value function $Q^\star$ satisfies the Bellman equation

$$Q^\star(s,a) \;=\; \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\Big[\mathcal{R}(s,a,s') \;+\; \gamma \max_{a'} Q^\star(s',a')\Big].$$

MCTS *plans* from an initial state s_0 by repeatedly simulating trajectories of length H (for analysis), resulting in an estimate $\widehat{Q}_n(s_0,a)$ of the root action-value. Let $V^\star(s_0) \triangleq \max_a Q^\star(s_0,a)$ and $\widehat{V}_n(s_0) \triangleq \max_a \widehat{Q}_n(s_0,a)$. The *simple regret* at time n is:

$$R(s_0,n) \;=\; V^\star(s_0) - \widehat{V}_n(s_0).$$

To analyze MCTS performance, we define planning horizon H and playout policy π_0 with value V_0. For node s_h at depth h from root s_0, estimated value functions are defined inductively: $\widetilde{V}(s_H) = V_0(s_H)$ at depth H, and for $h \leqslant H-1$:

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

where $r(s_h, a)$ is the expected reward for action a in state s_h. This yields the approximation bound:

$$|Q^\star(s_0, a) - \widetilde{Q}(s_0, a)| \leqslant \gamma^H \|V^\star - V_0\|_\infty,$$

where the norm is over states reachable within H steps from s_0. MCTS minimizes the simple regret by accurately estimating $\widetilde{Q}(s_0, a)$ and $\widetilde{V}(s_0)$ to approximate $V^\star(s_0), Q^\star(s_0, a)$ and identify optimal action $a_\star = \arg\max_a Q^\star(s_0, a)$ at the root.

Our main theorems show that both CATSO and PATSO achieve the convergence rate of $\mathcal{O}(n^{1/2})$, matching that of earlier methods like Fixed-Depth MCTS (Shah et al., 2022).

2.1 Monte-Carlo Tree Search Background

MCTS iterates four key steps: *Selection* (from s_0, select actions according to a tree policy until reaching a leaf node or an unexpanded node), *Expansion* (add the newly visited node to the search tree), *Simulation* (perform a short rollout or simulation from the leaf), and *Backpropagation* (use the reward outcomes to update node statistics along the path). Unlike standard MCTS that stores a single mean Q-value, we store *distributional* Q-values, updated with each new sample.

2.2 Distributional RL in Tree Search

Formally, we view each Q-value as a distribution over possible returns. By letting $\mathcal{X}(s, a)$ denote the immediate reward distribution, we have

$$\mathcal{Q}(s, a) \overset{D}{=} \mathcal{X}(s, a) + \gamma\, \mathcal{V}(s'),$$

where $s' \sim \mathcal{P}(\cdot \mid s, a)$ and $\mathcal{V}(s')$ merges the Q distributions under the chosen policy (tree policy in MCTS). Conceptually, we track a distribution $\mathcal{Q}(s, a)$ at each node rather than a single mean. Next, we show how to implement this using two parameterizations: *categorical* and *particles*.

3 Distributional Thompson Sampling in Tree Search

A pure Thompson Sampling (TS) approach can sometimes *under*-estimate the value of less-visited actions, leading to insufficient exploration in stochastic environments. To address this, several works have proposed *combining* TS with an explicit optimism bonus or confidence bound. For instance, *Optimistic Thompson Sampling* May et al. (2012) adds a UCB-style term to the TS-sampled value in multi-armed bandits; *Bayes-UCB* Kaufmann et al. (2012) selects actions by the upper-quantile of a Bayesian posterior; and *PSRL + Optimism* Osband et al. (2013; 2018) incorporates additional confidence bonuses into posterior sampling for tabular or deep RL. In Monte Carlo Tree Search (MCTS), Bai et al. (2013; 2014) investigated Bayesian model adaptation with variance-based exploration, effectively mixing posterior sampling and optimism. These lines of work underscore a shared principle: *augmenting Thompson-drawn estimates with a boost for rarely visited actions* can enhance exploration if the posterior is temporarily overconfident or miscalibrated.

Building upon these ideas, we propose two new *distributional* MCTS algorithms that combine TS *and* an optimism bonus:

- **CATSO** (*Categorical Thompson Sampling with Optimistic Bonus*),
- **PATSO** (*Particle Thompson Sampling with Optimistic Bonus*).

Both methods maintain a *distribution* over Q-values at each node, using Thompson sampling to select actions, and then *add* a polynomial exploration bonus. In **CATSO**, Q-nodes track a fixed set of categorical *atoms*; in **PATSO**, Q-nodes maintain a growing *particle set*. This synergy aims to capture the uncertainty in return estimates while preventing underexploration.

