CONVEX REGULARIZATION IN MONTE-CARLO TREE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Monte-Carlo planning and Reinforcement Learning (RL) are essential to sequential decision making. The recent AlphaGo and AlphaZero algorithms have shown how to successfully combine these two paradigms in order to solve large scale sequential decision problems. These methodologies exploit a variant of the well-known UCT algorithm to trade off exploitation of good actions and exploration of unvisited states, but their empirical success comes at the cost of poor sample-efficiency and high computation time. In this paper, we overcome these limitations by studying the benefit of convex regularization in Monte-Carlo Tree Search (MCTS) to efficiently drive exploration and to improve policy updates, as already observed in RL. First, we introduce a unifying theory on the use of generic convex regularizers in MCTS, deriving the first regret analysis of regularized MCTS and showing that it guarantees exponential convergence rate. Second, we exploit our theoretical framework to introduce novel regularized backup operators for MCTS, based on the relative entropy of the policy update, and on the Tsallis entropy of the policy. Finally, we show how our framework can easily be incorporated in AlphaGo and AlphaZero, and we empirically show the superiority of convex regularization w.r.t. representative baselines, on well-known RL problems and across several Atari games.

1 INTRODUCTION

Monte-Carlo Tree Search (MCTS) is a well-known algorithm to solve decision making problems through the combination of Monte-Carlo planning with an incremental tree structure (Coulom, 2006). Although standard MCTS is only suitable for problems with discrete state and action spaces, recent advances have shown how to enable the use of MCTS in continuous problems (Silver et al., 2016; Yee et al., 2016). Most remarkably, AlphaGo (Silver et al., 2016) and AlphaZero (Silver et al., 2017b;a) couple MCTS with neural networks trained using Reinforcement Learning (RL) (Sutton & Barto, 1998) methods, e.g. Deep Q-Learning (Mnih et al., 2015), to speed up learning of large scale problems with continuous state space. In particular, a neural network is used to compute value function estimates of states as a replacement of time-consuming Monte-Carlo rollouts, and another neural network is used to estimate policies as a probability prior for the therein introduced PUCT action selection method, a variant of well-known UCT sampling strategy commonly used in MCTS for exploration (Kocsis et al., 2006). Despite AlphaGo and AlphaZero achieving state-of-the-art performance in games with high branching factor like Go (Silver et al., 2016) and Chess (Silver et al., 2017a), both methods suffer from poor sample-efficiency, mostly due to the polynomial convergence rate of PUCT (Xiao et al., 2019). This problem, combined with the high computational time to evaluate the deep neural networks, significantly hinder the applicability of both methodologies.

In this paper, we provide a unified theory of the use of convex regularization in MCTS, that proved to be an efficient solution for driving exploration and stabilizing learning in RL (Schulman et al., 2015; 2017a; Haarnoja et al., 2018). In particular, we show how a regularized objective function in MCTS can be seen as an instance of the Legendre-Fenchel transform, similarly to previous findings in regularized Markov Decision Processes (Mensch & Blondel, 2018; Geist et al., 2019). Establishing our theoretical framework, we are able to derive the first regret analysis of regularized MCTS, and to prove that a generic convex regularizer guarantees an exponential convergence rate to the solution of the regularized objective function, which improves on the polynomial rate of PUCT. These results provide a theoretical ground for the use of arbitrary entropy-based regularizers in MCTS,

until now limited to maximum entropy (Xiao et al., 2019), among which we specifically study the relative entropy of policy updates, drawing on similarities with trust-region and proximal methods in RL (Schulman et al., 2015; 2017b), and the Tsallis entropy, used for enforcing the learning of sparse policies (Lee et al., 2018). Finally, we empirically evaluate the proposed operators in AlphaGo and AlphaZero on problems of increasing complexity, from classic RL problems to an extensive analysis of Atari games, confirming the benefit of our novel operators compared to maximum entropy and, in general, the superiority of convex regularization in MCTS w.r.t. classic methods.

