A Unified Perspective on Value Backup and Exploration in Monte-Carlo Tree Search

Tuan Dam[1], **Carlo D'Eramo**[2,3], **Joni Pajarinen**[5], **Jan Peters**[3,4]
[1]Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189-CRIStAL, F-59000 Lille, France
[2]Center for Artificial Intelligence and Data Science, University of Würzburg, Germany
[3]Hessian.ai, Germany
[4]Department of Computer Science, Technical University of Darmstadt, Germany
[5]Department of Electrical Engineering and Automation, Aalto University, Finland

Abstract

Monte-Carlo Tree Search (MCTS) is a class of methods for solving complex decision-making problems through the synergy of Monte-Carlo planning and Reinforcement Learning (RL). The highly combinatorial nature of the problems commonly addressed by MCTS requires the use of efficient exploration strategies for navigating the planning tree and quickly convergent value backup methods. These crucial problems are particularly evident in recent advances that combine MCTS with deep neural networks for function approximation. In this work, we introduce a mathematical framework based on using the α-divergence for backup and exploration in MCTS. We show that this theoretical formulation unifies different approaches, including our newly introduced ones (Power-UCT and E3W), under the same mathematical framework, allowing us to obtain different methods by simply changing the value of α. In practice, our unified perspective offers a flexible way to balance exploration and exploitation by tuning the single α parameter according to the problem at hand. We validate our methods through a rigorous empirical study of a basic toy task Synthetic Tree problem as well as across several Atari games.

1 Introduction

Monte-Carlo Tree Search (MCTS) is an effective method that combines a random sampling strategy with tree search to determine the optimal decision for on-the-fly planning tasks. MCTS has yielded impressive results in Go [Silver *et al.*, 2016] (AlphaGo), Chess [Silver *et al.*, 2017a] (AlphaZero), or video games [Osband *et al.*, 2016], and it has been further exploited successfully in motion planning [Nguyen *et al.*, 2017; Sukkar *et al.*, 2019], autonomous car driving [Volpi *et al.*, 2017; Chen *et al.*, 2020], and autonomous robotic assembly tasks [Funk *et al.*, 2021]. Many of the MCTS successes [Silver *et al.*, 2016, 2017a,b] rely on coupling MCTS with neural networks trained using Reinforcement Learning (RL) [Sutton and Barto, 1998] methods such as Deep Q-Learning [Mnih *et al.*, 2015], to speed up learning of large scale problems.

Despite AlphaGo and AlphaZero achieving state-of-the-art performance in games with high branching factors like Go [Silver *et al.*, 2016] and Chess [Silver *et al.*, 2017a], both methods suffer from poor sample efficiency, mostly due to the inefficiency of the average mean backup operator, which is well-known for the issue of underestimating the optimum and leading to the polynomial convergence rate of PUCT [Xiao *et al.*, 2019]. This problem, combined with the need for effective exploration techniques, particularly in highly stochastic environments, poses an open research problem for the MCTS community: effective exploration methods and sufficient backup operators for the planning tree.

In this work, we provide a theory of the use of α-divergence in MCTS, respectively showing how the different range of α parameter solves the exploration-exploitation trade-off schema and prove that a

16th European Workshop on Reinforcement Learning (EWRL 2023).

class of our novel backup operators ensure the exponential convergence rate, showing the advantages over the polynomial convergence rate of UCT [Kocsis *et al.*, 2006]. We further draw the connection between the two recent advanced MCTS methods, Power-UCT [Dam *et al.*, 2019] and E3W [Dam *et al.*, 2021], which have been proven to provide effective solutions for the exploration and backup operator problems in the tree, by providing a rigorous theoretical study of α-divergence in MCTS and analyze how α-divergence can help to derive power mean and entropic regularization in MCTS.

α-divergence has been first extensively studied in RL context by Belousov and Peters [2019], and later on, has been proposed to use in Lee *et al.* [2019a] as a generalized Tsallis Entropy regularizer in MDP. However, the study of α-divergence in MCTS is still an open question. In this work, we first show that power mean (the new backup operator used in Power-UCT) can be derived as a closed-form solution of a mean of distribution by considering α-divergence as the probability distance, generalizing the eclipse distance that is used to derive average mean of a distribution. We further exploit the convex regularization framework in MCTS by analyzing the α-divergence function as the regularizer to introduce novel regularized backup operators for MCTS, relatively derive the maximum entropy, the relative entropy of the policy update, and, more importantly, derive the Tsallis entropy of the policy those has been proposed in E3W [Dam *et al.*, 2021]. Finally, we measure α-divergence in Synthetic Tree and show how α-divergence help to achieve competitive results in challenging problems.

2 Related Work

We want to improve the efficiency and performance of MCTS by addressing the two crucial problems of value backup and exploration. Our contribution follows on from a plethora of previous works that we briefly summarize in the following.

Backup operators. To improve upon the UCT algorithm in MCTS, Khandelwal *et al.* [2016] formalize and analyze different on-policy and off-policy complex backup approaches for MCTS planning based on techniques in the RL literature. Khandelwal *et al.* [2016] propose four complex backup strategies: MCTS(λ), MaxMCTS(λ), MCTS$_\gamma$, MaxMCTS$_\gamma$, and report that MaxMCTS(λ) and MaxMCTS$_\gamma$ perform better than UCT for certain parameter setups. Vodopivec *et al.* [2017] propose an approach called SARSA-UCT, which performs the dynamic programming backups using SARSA [Rummery, 1995]. Both Khandelwal *et al.* [2016] and Vodopivec *et al.* [2017] directly borrow value backup ideas from RL in order to estimate the value at each tree node. However, they do not provide any proof of convergence. The recently introduced MENTS algorithm [Xiao *et al.*, 2019], uses softmax backup operator at each node in combination with an entropy-based exploration policy, and shows a better convergence rate w.r.t. UCT.

Exploration. Entropy regularization is a common tool for controlling exploration in RL and has led to several successful methods [Schulman *et al.*, 2015; Haarnoja *et al.*, 2018; Schulman *et al.*, 2017; Mnih *et al.*, 2016]. Typically specific forms of entropy are utilized such as maximum entropy [Haarnoja *et al.*, 2018] or relative entropy [Schulman *et al.*, 2015]. This approach is an instance of the more generic duality framework, commonly used in convex optimization theory. Duality has been extensively studied in game theory [Shalev-Shwartz and Singer, 2006; Pavel, 2007] and more recently in RL, for instance considering mirror descent optimization [Montgomery and Levine, 2016; Mei *et al.*, 2019], drawing the connection between MCTS and regularized policy optimization [Grill *et al.*, 2020], or formalizing the RL objective via Legendre-Rockafellar duality [Nachum and Dai, 2020a]. Recently [Geist *et al.*, 2019] introduced regularized Markov Decision Processes, formalizing the RL objective with a generalized form of convex regularization, based on the Legendre-Fenchel transform. Several works focus on modifying classical MCTS to improve exploration. For instance, Tesauro *et al.* [2012] propose a Bayesian version of UCT to improve estimation of node values and uncertainties given limited experience.