3.1 Categorical Thompson Sampling with Optimistic Bonus (CATSO)

Each Q-node (s, a) maintains:

- **Atom locations:** $\{z_i(s, a)\}_{i=0}^{N-1}$, equally spaced in $[Q_{\min}, Q_{\max}]$,

Algorithm 1: CATSO

SelectAction (s_h) *(Sec 3.3)*
- **for** $a \in [A]$ **do**
  - $L(s_h, a) \sim \text{Dir}(\alpha^0(s_h, a), \ldots, \alpha^N(s_h, a))$
  - $\overline{\varphi}(s_h, a) = [z_0(s_h, a), \ldots, z_N(s_h, a)]^\top L(s_h, a)$
- **end**
- $a = \underset{a}{\text{argmax}} \{\overline{\varphi}(s_h, a) + B(n, s_h, a)\}$
- **return** a

SimulateV (s_h, t) *(Sec 3.3)*
- $a =$ `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 3.3)*
- $s_{h+1} \sim \mathbb{P}(\cdot|s_h, a)$, $r_t(s_h, a) \sim \mathcal{R}(s_h, a, s_{h+1})$
- **if** *Node* s_{h+1} *not expanded* **then**
  - `Rollout` (s_{h+1})
- **else**
  - `SimulateV` (s_{h+1}, t)
- **end**
- $T_{s_h,a}(t) = T_{s_h,a}(t) + 1$
- $\overline{Q}_t(s_h, a) = r_t(s_h, a) + \gamma \widehat{V}(s_{h+1})$
- **if** $\overline{Q}_t(s_h, a) \notin [Q_{\min}(s_h, a), Q_{\max}(s_h, a)]$ **then**
  - $Q_{\max}(s_h, a) = \max\{\overline{Q}_t(s_h, a), Q_{\max}(s_h, a)\}$
  - $Q_{\min}(s_h, a) = \min\{\overline{Q}_t(s_h, a), Q_{\min}(s_h, a)\}$
  - $\triangle z = \frac{Q_{max} - Q_{min}}{N}$
  - $z_i(s_h, a) = Q_{min} + i \triangle z : 0 \leqslant i \leqslant N$
- Update $p(s_h, a) = [p_0(s_h, a), \ldots, p_N(s_h, a)]$

Algorithm 2: PATSO

SelectAction (s_h) *(Sec 3.3)*
- **for** $a \in [A]$ **do**
  - $L(s_h, a) \sim \text{Dir}(\alpha(s_h, a))$
  - $\overline{\varphi}(s_h, a) = \mathcal{S}(s_h, a)^\top L(s_h, a)$
- **end**
- $a = \underset{a}{\text{argmax}} \{\overline{\varphi}(s_h, a) + B(n, s_h, a)\}$
- **return** a

SimulateV (s_h, t) *(Sec 3.3)*
- $a =$ `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 3.3)*
- $s_{h+1} \sim \mathbb{P}(\cdot|s_h, a)$, $r_t(s_h, a) \sim \mathcal{R}(s_h, a, s_{h+1})$
- **if** *Node* s_{h+1} *not expanded* **then**
  - `Rollout` (s_{h+1})
- **else**
  - `SimulateV` (s_{h+1}, t)
- **end**
- $T_{s_h,a}(t) = T_{s_h,a}(t) + 1$
- $\overline{Q}_t(s_h, a) = r_t(s_h, a) + \gamma \widehat{V}(s_{h+1})$
- **if** $\overline{Q}_t(s, a) \in \{\mathcal{S}(s_h, a)\}$ **then**
  - $\alpha_t(s_h, a)$ += 1 //$\alpha_t(s_h, a)$: weight of $\overline{Q}_t(s_h, a)$
- **else**
  - $\mathcal{S}(s_h, a) := (\mathcal{S}(s_h, a), \overline{Q}_t(s_h, a))$
  - $\alpha(s_h, a) := (\alpha(s_h, a), 1)$
- **end**

Figure 1: Comparing CATSO (left) and PATSO (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 CATSO, we init $(\alpha^0(s, a), \ldots, \alpha^N(s, a)) = (1, \ldots, 1)$ and in PATSO, $\mathcal{S}(s, a) = (1), \alpha(s, a) = (\varnothing)$ for each s, a.

- **Probabilities:** $\{p_i(s, a)\}_{i=0}^{N-1}$, forming a categorical distribution over those atoms,
- **Dirichlet prior:** $\text{Dir}(\alpha^0(s, a), \ldots, \alpha^{N-1}(s, a))$ enabling Thompson sampling.

Initially set $Q_{\min} = 0$ and $Q_{\max} = 0.001$. Whenever a newly observed return

$$\overline{Q}_t(s, a) \;=\; r_t \;+\; \gamma\, \widehat{V}(s')$$

exceeds this range, we expand $[Q_{\min}, Q_{\max}]$ accordingly. If $\overline{Q}_t(s, a)$ falls in the interval of atom $z_m(s, a)$, we increment $\alpha^m(s, a)$. In detail,

Q-node Atom Update. To simplify the notation, we express all related quantities as functions of (s, a), such that $\overline{Q}_t, z_i, \alpha^i$ are written as $\overline{Q}_t(s, a), z_i(s, a), \alpha^i(s, a)$, respectively. To incorporate a new return $\overline{Q}_t$, we consider two cases:

(i) *Inside current support* ($z_0 \leq \overline{Q}_t \leq z_{N-1}$):
- Identify $m = \arg\min_i |\overline{Q}_t - z_i|$.
- Update $\alpha^m \leftarrow \alpha^m + 1$.