2 Preliminaries

2.1 Markov Decision Processes

We consider the classical definition of a finite-horizon Markov Decision Process (MDP) as a 5-tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition kernel, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi \in \Pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a probability distribution of the event of executing an action a in a state s. A policy π induces a value function corresponding to the expected cumulative discounted reward collected by the agent when executing action a in state s, and following the policy π thereafter: $Q^\pi(s, a) \triangleq \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{i+k+1} | s_i = s, a_i = a, \pi\right]$, where r_{i+1} is the reward obtained after the i-th transition. An MDP is solved finding the optimal policy π^*, which is the policy that maximizes the expected cumulative discounted reward. The optimal policy corresponds to the one satisfying the optimal Bellman equation (Bellman, 1954) $Q^*(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q^*(s', a')\right] ds'$, and is the fixed point of the optimal Bellman operator $\mathcal{T}^* Q(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \max_{a'} Q(s', a')\right] ds'$. Additionally, we define the Bellman operator under the policy π as $\mathcal{T}_\pi Q(s, a) \triangleq \int_{\mathcal{S}} \mathcal{P}(s'|s, a) \left[\mathcal{R}(s, a, s') + \gamma \int_{\mathcal{A}} \pi(a'|s') Q(s', a') da'\right] ds'$, the optimal value function $V^*(s) \triangleq \max_{a \in \mathcal{A}} Q^*(s, a)$, and the value function under the policy π as $V^\pi(s) \triangleq \max_{a \in \mathcal{A}} Q^\pi(s, a)$.

2.2 Monte-Carlo Tree Search and Upper Confidence bounds for Trees

Monte-Carlo Tree Search (MCTS) is a planning strategy based on a combination of Monte-Carlo sampling and tree search to solve MDPs. MCTS builds a tree where the nodes are the visited states of the MDP, and the edges are the actions executed in each state. MCTS converges to the optimal policy (Kocsis et al., 2006; Xiao et al., 2019), iterating over a loop composed of four steps:

1. **Selection:** starting from the root node, a *tree-policy* is executed to navigate the tree until a node with unvisited children, i.e. expandable node, is reached;
2. **Expansion:** the reached node is expanded according to the tree policy;
3. **Simulation:** run a rollout, e.g. Monte-Carlo simulation, from the visited child of the current node to the end of the episode;
4. **Backup:** use the collected reward to update the action-values $Q(\cdot)$ of the nodes visited in the trajectory from the root node to the expanded 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. The Upper Confidence bounds for Trees (UCT) sampling strategy (Kocsis et al., 2006) extends the use of the well-known UCB1 sampling strategy for multi-armed bandits (Auer et al., 2002), to MCTS. Considering each node corresponding to a state $s \in \mathcal{S}$ as a different bandit problem, UCT selects an action $a \in \mathcal{A}$ applying an upper bound to the action-value function

$$\text{UCT}(s, a) = Q(s, a) + C \sqrt{\frac{\log N(s)}{N(s, a)}}, \tag{1}$$

where $N(s, a)$ is the number of executions of action a in state s, $N(s) = \sum_a N(s, a)$, and C is a constant parameter to tune exploration. UCT asymptotically converges to the optimal action-value function Q^*, for all states and actions, with the probability of executing a suboptimal action at the root node approaching 0 with a polynomial rate $O(\frac{1}{t})$, for a simulation budget t (Kocsis et al., 2006; Xiao et al., 2019).

3 REGULARIZED MONTE-CARLO TREE SEARCH

The success of RL methods based on entropy regularization comes from their ability to achieve state-of-the-art performance in decision making and control problems, while enjoying theoretical guarantees and ease of implementation (Haarnoja et al., 2018; Schulman et al., 2015; Lee et al., 2018). However, the use of entropy regularization is MCTS is still mostly unexplored, although its advantageous exploration and value function estimation would be desirable to reduce the detrimental effect of high-branching factor in AlphaGo and AlphaZero. To the best of our knowledge, the MENTS algorithm (Xiao et al., 2019) is the first and only method to combine MCTS and entropy regularization. In particular, MENTS uses a maximum entropy regularizer in AlphaGo, proving an exponential convergence rate to the solution of the respective softmax objective function and achieving state-of-the-art performance in some Atari games (Bellemare et al., 2013). In the following, motivated by the success in RL and the promising results of MENTS, we derive a unified theory of regularization in MCTS based on the Legendre-Fenchel transform (Geist et al., 2019), that generalizes the use of maximum entropy of MENTS to an arbitrary convex regularizer. Notably, our theoretical framework enables to rigorously motivate the advantages of using maximum entropy and other entropy-based regularizers, such as relative entropy or Tsallis entropy, drawing connections with their RL counterparts TRPO (Schulman et al., 2015) and Sparse DQN (Lee et al., 2018), as MENTS does with Soft Actor-Critic (SAC) (Haarnoja et al., 2018).