α-divergence. The use of α-divergence in RL has been widely explored, particularly by Belousov and Peters [2019], who proposed using it to measure the divergence in policy search. Their work generalizes relative entropy policy search to constrain policy updates. Belousov and Peters [2019] studied a particular class of f-divergence, known as α-divergence, which resulted in compatible policy update and value function improvement in actor-critic methods. Another study by Lee *et al.* [2019a] analyzed α-divergence as a generalized Tsallis Entropy regularizer in MDP. By scaling the α parameter as an entropic index, Lee *et al.* [2019a] controlled the generalized Tsallis Entropy regularizer and derived Shannon-Gibbs entropy and Tsallis Entropy as special cases.

3 Preliminaries

3.1 Markov Decision Process

In the context of Reinforcement Learning (RL), an agent's goal is to determine how to interact with the environment modeled as a Markov Decision Process (MDP), which is a well-known mathematical framework for sequential decision-making. Our focus is on an infinite-horizon discounted MDP that can be represented as a 5-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the finite discrete action space with $|\mathcal{A}|$ representing the number of actions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the probability distribution over the next state s' given the current state s and action a, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi \in \Pi : \mathcal{S} \to \mathcal{A}$ is a probability distribution over possible actions a given the current state s.

A policy π induces a $\mathcal{Q}$ value function: $Q^\pi(s, a) \triangleq \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) | s_0 = s, a_0 = a, \pi\right]$, where $r(s_{i+1}, a_{i+1})$ is the reward obtained after the i-th transition induces by the policy π, respectively defining the value function under the policy π as $V^\pi(s) \triangleq \max_{a \in \mathcal{A}} Q^\pi(s, a)$. The Bellman operator under the policy π is defined as

$$\mathcal{T}_\pi Q(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \int_{\mathcal{A}} \pi(a'|s') Q(s', a') da'\right] ds'. \tag{1}$$

The goal is to find the optimal policy that satisfies the optimal Bellman equation [Bellman, 1954]

$$Q^*(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q^*(s', a')\right] ds', \tag{2}$$

which is the fixed point of the optimal Bellman operator

$$\mathcal{T}^* Q(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q(s', a')\right] ds'. \tag{3}$$

The optimal value function is defined $V^*(s) \triangleq \max_{a \in \mathcal{A}} Q^*(s, a)$.

3.2 Monte-Carlo Tree Search

Monte-Carlo Tree Search (MCTS) is a tree search method for MDPs that combines Monte-Carlo sampling, tree search, and multi-armed bandits to make optimal decisions efficiently. The MCTS tree is composed of nodes and edges that represent visited states and actions taken in each state, respectively. The algorithm consists of four main steps: Selection: where a *tree-policy* is used to traverse the tree from the root node to a leaf node. Expansion: where the new node is added to the tree according to the tree policy; Simulation: where a Monte-Carlo rollout or a neural network is used to estimate the value of the new node. Backup: where the collected reward is used to update the action-values along the path from the leaf node to the root node. The tree-policy used to select the action to execute in each node needs to balance the use of already known good actions, and the visitation of unknown states.

3.3 Upper Confidence bound for Trees

In this section, we present the MCTS algorithm UCT (Upper Confidence bounds for Trees) [Kocsis *et al.*, 2006], an extension of the well-known UCB1 [Auer *et al.*, 2002] multi-armed bandit algorithm. UCB1 chooses the arm (action a) using

$$a = \underset{i \in \{1 \dots K\}}{\arg\max} \overline{X}_{i, T_i(n-1)} + C\sqrt{\frac{\log n}{T_i(n-1)}}, \tag{4}$$

where $T_i(n) = \sum_{t=1}^{n} \mathbf{1}\{t = i\}$ is the number of times arm i is played up to time n. $\overline{X}_{i, T_i(n-1)}$ denotes the average reward of arm i up to time $n-1$ and $C = \sqrt{2}$ is an exploration constant. In UCT, each node is a separate bandit, where the arms correspond to the actions, and the payoff is the reward of the episodes starting from them. In the backup phase, value is backed up recursively from the leaf node to the root as

$$\overline{X}_n = \sum_{i=1}^{K} \left(\frac{T_i(n)}{n}\right) \overline{X}_{i, T_i(n)}. \tag{5}$$

Kocsis *et al.* [2006] proved that UCT asymptotically converges in the limit to the optimal policy.

3.4 α-divergence

The f-divergence[Csiszár, 1964] generalizes the definition of the distance between two probabilistic distributions P and Q on a finite set $\mathcal{A}$ as

$$D_f\left(P\|Q\right) = \sum_{a\in\mathcal{A}} Q(a) f\left(\frac{P(a)}{Q(a)}\right), \tag{6}$$

where f is a convex function on $(0,\infty)$ such as $f(1) = 0$. For example, the KL-divergence corresponds to $f_{KL} = x\log x - (x-1)$. The $\alpha-$divergence is a subclass of f-divergence generated by $\alpha-$function with $\alpha \in \mathbb{R}$. $\alpha-$function is defined as

$$f_\alpha(x) = \frac{(x^\alpha - 1) - \alpha(x-1)}{\alpha(\alpha-1)}. \tag{7}$$

The $\alpha-$divergence between two probabilistic distributions P and Q on a finite set $\mathcal{A}$ is defined as

$$D_\alpha\left(P\|Q\right) = \sum_{a\in\mathcal{A}} Q(a) f_\alpha\left(\frac{P(a)}{Q(a)}\right), \tag{8}$$

where $\sum_{a\in\mathcal{A}} Q(a) = \sum_{a\in\mathcal{A}} P(a) = 1$.
Furthermore, given the $\alpha-$function, we can derive the generalization of Tsallis entropy of a policy π as

$$H_\alpha(s) = \frac{1}{\alpha(1-\alpha)}\left(1 - \sum_{a\in\mathcal{A}} \pi(s,a)^\alpha\right). \tag{9}$$

In addition, we have

$$\lim_{\alpha\to 1} H_\alpha(s) = -\sum_{a\in\mathcal{A}} \pi(s,a)\log\pi(s,a) \qquad , \tag{10}$$

$$H_2(s) = \frac{1}{2}\left(1 - \sum_{a\in\mathcal{A}} \pi(s,a)^2\right), \tag{11}$$

respectively, the Shannon entropy (10) and the Tsallis entropy (11) functions.

3.5 Legendre-Fenchel Transform

Consider an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma\rangle$, as previously defined. Let $\Omega : \Pi \to \mathbb{R}$ be a strongly convex function. For a policy $\pi_s = \pi(\cdot|s)$ and $Q_s = Q(s,\cdot) \in \mathbb{R}^{\mathcal{A}}$, we observe that the Bellman operator $\mathcal{T}_{\pi_s} Q_s = \langle \pi(\cdot|s), Q(s,\cdot)\rangle = \langle \pi_s, Q_s\rangle$. The Legendre-Fenchel transform (or convex conjugate) of Ω is $\Omega^* : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$, defined as:

$$\Omega^*(Q_s) \triangleq \max_{\pi_s\in\Pi_s} \left\{\langle \pi_s, Q_s\rangle - \tau\Omega(\pi_s)\right\}, \tag{12}$$

where the temperature τ specifies the strength of regularization. Among the several properties of the Legendre-Fenchel transform, we use the following [Mensch and Blondel, 2018; Geist *et al.*, 2019; Lee *et al.*, 2019b].