(ii) *Outside current support* ($\overline{Q}_t < z_0$ or $\overline{Q}_t > z_{N-1}$):
- Compute new bounds $z'_{\min} = \min(z_0, \overline{Q}_t)$, $z'_{\max} = \max(z_{N-1}, \overline{Q}_t)$.
- Create a new grid $z'_0, \ldots, z'_{N-1} = \texttt{linspace}(z'_{\min}, z'_{\max}, N)$.
- Initialize $\boldsymbol{\alpha}' = (\alpha'^0, \ldots, \alpha'^{N-1}) = (1, \cdots, 1)$.
- For each old z_i: find $j(i) = \arg\min_k |z'_k - z_i|$, then $\alpha'^{j(i)} \leftarrow \alpha'^{j(i)} + \alpha^i$.
- Find $m = \arg\min_k |\overline{Q}_t - z'_k|$, increment $\alpha'^m \leftarrow \alpha'^m + 1$.
- Overwrite (z_i, α^i) with (z'_i, α'^i) for $i = 0, \ldots, N-1$.

Either way, the revised categorical PMF is

$$p_i^{\text{new}} = \frac{\alpha^{\text{i new}}}{\sum_{j=0}^{N-1} \alpha^{\text{i new}}} \quad \text{for } i = 0, \ldots, N-1.$$

Action Selection. At each node s_h:

(i) **Sample (Thompson):** For each action a, draw $L(s_h, a) \sim \mathrm{Dir}\big(\alpha^0(s_h, a), \ldots, \alpha^{N-1}(s_h, a)\big)$, then compute $\overline{\varphi}(s_h, a) = \sum_{i=0}^{N-1} z_i(s_h, a)\, L_i(s_h, a)$.

(ii) **Add optimism bonus Shah et al. (2020):** $B\big(n, s_h, a\big) = C\, \frac{T_{s_h}(n)^{1/4}}{T_{s,a}(n)^{1/2}}$, where $T_{s_h}(n)$ and $T_{s,a}(n)$ are the respective visit counts. Then $\overline{\varphi}'(s_h, a) = \overline{\varphi}(s_h, a) + B\big(n, s_h, a\big)$.

(iii) **Choose:** $a^* = \arg\max_{a'} \overline{\varphi}'(s_h, a')$.

In short, CATSO uses a fixed set of atom locations to track Q-values with Thompson sampling, while an added polynomial bonus encourages sufficient exploration.

3.2 Particle Thompson Sampling with Optimistic Bonus (PATSO)

Each Q-node (s, a) maintains:

- **Set of particles:** $\mathcal{S}(s, a)$, each storing a distinct observed return value.
- **Weights:** $\alpha(s, a)$, tracking how many times each particle has been revisited.

When a new sample $\overline{Q}_t(s, a)$ arrives:

- If $\overline{Q}_t(s, a) \in \mathcal{S}(s, a)$, increment its weight in $\alpha(s, a)$.
- Otherwise, add a new particle with weight 1.

Action Selection. At node s_h:

(i) **Sample Q-values (Thompson):** For each action a, draw $L(s_h, a) \sim \mathrm{Dir}\big(\alpha(s_h, a)\big)$, and then

$$\overline{\varphi}(s_h, a) = \sum_{i \in \mathcal{S}(s_h, a)} i \cdot L_i(s_h, a).$$

(ii) **Add optimism bonus Shah et al. (2020):** $B\big(n, s_h, a\big) = C\, \frac{T_{s_h}(n)^{1/4}}{T_{s,a}(n)^{1/2}}$, yielding

$$\overline{\varphi}'(s_h, a) = \overline{\varphi}(s_h, a) + B\big(n, s_h, a\big).$$

(iii) **Select:** $a^* = \arg\max_{a' \in \mathcal{A}} \overline{\varphi}'(s_h, a')$.

Because $\mathcal{S}(s, a)$ dynamically expands, PATSO can capture a more flexible approximation of the underlying Q distribution, while the polynomial optimism bonus ensures active exploration of rarely tried actions.

Remark 1 (CATSO vs. PATSO)**.** CATSO discretizes Q-values into $N + 1$ fixed atoms, potentially introducing approximation errors if N is small. PATSO grows its particle set automatically, often yielding a finer representation of the return distribution. However, carefully picking N in CATSO can be more efficient, whereas PATSO 's flexibility may require more storage or computation. Both methods share similar theoretical guarantees on regret.