3.1 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 $Q_s = Q(s, \cdot) \in \mathbb{R}^{\mathcal{A}}$, the Legendre-Fenchel transform (or convex conjugate) of Ω is $\Omega^* : \mathbb{R}^{\mathcal{A}} \to \mathbb{R}$, defined as:

$$\Omega^*(Q_s) \triangleq \max_{\pi_s \in \Pi_s} \mathcal{T}_{\pi_s} Q_s - \tau \Omega(\pi_s), \tag{2}$$

where the temperature τ specifies the strength of regularization. Among the several properties of the Legendre-Fenchel transform, we use the following (Mensch & Blondel, 2018; Geist et al., 2019).

Proposition 1 *Let Ω be strongly convex.*

- *Unique maximizing argument: $\nabla\Omega^*$ is Lipschitz and satisfies*
$$\nabla\Omega^*(Q_s) = \arg\max_{\pi_s \in \Pi_s} \mathcal{T}_{\pi_s} Q_s - \tau \Omega(\pi_s). \tag{3}$$
- *Boundedness: if there are constants L_Ω and U_Ω such that for all $\pi_s \in \Pi_s$, we have $L_\Omega \leq \Omega(\pi_s) \leq U_\Omega$, then*
$$\max_{a \in \mathcal{A}} Q_s(a) - \tau U_\Omega \leq \Omega^*(Q_s) \leq \max_{a \in \mathcal{A}} Q_s(a) - \tau L_\Omega. \tag{4}$$
- *Contraction: for any $Q_1, Q_2 \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$*
$$\| \Omega^*(Q_1) - \Omega^*(Q_2) \|_\infty \leq \gamma \| Q_1 - Q_2 \|_\infty . \tag{5}$$

Although the Legendre-Fenchel transform Ω^* applies to every strongly convex function, for the purpose of this work we only consider a representative set of entropic regularizers.

3.2 REGULARIZED BACKUP AND TREE POLICY

In MCTS, each node of the tree represents a state $s \in \mathcal{S}$ and contains a visitation count $N(s, a)$. Given a trajectory, we define $n(s_T)$ as the leaf node corresponding to the reached state s_T. Let $s_0, a_0, s_1, a_1 ..., s_T$ be the state action trajectory in a simulation, where $n(s_T)$ is a leaf node of $\mathcal{T}$. Whenever a node $n(s_T)$ is expanded, the respective action values (Equation 6) are initialized as $Q_\Omega(s_T, a) = 0$, and $N(s_T, a) = 0$ for all $a \in \mathcal{A}$. For all nodes in the trajectory, the visitation count is updated by $N(s_t, a_t) = N(s_t, a_t) + 1$, and the action-values by

$$Q_\Omega(s_t, a_t) = \begin{cases} r(s_t, a_t) + \gamma\rho & \text{if } t = T \\ r(s_t, a_t) + \gamma\Omega^*(Q_\Omega(s_{t+1})/\tau)) & \text{if } t < T \end{cases} \tag{6}$$

where $Q_\Omega(s_{t+1}) \in \mathbb{R}^{\mathcal{A}}$ with components $Q_\Omega(s_{t+1}, a), \forall a \in \mathcal{A}$, and ρ is an estimate returned from an evaluation function computed in s_T, e.g. a discounted cumulative reward averaged over multiple rollouts, or the value-function of node $n(s_{T+1})$ returned by a value-function approximator, e.g. a neural network pretrained with deep Q-learning (Mnih et al., 2015), as done in (Silver et al., 2016; Xiao et al., 2019). We revisit the E2W sampling strategy limited to maximum entropy regularization (Xiao et al., 2019) and, through the use of the convex conjugate in Equation (6), we derive a novel sampling strategy that generalizes to any convex regularizer