Proposition 1. *Let Ω be strongly convex.*

- *Unique maximizing argument: $\nabla\Omega^*$ is Lipschitz and satisfies*

$$\nabla\Omega^*(Q_s) = \arg\max_{\pi_s\in\Pi_s} \left\{\langle \pi_s, Q_s\rangle - \tau\Omega(\pi_s)\right\}. \tag{13}$$

- *Boundedness: if there are constants L_Ω and U_Ω such that for all $\pi_s \in \Pi_s$, we have $L_\Omega \le \Omega(\pi_s) \le U_\Omega$, then*

$$\max_{a\in\mathcal{A}} Q_s(a) - \tau U_\Omega \le \Omega^*(Q_s) \le \max_{a\in\mathcal{A}} Q_s(a) - \tau L_\Omega. \tag{14}$$

- *Contraction: for any $Q_1, Q_2 \in \mathbb{R}^{\mathcal{A}}$*

$$\|\ \Omega^*(Q_1) - \Omega^*(Q_2)\ \|_\infty \le \gamma\ \|\ Q_1 - Q_2\ \|_\infty\ . \tag{15}$$

Note that if $\Omega(\cdot)$ is strongly convex, $\tau\Omega(\cdot)$ is also strongly convex; thus all the properties shown in Proposition 1 still hold[1]. Solving equation (12) leads to the solution of the optimal primal policy function $\nabla\Omega^*(\cdot)$. Since $\Omega(\cdot)$ is strongly convex, the dual function $\Omega^*(\cdot)$ is also convex. One can solve the optimization problem (12) in the dual space [Nachum and Dai, 2020b] as

$$\Omega(\pi_s) = \max_{Q_s \in \mathbb{R}^{\mathcal{A}}} \{\langle \pi_s, Q_s \rangle - \tau\Omega^*(Q_s)\} \tag{16}$$

and find the solution of the optimal dual value function as $\Omega^*(\cdot)$. We investigate the α-divergence function as a particular form of the convex regularizer with a specific value of a constant α to derive the entropy-based regularization methods in MCTS.

4 α-divergence in Monte-Carlo Tree Search

In this section, we show how to use α-divergence as a convex regularizer to generalize the entropy regularization in MCTS and respectively derive Maximum Entropy for Tree Search (MENTS), Relative Entropy for Tree Search (RENTS) and Tsallis Entropy for Tree Search (TENTS). Additionally, we show how to derive power mean (which is used as the backup operator in Power-UCT) using α-divergence as the distance function to replace the Euclidean distance in the definition of the empirical average mean value.

4.1 α-divergence Regularization

We introduce α-divergence regularization to MCTS. Denote the Legendre-Fenchel transform (or convex conjugate) of α-divergence regularization with $\Omega^* : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$, defined as:

$$\Omega^*(Q_s) \triangleq \max_{\pi_s \in \Pi_s} \langle \pi_s, Q_s \rangle - \tau H_\alpha(\pi_s), \tag{17}$$

where the temperature τ specifies the strength of regularization, and $H_\alpha(\pi_s)$ is the generalized Tsallis entropy derived from α function defined in (9). Note that α-divergence of the current policy π_s and the uniform policy has the same form as $H_\alpha(\pi_s)$. It is known that:

- The limit as $\alpha \to 1$ recovers the regularizer $H_1(\pi_s) = -\sum_{a \in \mathcal{A}} \pi(s,a) \log \pi(s,a)$, which is the Shannon entropy (MENTS [Dam *et al.*, 2021]). Note that if we apply the α-divergence with the limit as $\alpha \to 1$, we get Relative Entropy (RENTS [Dam *et al.*, 2021]);
- when $\alpha = 2$, we have the regularizer $H_2(\pi_s) = \frac{1}{2}\left(1 - \sum_{a \in \mathcal{A}} \pi(s,a)^2\right)$, and derive Tsallis entropy (TENTS [Dam *et al.*, 2021]).

For $\alpha > 1, \alpha \neq 2$ we can derive [Chen *et al.*, 2018]

$$\nabla\Omega^*(Q_t) = \left(\max\left\{\frac{Q^{\pi^*_\tau(s,a)}}{\tau} - \frac{c(s)}{\tau}, 0\right\}(\alpha - 1)\right)^{\frac{1}{\alpha-1}}, \tag{18}$$

where

$$c(s) = \tau\frac{\sum_{a \in \mathcal{K}(s)} \frac{Q^{\pi^*_\tau(s,a)}}{\tau} - 1}{\|\mathcal{K}(s)\|} + \tau\left(1 - \frac{1}{\alpha - 1}\right), \tag{19}$$

with $\mathcal{K}(s)$ representing the set of actions with non-zero chance of exploration in state s, as determined below

$$\mathcal{K}(s) = \left\{a_i \middle| 1 + i\frac{Q^{\pi^*_\tau(s,a_i)}}{\tau} > \sum_{j=1}^{i} \frac{Q^{\pi^*_\tau(s,a_j)}}{\tau} + i(1 - \frac{1}{\alpha - 1})\right\}, \tag{20}$$

where a_i denotes the action with the $i-$th highest Q-value in state s. and the regularized value function

$$\Omega^*(Q_t) = \left\langle \nabla\Omega^*(Q_t), Q^{\pi^*_\tau(s,a)} \right\rangle. \tag{21}$$

Using α-divergence, we can relatively derive MENTS, RENTS ($\alpha = 1$) and TENTS ($\alpha = 2$), which have been studied as entropy regularization in MCTS [Dam *et al.*, 2021]. Next, we will show how to connect to the power mean backup operator used in Power-UCT [Dam *et al.*, 2019] using α-divergence.

[1] Other works use the same formula, e.g. Equation (12) in Niculae and Blondel [2017].

4.2 Connecting Power Mean with α-divergence

In order to connect the Power-UCT [Dam *et al.*, 2019] approach with α-divergence, we study here the entropic mean [Ben-Tal *et al.*, 1989] which uses f-divergence, of which α-divergence is a special case, as the distance measure. Since power mean is a special case of the entropic mean, the entropic mean allows us to connect the geometric properties of the power mean used in Power-UCT with α-divergence.