3.3 Value Backup for V-Nodes

Although Q-nodes track full distributions, our V-nodes only store a *power-mean backup* of child Q-means:

$$\widehat{V}(s) = \Big(\sum_a \frac{T_{s,a}(n)}{T_s(n)} \big[\widehat{Q}(s, a)\big]^p\Big)^{\frac{1}{p}}, \quad p \geq 1.$$

This helps balance between plain averaging and a maximum operator Dam et al. (2019). In CATSO, $\widehat{Q}(s, a)$ is the expected value of the categorical distribution; in PATSO, it is the weighted average of all observed particles.

4 Theoretical Analysis

Our main goal is to prove that both CATSO and PATSO achieve a simple regret rate of $\mathcal{O}(n^{-1/2})$ under suitable assumptions. We begin by noting that each node in the search tree can be modeled as a *non-stationary multi-armed bandit* problem, since evolving the search (by expanding deeper nodes) gradually modifies the empirical rewards in a non-static manner. Subsequently, we extend these bandit-based findings to the entire MCTS procedure.

Algorithm 3: CATSO in Non-stationary bandits

Require: K arms; n: number of plays;
$N+1$ support size of categorical distributions
Init $(\alpha_a^0, \ldots, \alpha_a^N) = (1, \ldots, 1)$ for each $a \in [K]$
Main *()*
  for *t = 0,1,2,..., n* **do**
    for $a \in [A]$ **do**
      $L_{a,t} \sim \text{Dir}(\alpha_a^0, \ldots, \alpha_a^N)$
      $\overline{\varphi}_{a,t} = [0, \frac{R(t)}{N}, \frac{2R(t)}{N}, \cdots, R(t)]^\top L_{a,t}$
    end
    $a = \underset{a}{\text{argmax}} \left\{ \overline{\varphi}_{a,t} + C \frac{t^{\frac{1}{4}}}{T_a(t)^{\frac{1}{2}}} \right\}$
    $T_a(t) = T_a(t) + 1$
    Pull arm a and observe reward
    $R_{a,t} = \frac{mR(t)}{N}$ where $m \in \{0, 1, \ldots N\}$
    Update $\alpha_a^m = \alpha_a^m + 1$
  end

Algorithm 4: PATSO in Non-stationary bandits

Require: K arms; n: number of plays;
Init $\alpha_a = (1)$; $\mathcal{S}_a = (1)$ for each $a \in [K]$
Main *()*
  for *t = 0,1,2,..., n* **do**
    for $a \in [A]$ **do**
      $L_{a,t} \sim \text{Dir}(\alpha_a)$
      $\overline{\varphi}_{a,t} = \mathcal{S}_a^\top L_{a,t}$
    end
    $a = \underset{a}{\text{argmax}} \left\{ \overline{\varphi}_{a,t} + C \frac{t^{\frac{1}{4}}}{T_a(t)^{\frac{1}{2}}} \right\}$
    $T_a(t) = T_a(t) + 1$
    Pull arm a and observe reward $R_{a,t}$
    if $R_{a,t} \in \{\mathcal{S}_a\}$ **then**
      α_a^t += 1 //α_a^t : weight of $R_{a,t}$
    else
      $\mathcal{S}_a := (\mathcal{S}_a, R_{a,t})$
      $\alpha_a := (\alpha_a, 1)$
    end
  end

Figure 2: An illustration of CATSO (left) and PATSO (right) adapted to non-stationary bandits. Both incorporate a Thompson-sampled estimate plus an optimism bonus $C\, t^{1/4}/T_a(t)^{1/2}$.

4.1 Non-Stationary Bandit Setting

Consider a bandit with K actions (arms). Each arm a is associated with a possibly non-stationary reward process $\{R_{a,t}\}$, taking values in $[0, R]$. We let $\widehat{\mu}_{a,n}$ denote the empirical mean reward of arm a after it has been pulled n times, and we assume

$$\mu_a \;=\; \lim_{n\to\infty} \mathbb{E}\big[\widehat{\mu}_{a,n}\big].$$

Throughout, either a categorical-based or particle-based variant of Thompson Sampling (as described in CATSO and PATSO) governs how arms are selected. We further define a *power mean* operator to combine the arms' empirical means:

$$\widehat{\mu}_n(p) \;=\; \left(\sum_{a=1}^{K} \frac{T_a(n)}{n} \left[\widehat{\mu}_{a,T_a(n)}\right]^p \right)^{1/p},$$

where $T_a(n)$ counts the number of times arm a has been chosen in n total rounds. Our objective is to show that this power-mean estimator $\widehat{\mu}_n(p)$ concentrates around the largest mean reward $\mu_\star = \max_{a\in[K]} \mu_a$, at a rate implying an overall convergence of $\mathcal{O}(n^{-1/2})$ in our MCTS.