$$\pi_t(a_t|s_t) = (1 - \lambda_{s_t})\nabla\Omega^*(Q_\Omega(s_t)/\tau)(a_t) + \frac{\lambda_{s_t}}{|\mathcal{A}|}, \tag{7}$$

where $\lambda_{s_t} = {}^{\epsilon|\mathcal{A}|}/_{\log(\sum_a N(s_t,a)+1)}$ with $\epsilon > 0$ as an exploration parameter, and $\nabla\Omega^*$ depends on the measure in use (see Table 1 for maximum, relative, and Tsallis entropy). We call this sampling strategy *Extended Empirical Exponential Weight* (E3W) to highlight the extension of E2W from maximum entropy to a generic convex regularizer.

3.3 Convergence rate to regularized objective

We show that the regularized value V_Ω can be effectively estimated at the root state $s \in \mathcal{S}$, with the assumption that each node in the tree has a σ^2-subgaussian distribution. This result extends the analysis provided in (Xiao et al., 2019), which is limited to the use of maximum entropy.

Theorem 1 *At the root node s where $N(s)$ is the number of visitations, with $\epsilon > 0$, $V_\Omega(s)$ is the estimated value, with constant C and $\hat{C}$, we have*

$$\mathbb{P}(|V_\Omega(s) - V^*_\Omega(s)| > \epsilon) \leq C\exp\{-\frac{N(s)\epsilon}{\hat{C}\sigma(\log(2 + N(s)))^2}\}, \tag{8}$$

where $V_\Omega(s) = \Omega^*(Q_s)$ and $V^*_\Omega(s) = \Omega^*(Q^*_s)$. From this theorem, we obtain that the convergence rate of choosing the best action at the root node, when using the E3W strategy, is exponential.

Theorem 2 *Let a_t be the action returned by E3W at step t. For large enough t and constants $C, \hat{C}$*

$$\mathbb{P}(a_t \neq a^*) \leq Ct\exp\{-\frac{t}{\hat{C}\sigma(\log(t))^3}\}. \tag{9}$$

4 Entropy-regularization backup operators

From the introduction of a unified view of generic strongly convex regularizers as backup operators in MCTS, we narrow the analysis to entropy-based regularizers. For each entropy function, Table 1 shows the Legendre-Fenchel transform and the maximizing argument, which can be respectively replaced in our backup operation (Equation 6) and sampling strategy E3W (Equation 7). Using maximum entropy retrieves the maximum entropy MCTS problem introduced in the MENTS algorithm (Xiao et al., 2019). This approach closely resembles the maximum entropy RL framework used to encourage exploration (Haarnoja et al., 2018; Schulman et al., 2017a). We introduce two novel MCTS algorithms based on the minimization of relative entropy of the policy update, inspired by trust-region (Schulman et al., 2015) and proximal optimization methods (Schulman et al., 2017b) in RL, and on the maximization of Tsallis entropy, which has been more recently introduced in RL as an effective solution to enforce the learning of sparse policies (Lee et al., 2018). We call these algorithms RENTS and TENTS. Contrary to maximum and relative entropy, the definition of the Legendre-Fenchel and maximizing argument of Tsallis entropy is non-trivial, being

$$\Omega^*(Q_t) = \tau \cdot \text{spmax}(Q_t(s, \cdot)/\tau), \tag{10}$$

$$\nabla\Omega^*(Q_t) = \max\left(\frac{Q_t(s,a)}{\tau} - \frac{\sum_{a\in\mathcal{K}} Q_t(s,a)/\tau - 1}{|\mathcal{K}|}, 0\right), \tag{11}$$

where spmax is defined for any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as

$$\text{spmax}(f(s,\cdot)) \triangleq \sum_a \left(\frac{f(s,a)^2}{2} - \frac{(\sum_{a\in\mathcal{K}} f(s,a) - 1)^2}{2|\mathcal{K}|^2}\right) + \frac{1}{2}, \tag{12}$$

and $\mathcal{K}$ is the set of actions that satisfy $1 + if(s, a_i) > \sum_{i=1}^{j} f(s, a_i)$, with a_i indicating the action with the i-th largest value of $f(s, a)$.