In more detail, let $a = (a_1, a_2, ...a_n)$ be given strictly positive numbers and let $w = (w_1, w_2, ..., w_n)$ be given weights and $\sum_{i=1}^{n} w_i = 1, w_i > 0, i = 1...n$. Let's define $dist(\alpha, \beta)$ as the distance measure between $\alpha, \beta > 0$ that satisfies

$$dist(\alpha, \beta) = \begin{cases} 0 \text{ if } \alpha = \beta \\ > 0 \text{ if } \alpha \neq \beta \end{cases} \tag{22}$$

When we consider the distance as f-divergence between the two distributions, we get the entropic mean of $a = (a_1, a_2, ...a_n)$ with weights $w = (w_1, w_2, ..., w_n)$ as

$$\text{mean}_w(a) = \arg\min_{x>0} \left\{ \sum_{i=1}^{n} w_i a_i f\left(\frac{x}{a_i}\right) \right\}. \tag{23}$$

When applying $f_\alpha(x) = \frac{x^{1-p}-p}{p(p-1)} + \frac{x}{p}$, with $p = 1 - \alpha$, we get

$$\text{mean}_w(a) = \left(\sum_{i=1}^{n} w_i a_i^p \right)^{\frac{1}{p}}, \tag{24}$$

which is equal to the power mean. In the next section, we will present our α-divergence MCTS method using the regularized value backup and tree policy sampling.

5 Regularized Backup and Tree Policy

s: state
a: action
$N(s)$: number of simulations of V_Node of state s
$n(s, a)$: number of simulations of Q_Node of state s and action a
$V(s)$: Value of V_Node at state s. Default is 0
$Q(s, a)$: Value of Q_Node at state s, action a. Default is 0
$\tau(s, a)$: transition function
γ: discount factor
$\epsilon > 0$:

R = `Rollout`$(s, depth)$
- **if** $\gamma^{depth} < \epsilon$ **then**
  - **return** 0
- $a \sim \pi_{\text{Rollout}}(.)$
- $(s', r) \sim \tau(s, a)$
- **return** $r + \gamma$Rollout $(s', depth + 1)$

a = `SelectAction`(s)
- $a \sim \pi(a|s) = (1-\lambda_s)\nabla\Omega^*(Q(s, \cdot)/\tau)(a) + \frac{\lambda_s}{|\mathcal{A}|}$,
- where $\lambda_s = \epsilon|\mathcal{A}|/\log(\sum_a N(s,a)+1)$ with $\epsilon > 0$ as an exploration parameter,
- $\nabla\Omega^*$ defined at 18
- **return** a

a = `Search`(s)
- **while** *Time remaining* **do**
  - `SimulateV` $(s, 0)$
- **return** `SelectAction`(s)

`SimulateV`$(s, depth)$
- $a =$`SelectAction` (s)
- `SimulateQ` $(s, a,$ depth$)$
- $N(s) \leftarrow N(s) + 1$
- $V(s) \leftarrow \langle \nabla\Omega^*(Q(s, \cdot)), Q(s, \cdot) \rangle$,
- where $\nabla\Omega^*(Q(s, \cdot))$ defined at 18

`SimulateQ`$(s, a, depth)$
- $(s', r) \sim \tau(s, a)$
- **if** *Node* s' *not expanded* **then**
  - `Rollout`$(s',$ depth$)$
- **else**
  - `SimulateV` $(s',$ depth $+ 1)$
- $n(s, a) \leftarrow n(s, a) + 1$
- $Q(s, a) \leftarrow \frac{(\sum_a r_{s,a}) + \gamma . \sum_{s'} N(s') . V(s')}{n(s,a)}$

`MainLoop`
- **while** *resource budget remains* **do**
  - $a =$ `Search`(s)

Algorithm 1: Pseudocode of α-divergence MCTS.

The pseudocode of $\alpha-$divergence MCTS has been shown in Algorithm 1, which is identical to the four basic steps of an MCTS algorithm. MCTS has two types of nodes: V_Nodes corresponding to

state-values, and Q_Nodes corresponding to state-action values. An action is taken from the V_Node of the current state leading to the respective Q_Node, then it leads to the V_Node of the reached state. For each state s, the backup value of corresponding V_node is

$$V(s) \leftarrow \langle \nabla\Omega^*(Q(s,\cdot)), Q(s,\cdot)\rangle \,. \tag{25}$$

On the other hand, the backup value of Q_nodes is

$$Q(s,a) \leftarrow \frac{(\sum_a r_{s,a}) + \gamma \sum_{s'} N(s')V(s')}{n(s,a)}, \tag{26}$$

where γ is the discount factor, s' is the next state after taking action a from state s, and $r_{s,a}$ is the reward obtained executing action a in state s, $N(s')$ is the number of visits to state s', $n(s,a)$ is the number of visits of action a in state s.

Action is selected by sampling from a policy that is

$$\pi(a|s) = (1-\lambda_s)\nabla\Omega^*(Q_\Omega(s)/\tau)(a) + \frac{\lambda_s}{|\mathcal{A}|}, \tag{27}$$

where $\lambda_s = {\epsilon|\mathcal{A}|}/{\log(\sum_a n(s,a)+1)}$ with $\epsilon > 0$ as an exploration parameter, and $\nabla\Omega^*$ defined at 18. We call this sampling strategy α Extended Empirical Exponential Weight(α-E3W) to highlight the use of $\alpha-$divergence as a convex regularizer in MCTS.

6 Regret and Error Analysis of α-divergence in Monte-Carlo Tree Search

The exponential convergence of choosing the optimal regularized action at the root node has been guaranteed as the direct results from Dam *et al.* [2021] as $\alpha-$divergence is a special case of convex regularization in MCTS. Next, we will provide further results on the regret analysis and error analysis of value estimation in the tree.

6.1 Regret Analysis

At the root node, let each children node i be assigned with a random variable X_i, with mean value V_i, while the quantities related to the optimal branch are denoted by $*$, e.g. mean value V^*. At each timestep n, the mean value of variable X_i is V_{i_n}. The pseudo-regret [Coquelin and Munos, 2007] at the root node, at timestep n, is defined as $R_n^{\text{UCT}} = nV^* - \sum_{t=1}^n V_{i_t}$. Similarly, we define the regret of α-E3W at the root node of the tree as

$$R_n = nV^* - \sum_{t=1}^n V_{i_t} = nV^* - \sum_i \sum_{t=1}^n \mathbb{I}(i_t = i)V_{i_t} = nV^* - \sum_i V_i \sum_{t=1}^n \hat{\pi}_t(a_i|s), \tag{28}$$

where $\hat{\pi}_t(\cdot)$ is the policy at time step t, and $\mathbb{I}(\cdot)$ is the indicator function.
The expected regret is defined as

$$\mathbb{E}[R_n] = nV^* - \sum_{t=1}^n \langle \hat{\pi}_t(\cdot), V(\cdot)\rangle \,. \tag{29}$$

We measure how different values of α in the α-divergence function affect the regret in MCTS.

In the next theorems, we show the regret bound of α-E3W [Dam *et al.*, 2021] in MCTS with different ranges of α parameters.