Definition 1. A sequence of estimators $(\widehat{V}_n)_{n\geqslant 1}$ *converges* to some value V with *concentration* if it satisfies:

(A) **Concentration**: For every $n \geqslant 1$ and any $\varepsilon > 0$, there is a constant $c > 0$ such that $\mathbb{P}\Big(|\widehat{V}_n - V| > \varepsilon\Big) \;\leq\; c\, \frac{n^{-1}}{\varepsilon^2}$.

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

When these two conditions hold, we write $\operatorname*{plim}_{n\to\infty} \widehat{V}_n = V$.

Algorithms in the Bandit Context. We combine Thompson Sampling with an optimistic term akin to *Stochastic-Power-UCT* Dam et al. (2024), ensuring each arm is initially sampled at least once. Figure 2 sketches the pseudocode of CATSO and PATSO in the simpler bandit domain:

Fundamental Assumption. To formalize the notion that each arm's empirical means converge, we adopt:

Assumption 1. *For each arm $a \in [K]$, let $\big(\widehat{\mu}_{a,n}\big)_{n\geq 1}$ be a sequence of estimators such that*

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

By "plim" we refer to the limit notion from Definition 1, ensuring both concentration and convergence in expectation. Under this assumption, we can prove the next two theorems.

Theorem 1. ***(CATSO in Non-Stationary Bandits)*** *Let $(\widehat{\mu}_{a,n})_{n\geq 1}$ be a bounded sequence in $[0, R]$ satisfying Assumption 1, and let $\mu_\star = \max_{a\in[K]} \mu_a$. Assume CATSO (the categorical-based TS) chooses each arm once initially, then follows the exploration strategy in Fig. 2. For any $p \geq 1$, the power-mean estimator $\widehat{\mu}_n(p)$ converges to $\mu_\star$ in the sense of Definition 1, i.e. $\operatorname*{plim}_{n\to\infty} \widehat{\mu}_n(p) = \mu_\star$.*

Theorem 2. ***(PATSO in Non-Stationary Bandits)*** *Under the same assumptions as Theorem 1, if instead PATSO (the particle-based TS) is employed, the identical convergence result holds: $\operatorname*{plim}_{n\to\infty} \widehat{\mu}_n(p) = \mu_\star$.*

Proofs of both theorems appear in the Appendix. Based on these bandit analyses, we can next examine how Q-value nodes converge in MCTS.

4.2 Extending to MCTS

Within MCTS, each node of the search tree can be interpreted as a (non-stationary) bandit instance: as the search tree grows, the empirical rewards from each node's children evolve. By applying Theorems 1 and 2 at every internal node, we show that each Q-node value concentrates around its truncated optimal value $\widetilde{Q}(s_h, a)$, and thus $\widetilde{V}(s_h)$ converges accordingly.

At the root s_0, we establish the convergence of the expected payoff, with detailed arguments deferred to the Appendix.

Theorem 5. ***(Convergence of Expected Payoff of CATSO)*** *We have at the root node s_0, there exists a constant $C' > 0$ such that $\mathbb{E}\big[\big|\widehat{V}_n(s_0) - \widetilde{V}(s_0)\big|\big] \leq C'\, n^{-\frac{1}{2}}$,*

Theorem 6. ***(Convergence of Expected Payoff of PATSO)*** *We have at the root node s_0, there exists a constant $C' > 0$ such that $\mathbb{E}\big[\big|\widehat{V}_n(s_0) - \widetilde{V}(s_0)\big|\big] \leq C'\, n^{-\frac{1}{2}}$,*

Remark 2. For a power-mean exponent $p \geq 1$, both CATSO and PATSO inherit the same $\mathcal{O}(n^{-1/2})$ rate seen in Stochastic-Power-UCT Dam et al. (2024). This encompasses well-known backup rules like *max* (when $p \to \infty$) or *mean* (when $p = 1$).

Thus, both of our distributional Thompson sampling approaches achieve a convergence rate on par with the best known results for fixed-depth MCTS, bridging the analysis of non-stationary bandits to the multi-level structure of tree search.

4.3 Distributional MCTS and Wasserstein Robust Optimization

The distributional MCTS approach presented in our paper extends naturally beyond sequential decision-making to the broader domain of robust planning under uncertainty. In this section, we establish formal connections between our algorithms (CATSO and PATSO) and the field of Wasserstein Distributionally Robust Optimization (WDRO), which offers a principled framework for decision-making under distributional uncertainty. We establish that for an appropriate choice of constants C and ε, CATSO/PATSO produces a near-optimal policy for the Wasserstein Distributionally Robust MDP (WDRMDP):

$$\max_{\pi} \min_{P\in\mathcal{B}_\varepsilon(\widehat{P})} \mathbb{E}_P[V^\pi]$$

where $\widehat{P}$ is the empirical distribution represented by the categorical or particle approach, $\mathcal{B}_\varepsilon(\widehat{P})$ is the Wasserstein ball of radius ε around $\widehat{P}$, and W_p is the p-Wasserstein distance. Formally, we derive a rigorous sample complexity guarantee for robust planning as follows.