Table 1: List of entropy regularizers with Legendre-Fenchel transforms and maximizing arguments.

Entropy	**Regularizer** $\Omega(\pi_s)$	**Legendre-Fenchel** $\Omega^*(Q_s)$	**Max argument** $\nabla\Omega^*(Q_s)$
Maximum	$\sum_a \pi(a\|s)\log\pi(a\|s)$	$\log\sum_a e^{Q(s,a)}$	$\frac{e^{Q(s,a)}}{\sum_b e^{Q(s,b)}}$
Relative	$\mathrm{D_{KL}}(\pi_t(a\|s)\|\|\pi_{t-1}(a\|s))$	$\log\sum_a \pi_{t-1}(a\|s)e^{Q_t(s,a)}$	$\frac{\pi_{t-1}(a\|s)e^{Q_t(s,a)}}{\sum_b \pi_{t-1}(b\|s)e^{Q_t(s,b)}}$
Tsallis	$\frac{1}{2}(\parallel \pi(a\|s) \parallel_2^2 -1)$	Equation (10)	Equation (11)

4.1 Regret analysis

Let each node i in the tree be assigned with a random variable X_i, with mean value μ_i. At the root node, the quantities related to the optimal branch are denoted by $*$, e.g. mean value μ^*. At each timestep n, the mean value of variable X_i is μ_{i_n}. The pseudo-regret (Coquelin & Munos, 2007) at timestep n is defined as $R_n^{\text{UCT}} = n\mu^* - \sum_{t=1}^n \mu_{i_t}$. The regret of E3W at the root node $n(s)$ of the tree considering the value-function $V(\cdot)$ as the mean value μ is

$$R_n = nV^* - \sum_{t=1}^n V_{i_t} = nV^* - \sum_{t=1}^n \mathbb{I}(i_t = i)V_{i_t} = nV^* - \sum_i V_i \sum_{t=1}^n \hat{\pi}_t(a_i|s), \tag{13}$$

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

Theorem 3 *Consider an E3W policy applied to the tree. Let* $\kappa_i = \nabla\Omega^*(a_i|s) + \sqrt{\hat{C}\sigma^2 \log\frac{C}{\delta}/2n}$, *where* $\nabla\Omega^*(.|s)$ *is the policy with respect to the mean value vector* $V(\cdot)$ *at the root node* s. *For any* $\delta > 0$, *with probability at least* $1-\delta$, $\exists$ *constant* $L, p, C, \hat{C}$ *so that the pseudo regret* R_n *satisfies*

$$nV^* - n\sum_i V_i\Big(\kappa_i - \frac{L}{p}\big(\frac{\tau(U_\Omega - L_\Omega)}{1-\gamma}\big)\Big) \leq R_n \leq nV^* - n\sum_i V_i\Big(\kappa_i + \frac{L}{p}\big(\frac{\tau(U_\Omega - L_\Omega)}{1-\gamma}\big)\Big).$$

This theorem provides bounds for the regret of E3W using a generic convex regularizer Ω; thus, we can easily retrieve from it the regret bound for each entropy regularizer. Let $m = \min_a \nabla\Omega^*(a|s)$.

Corollary 1 *Maximum entropy:*
$nV^* - n\sum_i V_i\Big(\kappa_i + L\big(\frac{\tau\log|A|}{1-\gamma}\big)\Big) \leq R_n \leq nV^* - n\sum_i V_i\Big(\kappa_i - L\big(\frac{\tau\log|A|}{1-\gamma}\big)\Big).$

Corollary 2 *Relative entropy:*
$nV^* - n\sum_i V_i\Big(\kappa_i + L\big(\frac{\tau(\log|A| - \frac{1}{m})}{1-\gamma}\big)\Big) \leq R_n \leq nV^* - n\sum_i V_i\Big(\kappa_i - L\big(\frac{\tau(\log|A| - \frac{1}{m})}{1-\gamma}\big)\Big).$