Theorem 1. *When $\alpha \in (0,1)$, the regret of α-E3W is*

$$\mathbb{E}[R_n] \leq \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1) + n(2\tau)^{-1}|\mathcal{A}|^\alpha + \mathcal{O}(\frac{n}{\log n}).$$

For $\alpha \in (1,\infty)$, we derive the following results

Theorem 2. *When $\alpha \in (1,\infty)$, the regret of α-E3W is*

$$\mathbb{E}[R_n] \leq \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1) + \frac{n|\mathcal{K}|}{2} + \mathcal{O}(\frac{n}{\log n}).$$

where $|\mathcal{K}|$(defined at 20) is the number of actions that are assigned non-zero probability in the policy at the root node. Note that when $\alpha = 1, 2$, please refer to Corollary 1, 2, 3 [Dam *et al.*, 2021].

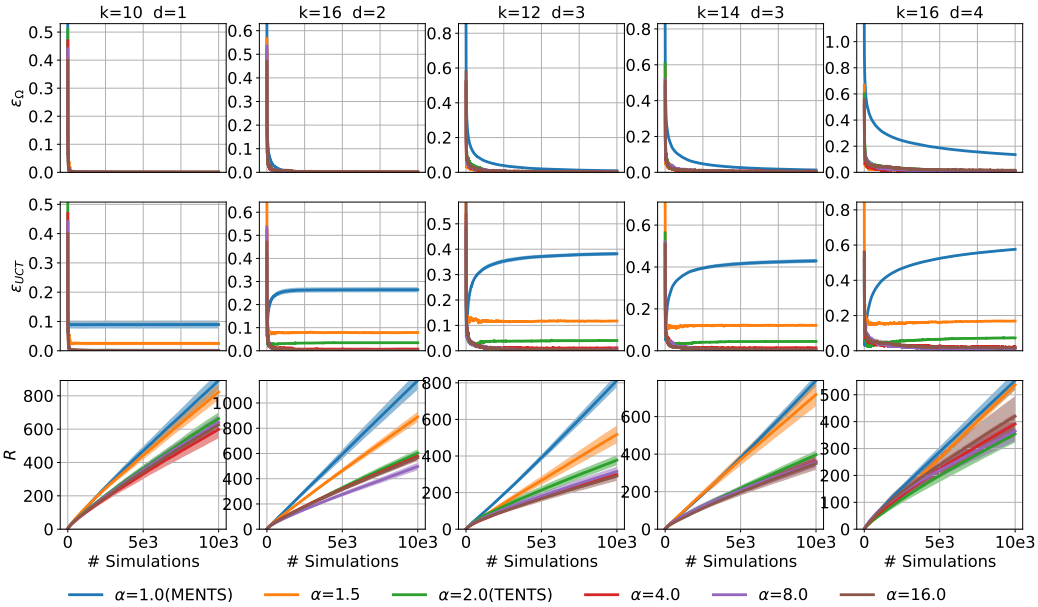


Figure 1: We show the convergence of the value estimate at the root node to the respective optimal value (top), to the UCT optimal value (middle), and the regret (bottom) with different α parameter of α-divergence in Synthetic tree environment with $\alpha = 1.0$ (MENTS), 1.5, 2.0 (TENTS), 4.0, 8.0, 16.0.

6.2 Error Analysis

We analyze the error of the regularized value estimate at the root node $n(s)$ w.r.t. the optimal value: $\varepsilon_\Omega = V_\Omega(s) - V^*(s)$. where Ω is the α-divergence regularizer $H_\alpha($ defined at 9).

Theorem 3. *For any $\delta > 0$ and α-divergence regularizer H_α ($\alpha \neq 1, 2$), with some constant $C, \hat{C}$, with probability at least $1 - \delta$, ε_Ω satisfies*

$$-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1) \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}. \tag{30}$$

For $\alpha = 1, 2$, please refer to Corollary 4, 5, 6 [Dam *et al.*, 2021]. We observe that when α increases, the error bound decreases.

7 Empirical Evaluation

In this section, we plan to measure the effectiveness of the difference range value of α parameter in MCTS and show how α-divergence help to trade off between exploration-exploitation.

7.1 Synthetic Tree

We first use the toy problem Synthetic Tree [Xiao *et al.*, 2019] to measure how the α-divergence helps to balance exploration and exploitation in MCTS. Synthetic Tree involves a tree with depth d and branching factor k. Each edge of the tree has a random value between 0 and 1, and at each leaf, a Gaussian distribution is used as an evaluation function resembling the return of random rollouts. The mean of the Gaussian distribution is the sum of the values assigned to the edges connecting the root node to the leaf, while the standard deviation is $\sigma = 0.05$[2] The mean value of each distribution at each node of the toy problem is normalized between 0 and 1 for stabilizing. We set the temperature $\tau = 0.1$ and the exploration $\epsilon = 0.1$. Figure 2 illustrates the heatmap of the absolute error of the value estimate at the root node after the last simulation of each algorithm w.r.t. the respective optimal regularized value, the optimal value of UCT and regret at the root node with $\alpha = 1.0$ (MENTS), 1.5,

[2]The value of the standard deviation is not provided in Xiao *et al.* [2019]. After trying different values, we observed that our results match the one in Xiao *et al.* [2019] when using $\sigma = 0.05$.

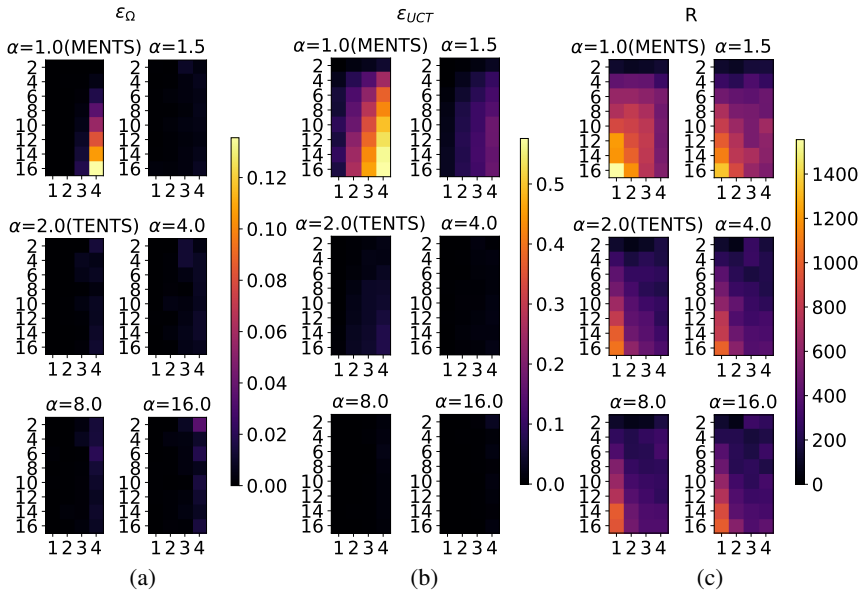


Figure 2: We show the effectiveness of α-divergence in Synthetic Tree environment with different branching factor k (rows) and depth d (columns). The heatmaps show: the absolute error of the value estimate at the root node after the last simulation of each algorithm w.r.t. the respective optimal value (a), and w.r.t. the optimal value of UCT (b); regret at the root node (c).