Theorem 7. *Let $\mathcal{M}$ be an MDP with state space $\mathcal{S}$, action space $\mathcal{A}$, bounded rewards in $[0, R_{\max}]$, and discount factor γ. The required number of samples to learn an (ε, δ)-robust policy using the distributional representations from CATSO/PATSO combined with a concentration-based Wasserstein radius is of order $O\left(\left[\frac{R_{\max}^3}{\varepsilon(1-\gamma)^3}\log\left(\frac{H|\mathcal{S}|^2|\mathcal{A}|}{\delta}\right)\right]^{2H}\right)$.*

Remark 3. (Robust Planning via Distributional Estimation). The connection to WDRO reveals that our distributional MCTS algorithms naturally provide robustness against model uncertainty. By maintaining and updating full distributions rather than point estimates, CATSO and PATSO implicitly construct uncertainty sets that can be formalized within the Wasserstein framework. The exponential dependence on horizon H is typical for finite-horizon robust optimization problems, while the polynomial dependence on problem parameters $|\mathcal{S}|$, $|\mathcal{A}|$, and $R_{\max}$ demonstrates computational tractability for moderate-sized problems.

4.4 Addressing Memory and Computation Costs

CATSO memory complexity. Each (s, a) edge stores a categorical support of N atoms and their probabilities, so the memory is $O(N)$ per Q-edge. In the worst case, if one expands a full $|S| \times |A|$ frontier the memory scales as $O(N\,|S|\,|A|)$. In practice MCTS is budget limited and biased toward the optimal branch, so the number of distinct Q-edges visited is $O(n)$ for n rollouts, keeping memory linear in the explored tree size. Since N is fixed, CATSO's per-edge footprint does not grow over time.

PATSO memory management (cap with merge-on-insert). Without intervention, PATSO's particle set on a Q-edge may grow with the number of distinct returns. We cap it at $M \le K$ particles via *merge-on-insert*:

(i) Maintain a *sorted* list of pairs $\{(Q_i, \alpha_i)\}_{i=1}^{M}$ (value, weight) with $M \le K$.
(ii) When a new sample z arrives:
  (a) If z matches an existing value (within numeric tolerance): increment its weight.
  (b) Else if $M < K$: insert $(z, 1)$ in sorted order.
  (c) Else ($M = K$; overflow): merge the closest neighboring pair to free one slot (preserving first moment), then insert $(z, 1)$.

Guarantees of the cap. Let $\texttt{range}(s, a) := Q_{\max}(s, a) - Q_{\min}(s, a)$ be the local return range on a Q-edge. Nearest-neighbor merge preserves the *first moment* exactly and increases the W_1 distance to the uncapped empirical distribution by at most the merged gap times the merged mass. With a size cap K and merges between nearest neighbors, the total discrepancy satisfies $W_1(\widehat{P}, \widetilde{P}) = O\Big(\frac{\texttt{range}(s,a)}{K}\Big)$. Because the Bellman operator is $1/(1-\gamma)$-Lipschitz in W_1, the cap induces at the root an *additive* value error $O\Big(\frac{\texttt{range}}{K(1-\gamma)}\Big)$ and thus in simple regret $O\Big(\frac{\texttt{range}}{K(1-\gamma)^2}\Big)$. Combining with the baseline simple-regret rate $O(n^{-1/2})$ established in our analysis yields simple regret $= O\big(n^{-1/2}\big) + O\Big(\frac{\texttt{range}}{K(1-\gamma)^2}\Big)$. *Practical choice of K.* We set K so that the additive term is below the statistical term at our rollout budgets, e.g., choose $K \propto \sqrt{n}$ or pick the smallest K with $\texttt{range}/(K(1-\gamma)^2) \ll n^{-1/2}$.

Why the merge is safe. The merge replaces two adjacent masses $m_1\delta_{x_1} + m_2\delta_{x_2}$ by a single mass $(m_1{+}m_2)\delta_{x^\star}$ at $x^\star = \frac{m_1x_1+m_2x_2}{m_1+m_2}$, which preserves the first moment. The induced W_1 increase is $(m_1 + m_2) \cdot |x^\star - \{x_1, x_2\}| \le (m_1{+}m_2) \cdot |x_2{-}x_1|$. Over K bins with nearest-neighbor merges, the average gap is $O(\texttt{range}/K)$, yielding the bound above.

Complexity with the cap.

Operation	Cost
Per-(s, a) memory	$2K$ floats $\Rightarrow O(K)$
Per-selection compute	One Dirichlet draw over K bins + a dot product $\Rightarrow O(K)$
Insertion (incl. merge)	binary search $O(\log K)$ + constant-time merge $\Rightarrow O(\log K)$

CATSO has fixed $O(N)$ per-edge memory and linear compute in N; PATSO with capping has $O(K)$ per-edge memory and $O(K)$ selection compute with $O(\log K)$ amortized insertion. The cap keeps memory bounded while preserving the $O(n^{-1/2})$ convergence rate up to a tunable, additive term that vanishes as K grows.