Corollary 3 *Tsallis entropy:*
$nV^* - n\sum_i V_i\Big(\kappa_i + \frac{L}{2}\big(\frac{|A|-1}{2|A|}\frac{\tau}{1-\gamma}\big)\Big) \leq R_n \leq nV^* - n\sum_i V_i\Big(\kappa_i - \frac{L}{2}\big(\frac{|A|-1}{2|A|}\frac{\tau}{1-\gamma}\big)\Big).$

Remarks. The regret bound of UCT and its variance have already been analyzed for non-regularized MCTS with binary tree (Coquelin & Munos, 2007). On the contrary, our regret bound analysis in Theorem 3 applies to generic regularized MCTS. From the specialized bounds in the corollaries, we observe that the maximum and relative entropy share similar results, although the bounds for relative entropy are slightly smaller due to $\frac{1}{m}$. Remarkably, the bounds for Tsallis entropy become tighter for increasing number of actions, which translates in limited regret in problems with high branching factor. This result establishes the advantage of Tsallis entropy in complex problems w.r.t. to other entropy regularizers, as empirically confirmed by the positive results in several Atari games described in Section 5.

4.2 ERROR ANALYSIS

We analyse the error of the regularized value estimate at the root node $n(s)$ w.r.t. the optimal value: $\varepsilon_\Omega = V_\Omega(s) - V^*(s)$.

Theorem 4 *For any $\delta > 0$ and generic convex regularizer Ω, with some constant $C, \hat{C}$, with probability at least $1-\delta$, ε_Ω satisfies*

$$-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{\tau(U_\Omega - L_\Omega)}{1-\gamma} \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}. \tag{14}$$

To give a better understanding of the effect of each entropy regularizer in Table 1, we specialize the bound in Equation 14 to each of them. From (Lee et al., 2018), we know that for maximum entropy $\Omega(\pi_t) = \sum_a \pi_t \log \pi_t$, we have $-\log|\mathcal{A}| \leq \Omega(\pi_t) \leq 0$; for relative entropy $\Omega(\pi_t) = \text{KL}(\pi_t||\pi_{t-1})$, if we define $m = \min_a \pi_{t-1}(a|s)$, then we can derive $0 \leq \Omega(\pi_t) \leq -\log|\mathcal{A}| + \log\frac{1}{m}$; and for Tsallis entropy $\Omega(\pi_t) = \frac{1}{2}(\parallel \pi_t \parallel_2^2 - 1)$, we have $-\frac{|\mathcal{A}|-1}{2|\mathcal{A}|} \leq \Omega(\pi_t) \leq 0$. Then,

Corollary 4 *maximum entropy error:* $-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{\tau \log|A|}{1-\gamma} \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}$.

Corollary 5 *relative entropy error:* $-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{\tau(\log|A| - \log\frac{1}{m})}{1-\gamma} \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}$.

Corollary 6 *Tsallis entropy error:* $-\sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}} - \frac{|A|-1}{2|A|}\frac{\tau}{1-\gamma} \leq \varepsilon_\Omega \leq \sqrt{\frac{\hat{C}\sigma^2 \log \frac{C}{\delta}}{2N(s)}}$.

These results show that when the number of actions $|\mathcal{A}|$ is large, TENTS enjoys smallest error; moreover, we also see that the error at the root node of RENTS is always smaller than for MENTS.