2.0 (TENTS), 4.0, 8.0, 16.0. Figure 1 shows the convergence of the value estimate and regret at the root node of α-divergence in the Synthetic Tree environment. It shows that the error of the value estimate at the root node with respect to the optimal UCT value and the regularized value decrease when α increase, which matches our theoretical results in Theorem 3. Regarding the regret, the performance is different depending on different branching factors k and depth d, which illustrates that the value of α helps trade-off between exploration and exploitation depending on each environment. For example, with $k = 16, d = 2$, the regret is smaller when we increase the value of α, and the regret is smallest with $\alpha = 8.0$. When $k = 14, d = 3$, the regret is smaller when we increase the value of α and the regret performance is the best with $\alpha = 16.0$, and when $k = 16, d = 4$, the regret enjoys the best performance with $\alpha = 2.0$ (TENTS).

7.2 Entropy-regularized AlphaGo

Atari. We then evaluate the effectiveness of our $\alpha-$divergence regularization using Atari 2600 [Bellemare *et al.*, 2013] games. We examine entropy-based regularization MCTS algorithms, namely MENTS, RENTS with $\alpha = 1$, and TENTS with $\alpha = 2$. While Atari 2600 [Bellemare *et al.*, 2013] serves as a popular benchmark for assessing Deep RL methods [Mnih *et al.*, 2015; Van Hasselt *et al.*, 2016; Bellemare *et al.*, 2017], it has not been extensively explored within MCTS. We modify the standard AlphaGo algorithm, PUCT, in this experiment, using our regularized value-backup operator and policy selection. We use a pre-trained deep Q-network, using the same experimental setting of Mnih *et al.* [2015] as prior, to initialize the action value function of each node after the expansion step in the tree. For MENTS and TENTS, the initialization takes the form $Q_{\text{init}}(s,a) = (Q(s,a) - V(s))/\tau$, following Xiao *et al.* [2019]. In the case of RENTS, the initialization is $Q_{\text{init}}(s,a) = \log P_{\text{prior}}(a|s)) + (Q(s,a) - V(s))/\tau$, where P_{prior} denotes the Boltzmann distribution derived from the action values $Q(s,.)$ computed from the network. Each experimental run consists of 512 MCTS simulations. To find hyperparameters for each of our regularized MCTS algorithms, we perform a grid search over the temperature parameter τ with a range from 0.01 to 1. In addition, the discount factor is set to $\gamma = 0.99$, and for the PUCT algorithm, we use an exploration constant of $c = 0.1$. The performance of our regularized MCTS algorithms and the standard PUCT and MaxMCTS baselines is evaluated using 22-Atari games in terms of cumulative reward. The results in Table 1 show that our regularized methods outperform the baselines, with TENTS scoring the highest in all games. In particular, TENTS performs significantly better in games with high

Table 1: Average score in Atari over 100 seeds per game. Bold denotes no statistically significant difference to the highest mean (t-test, $p < 0.05$). Bottom row shows # no difference to highest mean.

	UCT	MaxMCTS	$\alpha = 1$(MENTS)	$\alpha = 1$(RENTS)	$\alpha = 2$(TENTS)
Alien	$\mathbf{1,486.80}$	$\mathbf{1,461.10}$	$\mathbf{1,508.60}$	$\mathbf{1,547.80}$	$\mathbf{1,568.60}$
Amidar	115.62	**124.92**	**123.30**	**125.58**	**121.84**
Asterix	4,855.00	$\mathbf{5,484.50}$	$\mathbf{5,576.00}$	$\mathbf{5,743.50}$	$\mathbf{5,647.00}$
Asteroids	873.40	899.60	1,414.70	1,486.40	$\mathbf{1,642.10}$
Atlantis	35,182.00	$\mathbf{35,720.00}$	$\mathbf{36,277.00}$	35,314.00	$\mathbf{35,756.00}$
BankHeist	475.50	458.60	**622.30**	**636.70**	**631.40**
BeamRider	$\mathbf{2,616.72}$	$\mathbf{2,661.30}$	$\mathbf{2,822.18}$	2,558.94	$\mathbf{2,804.88}$
Breakout	**303.04**	296.14	**309.03**	300.35	**316.68**
Centipede	1,782.18	1,728.69	$\mathbf{2,012.86}$	$\mathbf{2,253.42}$	$\mathbf{2,258.89}$
DemonAttack	579.90	640.80	$\mathbf{1,044.50}$	$\mathbf{1,124.70}$	$\mathbf{1,113.30}$
Enduro	**129.28**	124.20	128.79	**134.88**	**132.05**
Frostbite	1,244.00	1,332.10	$\mathbf{2,388.20}$	$\mathbf{2,369.80}$	$\mathbf{2,260.60}$
Gopher	3,348.40	3,303.00	$\mathbf{3,536.40}$	$\mathbf{3,372.80}$	$\mathbf{3,447.80}$
Hero	3,009.95	3,010.55	$\mathbf{3,044.55}$	$\mathbf{3,077.20}$	$\mathbf{3,074.00}$
MsPacman	1,940.20	1,907.10	2,018.30	$\mathbf{2,190.30}$	$\mathbf{2,094.40}$
Phoenix	2,747.30	2,626.60	3,098.30	2,582.30	$\mathbf{3,975.30}$
Qbert	7,987.25	8,033.50	8,051.25	8,254.00	$\mathbf{8,437.75}$
Robotank	**11.43**	11.00	**11.59**	**11.51**	**11.47**
Seaquest	$\mathbf{3,276.40}$	$\mathbf{3,217.20}$	$\mathbf{3,312.40}$	$\mathbf{3,345.20}$	$\mathbf{3,324.40}$
Solaris	895.00	923.20	$\mathbf{1,118.20}$	$\mathbf{1,115.00}$	$\mathbf{1,127.60}$
SpaceInvaders	778.45	**835.90**	**832.55**	**867.35**	**822.95**
WizardOfWor	685.00	666.00	$\mathbf{1,211.00}$	$\mathbf{1,241.00}$	$\mathbf{1,231.00}$
# Highest mean	6/22	7/22	17/22	16/22	**22/22**

branching factors, such as Asteroids and Phoenix, confirming the results of our experiment with synthetic trees and the theoretical advantages of TENTS in Section 6.

8 Conclusion

We introduced a unified view of the use of α-divergence in Monte-Carlo Tree Search(MCTS). We show that Power-UCT and the convex regularization in MCTS can be connected using α-divergence. In detail, the Power Mean backup operator used in Power-UCT can be derived as the solution of using α function as the probabilistic distance to replace the Eclipse distance used to calculate the average mean, in which the closed-form solution is the generalized power mean. Furthermore, entropic regularization in MCTS can be derived using α-function regularization. We provided the analysis of the regret bound with respect to the α parameter. We further analyzed the error bound between the regularized value estimate and the optimal regularized value at the root node. Empirical results in Synthetic Tree and Atari showed the effective balance between exploration and exploitation of α-divergence in MCTS with different values of α.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF) (Project: 01IS22078). This work was also funded by Hessian.ai through the project 'The Third Wave of Artificial Intelligence – 3AI' by the Ministry for Science and Arts of the state of Hessen.