5 Experiments

We compare CATSO and PATSO with several baselines, including UCT (Kocsis et al., 2006), Fixed-Depth-MCTS (Shah et al., 2022), MENTS/TENTS (Xiao et al., 2019; Dam et al., 2021), BTS (Painter et al., 2024), and DNG (Bai et al., 2013). We first evaluate on a *synthetic tree* environment, then on 17 Atari games (details in the Appendix). In all experiments, we set $\gamma = 1$ for the synthetic domain and $\gamma = 0.99$ for Atari, using $N = 100$ atoms for CATSO.

5.1 SyntheticTree Environment

The *SyntheticTree* (Dam et al., 2021) has depth d, branching factor k, and stochastic edges. Each leaf node's reward is sampled from a Gaussian($\mu, \sigma = 0.5$), with $\mu \in [0, 1]$. Additionally, transitions are stochastic: with 50% probability the agent moves to an intended child and otherwise transitions uniformly among the other children. The agent's goal is to find the leaf node with the highest mean reward. We combine (k, d) in $\{2, 4, 6, 8, 10, 12, 14, 16, 100, 200\} \times \{1, 2, 3, 4\}$, repeating each combination over multiple runs.

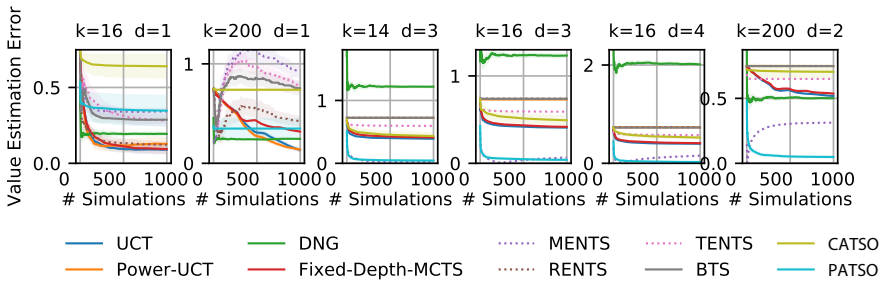

Figure 3: **Performance in the SyntheticTree environment.** We plot the absolute error of the estimated root value vs. the number of simulated trajectories n. Both CATSO and PATSO converge faster than non-distributional baselines, showcasing their robustness in stochastic transitions.

Table 1: Mean returns ± standard deviation on 12 Atari games. Best result per game is bold-faced.

Game	CATSO	PATSO	MENTS	RENTS	TENTS	UCT
Alien	1124 ± 154	**1980 ± 539**	238 ± 62	326 ± 116	1260 ± 372	1962 ± 689
Atlantis	**37540 ± 1828**	31340 ± 1920	10980 ± 2294	34980 ± 1995	16920 ± 6745	34580 ± 1743
BeamRider	**1973 ± 229**	1598 ± 178	406 ± 247	1594 ± 382	785 ± 89	1796 ± 292
Enduro	125 ± 11	130 ± 20	0 ± 0	45 ± 18	77 ± 25	**131 ± 28**
Frostbite	660 ± 567	**1794 ± 686**	100 ± 62	456 ± 427	242 ± 19	1146 ± 1003
Gopher	308 ± 210	380 ± 77	296 ± 183	**448 ± 445**	300 ± 206	376 ± 116
Hero	**3021 ± 22**	2994 ± 8	1645 ± 1282	2880 ± 53	2990 ± 0	2988 ± 26
MsPacman	2594 ± 705	1988 ± 56	244 ± 17	1724 ± 288	1566 ± 293	**2652 ± 818**
Phoenix	3760 ± 0	**5050 ± 0**	600 ± 0	4380 ± 0	4470 ± 465	1334 ± 500
Robotank	11.4 ± 1.0	**11.4 ± 3.1**	2.8 ± 1.6	10.0 ± 2.6	3.4 ± 2.2	11.0 ± 2.4
Seaquest	**3832 ± 560**	3316 ± 375	136 ± 59	296 ± 114	1176 ± 285	3004 ± 311
SpaceInvaders	637 ± 260	766 ± 270	209 ± 96	358 ± 168	468 ± 154	**1145 ± 464**

Figure 3 shows that both CATSO and PATSO converge more rapidly than baselines in terms of the root-value error. PATSO often outperforms CATSO because it does not rely on a fixed-atom approximation. However, the theoretical convergence rates of the two are the same.

5.2 Atari Experiments

We evaluated our algorithms on 12 Atari games using a pretrained DQN network as a feature extractor for MCTS variants, following Xiao et al. (2019); Dam et al. (2021). Table 1 summarizes the results. Our proposed algorithms demonstrate strong performance across the test suite. CATSO achieves the highest scores in 5 games (Atlantis, BeamRider, Hero, and Seaquest), while PATSO leads in 4 games (Alien, Frostbite, Phoenix, and Robotank). The performance gap is most significant in games requiring complex exploration - for instance, in Frostbite, PATSO outperforms MENTS by 17.9× and TENTS by 7.4×.