5 EMPIRICAL EVALUATION

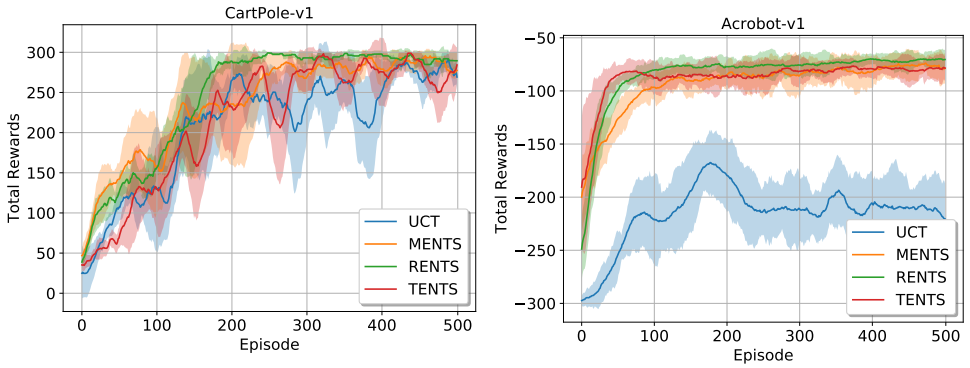
In this section, we empirically evaluate the benefits of the proposed entropic MCTS regularizers. We compare to AlphaGo and AlphaZero (Silver et al., 2016; 2017a), recently introduced to enable MCTS to solve large scale problems with high branching factor. Our implementation is a simplified version of the original algorithms, where we remove various tricks in favour of better interpretability. For the same reason, we do not compare with the most recent and state-of-the-art variant of AlphaZero known as MuZero (Schrittwieser et al., 2019), as this is a slightly different solution highly tuned to maximize performance, and a detailed description of its implementation is not available.

5.1 ENTROPY-REGULARIZED ALPHAZERO

In its standard form, AlphaZero (Silver et al., 2017a) uses the PUCT sampling strategy, a variant of UCT (Kocsis et al., 2006) that samples actions according to the policy

$$PUCT(s,a) = Q(s,a) + cP(s,a)\frac{\sqrt{N(s)}}{1+N(s,a)}, \tag{15}$$

where P is a prior probability on action selection, and c is an exploration constant. A value network and a policy network are used to compute, respectively, the action-value function Q and the prior policy P. We use a single neural network, with 2 hidden layers composed of 128 ELU units, and two output layer respectively for the action-value function and the policy. We run 500 AlphaZero episodes, where each episode is composed of 300 steps. A step consists of running 32 MCTS simulations from the root node, as defined in Section 2, using the action-value function computed by the value network instead of using Monte-Carlo rollouts. At the end of each cycle, the average action-value of the root node is computed and stored, the tree is expanded using the given sampling



(a) Cartpole. (b) Acrobot.

Figure 1: Cumulative rewards of AlphaZero with UCT and entropy-based operators, in CartPole (a) and Acrobot (b). Results are averaged over 5 and 10 seeds and show 95% confidence intervals.

strategy, and the root node is updated with the reached node. At the end of the episode, a minibatch of 32 samples is built from the 300 stored action-values, and the network is trained with one step of gradient descent using RMSProp with learning rate 0.001. The entropy-regularized variants of AlphaZero can be simply derived replacing the average backup operator, with the desired entropy function, and replacing PUCT with E3W using the respective maximizing argument and $\epsilon = 0.1$.

Cartpole and Acrobot. Figure 1 shows the cumulative reward of standard AlphaZero based on PUCT, and the three entropy-regularized variants, on the Cartpole and Acrobot discrete control problems (Brockman et al., 2016). While standard AlphaZero is clearly the worst approach for convergence and stability, the entropy-based variants behave differently according to the problem. First, although not significantly superior, RENTS exhibits the most stable learning and faster convergence, confirming the benefit of relative entropy in control problems as already known for trust-region methods in RL (Schulman et al., 2015). Second, considering the small number of discrete actions in the problems, TENTS cannot benefit from the learning of sparse policies and shows slightly unstable learning in Cartpole, even though the overall performance is satisfying in both problems. Last, MENTS solves the problems slightly slower than RENTS, but reaches the same final performance. Although the results on these simple problems are not conclusive to assert the superiority of one method over the other, they definitely confirm the advantage of regularization in MCTS, and hint at the benefit of the use of relative entropy in control problems. Further analysis on more complex control problems will be desirable (e.g. MuJoCo (Todorov et al., 2012)), but the need to account for continuous actions, a non-trivial setting for MCTS, makes it out of the scope of this paper.