References

Jacob D Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

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

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

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.

Richard Bellman. The theory of dynamic programming. Technical report, Rand corp santa monica ca, 1954.

Boris Belousov and Jan Peters. Entropic regularization of markov decision processes. *Entropy*, 21(7):674, 2019.

Aharon Ben-Tal, Abraham Charnes, and Marc Teboulle. Entropic means. *Journal of Mathematical Analysis and Applications*, 139(2):537–551, 1989.

Gang Chen, Yiming Peng, and Mengjie Zhang. Effective exploration for deep reinforcement learning via bootstrapped q-ensembles under tsallis entropy regularization, 2018.

Jienan Chen, Cong Zhang, Jinting Luo, Junfei Xie, and Yan Wan. Driving maneuvers prediction based autonomous driving control by deep monte carlo tree search. *IEEE transactions on vehicular technology*, 69(7):7146–7158, 2020.

Pierre-Arnaud Coquelin and Rémi Munos. Bandit algorithms for tree search, 2007.

Imre Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108, 1964.

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

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.

Niklas Funk, Georgia Chalvatzaki, Boris Belousov, and Jan Peters. Learn2assemble with structured representations and search for robotic architectural construction. In *5th Annual Conference on Robot Learning*, 2021.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169, 2019.

Jean-Bastien Grill, Florent Altché, Yunhao Tang, Thomas Hubert, Michal Valko, Ioannis Antonoglou, and Rémi Munos. Monte-carlo tree search as regularized policy optimization, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey 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.

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

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

Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, and Songhwai Oh. Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning, 2019.

Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, and Songhwai Oh. Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning. *CoRR*, abs/1902.00137, 2019.

Jincheng Mei, Chenjun Xiao, Ruitong Huang, Dale Schuurmans, and Martin Müller. On principled entropy exploration in policy optimization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3130–3136. AAAI Press, 2019.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pages 3462–3471, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

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

William H Montgomery and Sergey Levine. Guided policy search via approximate mirror descent. In *Advances in Neural Information Processing Systems*, pages 4008–4016, 2016.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *CoRR*, abs/2001.01866, 2020.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality, 2020.

Quan V Nguyen, Francis Colas, Emmanuel Vincent, and François Charpillet. Long-term robot motion planning for active sound source localization with monte carlo tree search. In *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, pages 61–65. IEEE, 2017.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention, 2017.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29:4026–4034, 2016.

Lacra Pavel. An extension of duality to a game-theoretic framework. *Automatica*, 43(2):226 – 237, 2007.

Gavin Adrian Rummery. *Problem solving with reinforcement learning*. PhD thesis, University of Cambridge Ph. D. dissertation, 1995.

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

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning, 2017.

Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and fenchel duality. *Advances in neural information processing systems*, 19:1265–1272, 2006.

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

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Fouad Sukkar, Graeme Best, Chanyeol Yoo, and Robert Fitch. Multi-robot region-of-interest reconstruction with dec-mcts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9101–9107. IEEE, 2019.

Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Gerald Tesauro, V T Rajan, and Richard Segal. Bayesian inference in monte-carlo tree search, 2012.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Tom Vodopivec, Spyridon Samothrakis, and Branko Ster. On monte carlo tree search and reinforcement learning. *Journal of Artificial Intelligence Research*, 60:881–936, 2017.

Nicola Catenacci Volpi, Yan Wu, and Dimitri Ognibene. Towards event-based mcts for autonomous cars. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 420–427. IEEE, 2017.

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.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for stochastic and adversarial bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 467–475. PMLR, 2019.

A Theoretical analysis of α-divergence in Monte-Carlo Tree Search

Theorem 1. *When $\alpha \in (0,1)$, the regret of α-E3W is*

$$\mathbb{E}[R_n] \leq \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha}-1) + n(2\tau)^{-1}|\mathcal{A}|^{\alpha} + \mathcal{O}(\frac{n}{\log n}).$$

Proof. We consider the generalized Tsallis Entropy $\Omega(\pi) = H_\alpha(\pi) = \frac{1}{1-\alpha}(1-\sum_i \pi^\alpha(a_i|s))$. According to Section 3 [Abernethy *et al.*, 2015], when $\alpha \in (0,1)$

$$\begin{aligned} \mathcal{D}_{\Omega^*}(\hat{V}_t(\cdot)+V(\cdot), \hat{V}_t(\cdot)) &\leq (\tau\alpha)^{-1}|\mathcal{A}|^{\alpha} \\ -\Omega(\hat{\pi}_n) &\leq \frac{1}{1-\alpha}(|\mathcal{A}|^{1-\alpha}-1). \end{aligned} \tag{31}$$

Then, for the generalized Tsallis Entropy, when $\alpha \in (0,1)$, as the result from Theorem 1 [Dam *et al.*, 2021] the regret is

$$\mathbb{E}[R_n] \leq \frac{\tau}{1-\alpha}(|\mathcal{A}|^{1-\alpha}-1) + n(\tau\alpha)^{-1}|\mathcal{A}|^{\alpha} + \mathcal{O}(\frac{n}{\log n}),$$

when $\alpha = 2$, which is the Tsallis entropy case we consider, according to Zimmert and Seldin [2019], By Taylor's theorem $\exists z \in \text{conv}(\hat{V}_t, \hat{V}_t + V)$, we have

$$\mathcal{D}_{\Omega^*}(\hat{V}_t(\cdot)+V(\cdot), \hat{V}_t(\cdot)) \leq \frac{1}{2}\left\langle V(\cdot), \nabla^2\Omega^*(z)V(\cdot)\right\rangle \leq \frac{|\mathcal{K}|}{2}.$$

So that when $\alpha = 2$, we have

$$\mathbb{E}[R_n] \leq \tau(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}) + \frac{n|\mathcal{K}|}{2} + \mathcal{O}(\frac{n}{\log n}).$$

when $\alpha = 1$, which is the maximum entropy case in our work, we derive.

$$\mathbb{E}[R_n] \leq \tau(\log|\mathcal{A}|) + \frac{n|\mathcal{A}|}{\tau} + \mathcal{O}(\frac{n}{\log n})$$

Finally, when the convex regularizer is relative entropy, One can simply write $KL(\pi_t||\pi_{t-1}) = -H(\pi_t) - \mathbb{E}_{\pi_t}\log\pi_{t-1}$, let $m = \min_a \pi_{t-1}(a|s)$, we have

$$\mathbb{E}[R_n] \leq \tau(\log|\mathcal{A}| - \frac{1}{m}) + \frac{n|\mathcal{A}|}{\tau} + \mathcal{O}(\frac{n}{\log n}).$$

□

Theorem 2. *When $\alpha \in (1,\infty)$, the regret of α-E3W is*

$$\mathbb{E}[R_n] \leq \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha}-1) + \frac{n|\mathcal{K}|}{2} + \mathcal{O}(\frac{n}{\log n}).$$

where $|\mathcal{K}|$ is the number of actions that are assigned non-zero probability in the policy at the root node.