UCT remains competitive in Enduro, MsPacman, and SpaceInvaders, suggesting traditional confidence bounds remain effective for certain game dynamics. Both our methods excel particularly in environments with high stochasticity (e.g., Breakout, Enduro) and those requiring strategic exploration of sparse reward landscapes.

These results validate the effectiveness of our proposed exploration strategies in complex environments. Full details and additional analysis are provided in the supplementary material.

6 Conclusion

We presented two distributional MCTS algorithms, CATSO and PATSO, which model Q-nodes via categorical or particle-based distributions and utilize Thompson Sampling with a polynomial exploration bonus. Both methods enjoy a simple regret convergence rate of $O(n^{-1/2})$, matching the known convergence rate of Fixed-Depth-MCTS Shah et al. (2020). Additionally, we established formal connections to Wasserstein Distributionally Robust Optimization, proving that our algorithms produce near-optimal policies for distributionally robust MDPs with polynomial sample complexity guarantees. Empirical results on both synthetic and Atari domains confirm the benefits of distributional planning in stochastic environments. Future directions include scaling up to larger state-action spaces, exploring partial observability, and studying risk-sensitive or distribution-aware policies in more detail.

REFERENCES

Aijun Bai, Feng Wu, and Xiaoping Chen. Bayesian mixture modelling and inference based thompson sampling in monte-carlo tree search. *Advances in neural information processing systems*, 26, 2013.

Aijun Bai, Feng Wu, Zongzhang Zhang, and Xiaoping Chen. Thompson sampling based monte-carlo planning in pomdps. *the International Conference on Automated Planning and Scheduling*, 24(1), 2014.

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

Marc G Bellemare, Will Dabney, and Rémi 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.

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

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pages 1096–1105. PMLR, 2018a.

Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018b.

Tuan Dam, Pascal Klink, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Generalized mean estimation in monte-carlo tree search. *arXiv preprint arXiv:1911.00384*, 2019.

Tuan Dam, Georgia Chalvatzaki, Jan Peters, and Joni Pajarinen. Monte-carlo robot path planning. *IEEE Robotics and Automation Letters*, 7(4):11213–11220, 2022.

Tuan Dam, Odalric-Ambrym Maillard, and Emilie Kaufmann. Power mean estimation in stochastic monte-carlo tree search. In Negar Kiyavash and Joris M. Mooij, editors, *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pages 894–918. PMLR, 15–19 Jul 2024. URL `https://proceedings.mlr.press/v244/dam24a.html`.

Tuan Q Dam, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Convex regularization in monte-carlo tree search. In *International Conference on Machine Learning*, pages 2365–2375. PMLR, 2021.

Stuart Eiffert, He Kong, Navid Pirmarzdashti, and Salah 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.

Niklas Funk, Georgia Chalvatzaki, Boris Belousov, and Jan Peters. Learn2assemble with structured representations and search for robotic architectural construction. In *Conference on Robot Learning*, pages 1401–1411. PMLR, 2022.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *Artificial intelligence and statistics*, pages 592–600. PMLR, 2012.

Levente Kocsis, Csaba Szepesvári, and Jan Willemson. Improved monte-carlo search. *Univ. Tartu, Estonia, Tech. Rep*, 1, 2006.

Borislav Mavrin, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. In *International conference on machine learning*, pages 4424–4434. PMLR, 2019.

Benedict C May, Nathan Korda, Anthony Lee, David S Leslie, and Nicolo Cesa-Bianchi. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(6), 2012.

Shuojie Mo, Xiaofei Pei, and Chaoxian Wu. Safe reinforcement learning for autonomous vehicle using monte carlo tree search. *IEEE Transactions on Intelligent Transportation Systems*, 23(7): 6766–6773, 2021.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Michael Painter, Mohamed Baioumy, Nick Hawes, and Bruno Lacerda. Monte carlo tree search with boltzmann exploration. *Advances in Neural Information Processing Systems*, 36, 2024.

Diego Perez, Spyridon Samothrakis, and Simon 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.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Devavrat Shah, Qiaomin Xie, and Zhi 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.

Devavrat Shah, Qiaomin Xie, and Zhi Xu. Nonasymptotic analysis of monte carlo tree search. *Operation Research*, 70(6):3234–3260, 2022.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi: 10.1038/nature16961.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017b. URL `http://dx.doi.org/10.1038/nature24270`.

Chenjun Xiao, Ruitong Huang, Jincheng Mei, Dale Schuurmans, and Martin Müller. Maximum entropy monte-carlo planning. In *Advances in Neural Information Processing Systems*, pages 9516–9524, 2019.