5.2 Entropy-regularized AlphaGo

The learning time of AlphaZero can be slow in problems with high branching factor, due to the need of a large number of MCTS simulations for obtaining good estimates of the randomly initialized action-values. To overcome this problem, AlphaGo (Silver et al., 2016) initializes the action-values using the values retrieved from a pretrained network, which is kept fixed during the training.

Atari. Atari 2600 (Bellemare et al., 2013) is a popular benchmark for testing deep RL methodologies (Mnih et al., 2015; Van Hasselt et al., 2016; Bellemare et al., 2017) but still relatively disregarded in MCTS. We use the Atari games in an extensive empirical comparison of standard AlphaGo based on PUCT, with AlphaGo using the described regularized backup operators. We use a Deep Q-Network, pretrained using the same experimental setting of Mnih et al. (2015), to initialize the action-value function of each node after expansion as $Q_{init}(s,a) = (Q(s,a) - V(s))/\tau$, for MENTS and TENTS, as done in Xiao et al. (2019). For RENTS we init $Q_{init}(s,a) = \log P_{\text{prior}}(a|s)) + (Q(s,a) - V(s))/\tau$, where P_{prior} is the Boltzmann distribution induced by action-values $Q(s,.)$ computed from the network. Each experimental run consists of 512 MCTS simulations. The temperature τ is optimized for each algorithm and game via grid-search between 0.01 and 1. The discount factor is $\gamma = 0.99$, and for PUCT the exploration constant is $c = 0.1$.

Table 2: Cumulative reward in Atari over 100 seeds. 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	MENTS	RENTS	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	$\mathbf{124.92}$	$\mathbf{123.30}$	$\mathbf{125.58}$	$\mathbf{121.84}$
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	$\mathbf{622.30}$	$\mathbf{636.70}$	$\mathbf{631.40}$
BeamRider	$\mathbf{2,616.72}$	$\mathbf{2,661.30}$	$\mathbf{2,822.18}$	$2,558.94$	$\mathbf{2,804.88}$
Breakout	$\mathbf{303.04}$	296.14	$\mathbf{309.03}$	300.35	$\mathbf{316.68}$
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	$\mathbf{129.28}$	124.20	128.79	$\mathbf{134.88}$	$\mathbf{132.05}$
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	$\mathbf{11.43}$	11.00	$\mathbf{11.59}$	$\mathbf{11.51}$	$\mathbf{11.47}$
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	$\mathbf{835.90}$	$\mathbf{832.55}$	$\mathbf{867.35}$	$\mathbf{822.95}$
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**

Table 2 shows the performance, in terms of cumulative reward, of standard AlphaGo with PUCT and our three regularized versions, on 22 Atari games. Moreover, we test also AlphaGo using the MaxMCTS backup (Khandelwal et al., 2016) for further comparison with classic baselines. We observe that regularized MCTS dominates the other baselines, in particular TENTS achieves the highest results in all the 22 games, showing that sparse policies are more effective in Atari. This can also be explained by Corollary 6 which shows that Tsallis entropy can lead to a lower error at the MCTS root node even with a high number of actions compared to relative or maximum entropy.

6 Conclusion

We introduced a theory of convex regularization in Monte-Carlo Tree Search (MCTS) based on the Legendre-Fenchel transform. Exploiting this theoretical framework, we studied the regret of MCTS when using a generic strongly convex regularizer, and we proved that it has an exponential convergence rate. We use these results to motivate the use of entropy regularization in MCTS, in particular considering maximum, relative, and Tsallis entropy. Finally, we test regularized MCTS algorithms in discrete control problems and Atari games, showing its advantages over other methods.

REFERENCES

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*, pp. 449–458. JMLR. org, 2017.

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

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Pierre-Arnaud Coquelin and Rémi Munos. Bandit algorithms for tree search. *arXiv preprint cs/0703062*, 2007.

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

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

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*, pp. 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, 2006.

Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1466–1473, 2018.

Arthur Mensch and Mathieu Blondel. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, pp. 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.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model, 2019.

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

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

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, 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, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017b.

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

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

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

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

Timothy Yee, Viliam Lisỳ, Michael H Bowling, and S Kambhampati. Monte carlo tree search in continuous action spaces with execution uncertainty. In *IJCAI*, pp. 690–697, 2016.