Proof. From Theorem 1 [Dam *et al.*, 2021], we have

$$\mathbb{E}[R_n] \leq -\tau\Omega(\hat{\pi}) + \sum_{t=1}^{n}\mathcal{D}_{\Omega^*}(\hat{V}_t(\cdot)+V(\cdot), \hat{V}_t(\cdot)) + \mathcal{O}(\frac{n}{\log n}).$$

Here, $\Omega(\hat{\pi}) = H_\alpha(\hat{\pi}) = \frac{1}{\alpha(1-\alpha)}(1-\sum_i \hat{\pi}^\alpha(a_i|s))$. So as the result from equation 31, we have

$$-\Omega(\hat{\pi}_n) \leq \frac{1}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha}-1).$$

By Taylor's theorem $\exists z \in \text{conv}(\hat{V}_t, \hat{V}_t + V)$, we have

$$\mathcal{D}_{\Omega^*}(\hat{V}_t(\cdot) + V(\cdot), \hat{V}_t(\cdot)) \leq \frac{1}{2} \left\langle V(\cdot), \nabla^2 \Omega^*(z) V(\cdot) \right\rangle.$$

So that according to Equations (18), (19), (20), (21), we have

$$\mathcal{D}_{\Omega^*}(\hat{V}_t(\cdot) + V(\cdot), \hat{V}_t(\cdot)) \leq \frac{1}{2} \left\langle V(\cdot), \nabla^2 \Omega^*(z) V(\cdot) \right\rangle \leq \frac{|\mathcal{K}|}{2}.$$

so that

$$\mathbb{E}[R_n] \leq \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1) + \frac{n|\mathcal{K}|}{2} + \mathcal{O}(\frac{n}{\log n}).$$

□

We analyze the error of the regularized value estimate at the root node $n(s)$ w.r.t. the optimal value: $\varepsilon_\Omega = V_\Omega(s) - V^*(s)$. where Ω is the α-divergence regularizer H_α.

Theorem 3. *For any $\delta > 0$ and α-divergence regularizer H_α ($\alpha \neq 1, 2$), with some constant $C, \hat{C}$, with probability at least $1 - \delta$, ε_Ω satisfies*

$$-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1) \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}. \tag{32}$$

Proof. We have

$$0 \leq -\Omega(\hat{\pi}_n) \leq \frac{1}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1).$$

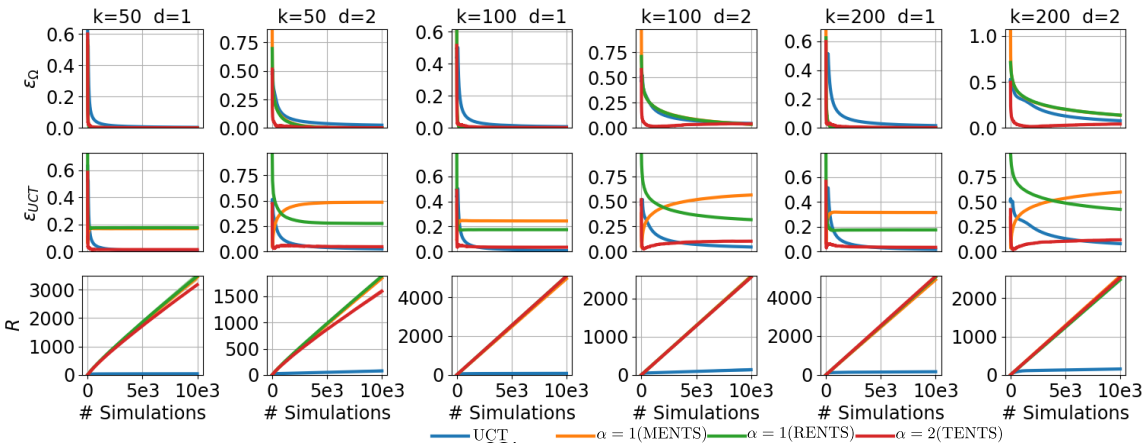
combine with Theorem 4 [Dam *et al.*, 2021] we will have

$$-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{\tau}{\alpha(1-\alpha)}(|\mathcal{A}|^{1-\alpha} - 1) \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}. \tag{33}$$

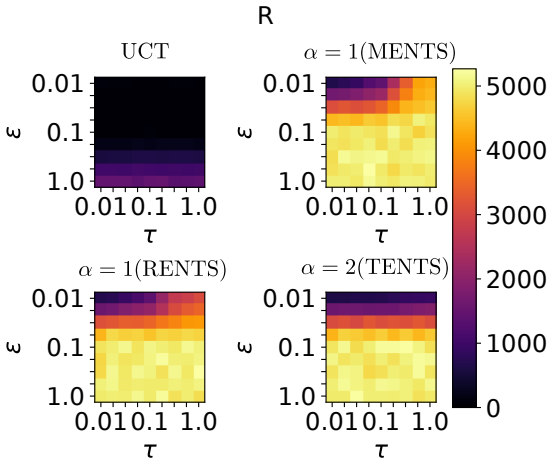
□

B Additional experiments in Synthetic Tree

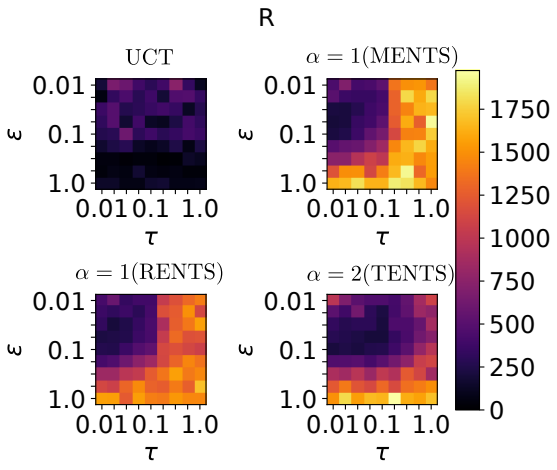
We perform additional experiments in the Synthetic Tree task with high branching factors to show the effectiveness of α divergence. We show in Figure 3(a) that TENTS outperforms other methods in high branching factor problems in terms of approximation error and regret. Additionally, we conduct a sensitivity analysis of each algorithm w.r.t. the values of the exploration coefficient ε and τ in two different trees in Figures 3(b) and 3(c). Our results demonstrate the superiority of TENTS in this toy problem, confirming our theoretical findings about the advantages of TENTS in problems with many actions in terms of approximation error and regret.



(a) Results in trees with high branching factor.



(b) $k = 100, d = 1$.



(c) $k = 8, d = 3$.

Figure 3: High branching factor trees (a), regret sensitivity study w.r.t. ε and τ (b, c).