SOFT MELLOWMAX MONTE CARLO PLANNING

Danilo Vucetic[*1,2], **Gauthier Gidel**[1,2,3]

[1]Université de Montréal [2]Mila [3]CIFAR

ABSTRACT

Soft mellowmax (SMM) recently emerged as an alternative operator in Q-learning, achieving impressive performance in games and scientific discovery tasks. Despite SMM's ability to achieve high returns and its enticing robustness, diversity, and sample efficiency characteristics, SMM has not yet been translated into a Monte Carlo tree search algorithm. To address this gap, a soft mellowmax-based Monte Carlo tree search algorithm, SMM-TS, is proposed and theoretically justified. It is empirically demonstrated that SMM-TS converges significantly faster than other tree search methods in synthetic environments, while maintaining competitive performance in games. The fast convergence of SMM-TS makes recursive self-improvement loops more scalable, while the stability gained via planning and the robustness of the operator make SMM-TS more practical for agents operating in uncertain and changing environments.

1 INTRODUCTION

Artificial intelligence algorithms based on Monte Carlo tree search (MCTS), such as AlphaZero, have achieved state-of-the-art performance in reinforcement learning (RL) tasks (Silver et al., 2016; 2017; 2018). By using additional computational resources at each decision step, MCTS algorithms can generate better policy or value estimates for an agent (Kocsis & Szepesvári, 2006; Xiao et al., 2019; Danihelka et al., 2022; Dam et al., 2021). The convergence speed, stability, and robustness of an MCTS algorithm are thus consequential in the practicality and scalability of the self-improvement loop.

However, MCTS algorithms like PUCT (AlphaZero's variant of UCT) converge to the optimal policies and values of standard RL, which can be brittle to changing environments (Auer et al., 2002; Kocsis & Szepesvári, 2006; Rosin, 2011). Robustness to perturbations in the environment is an essential component for agents facing real-world deployment (Eysenbach & Levine, 2022; Zhai et al., 2022). Regularised RL offers a compelling solution: by regularising the RL objective, the resulting optimal policy is robust to some reward and dynamics perturbations, while also improving sample efficiency and performance (Husain et al., 2021; Eysenbach & Levine, 2022; Haarnoja et al., 2018; Asadi & Littman, 2017; Kim et al., 2019; Gan et al., 2021). These benefits have motivated the development of convex regularised tree search (CRTS), which has impressive convergence speed and policy performance, but is limited by strict convexity assumptions Xiao et al. (2019); Dam et al. (2021); Painter et al. (2023).

This paper introduces soft mellowmax Monte Carlo tree search (SMM-TS), a novel planning algorithm that integrates the soft mellowmax operator into the CRTS framework. Soft mellowmax (SMM) is a non-convex operator which has stronger robustness guarantees than entropy or KL regularisation, while achieving better performance and sample efficiency (Gan et al., 2021; Jiralerspong et al., 2025). Soft mellowmax-based tree search presents unique theoretical challenges that we address through a new proof of convergence derived from the smoothness of the operator.

We argue that SMM-TS is particularly well-suited for recursive self-improvement for three reasons. First, our theoretical analysis proves that SMM-TS converges in policy and value estimates. Second, we empirically demonstrate that SMM-TS converges to optimal value estimates significantly faster than existing methods in synthetic environments. In a recursive loop, faster convergence implies that high-quality targets can be generated with less compute, accelerating the overall learning process.

[*]Correspondence to danilo.vucetic@mila.quebec

Finally, we show that SMM-TS is robust to initialization errors, a critical property when the prior policy and value are constantly shifting neural networks that may be biased or noisy during the early stages of training. By stabilizing and accelerating the planning phase, SMM-TS offers a more reliable engine for agents designed to recursively improve their own intelligence.

2 Background and related work

2.1 Markov decision processes and reinforcement learning

Markov decision processes (MDPs) are the standard problem setting for reinforcement learning (RL). An MDP is described by a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma, T)$ containing the state space, action space, Markovian transition function $p(s'|s,a)$, reward function $r(s,a)$, discount factor $\gamma \in [0,1)$, and a horizon T respectively (Sutton et al., 2018). It is common in tree search literature to assume a deterministic transition function (Silver et al., 2018; Xiao et al., 2019; Dam et al., 2021), as doing so removes so-called "chance nodes" from the tree search architecture and added theoretical difficulties. The goal in RL is to find a state-action trajectory, or policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, that maximises the value function, V^π: the expected sum of discounted rewards across the horizon (Puterman, 1994a;b). $V^\star(s)$ is called the optimal value function.

$$V^\star(s) = \max_\pi V^\pi(s) = \max_\pi \mathop{\mathbb{E}}_{\substack{a_t \sim \pi \\ s_{t+1} \sim p(s_{t+1}|s_t, a_t)}} \left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) \middle| s_0 = s \right]; \forall s \in \mathcal{S}$$

From the value function, the Q-value function can be derived, yielding the important relation of value functions as expected Q-value functions.

$$Q^\pi(s,a) = r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim p(s'|s,a)} \left[V^\pi(s') \right], \qquad V^\pi(s) = \mathop{\mathbb{E}}_{a_0 \sim \pi} \left[Q^\pi(s_0, a_0) | \, s_0 = s \right]$$

2.2 Regularised RL via the Legendre-Fenchel Dual

Regularised MDPs were introduced by Geist et al. (2019) by extending the usual MDP tuple with a strongly convex regularisation function $\Omega : \Delta_{\mathcal{A}} \to \mathbb{R}$. This regulariser is subtracted from the value function, to conveniently yield the Legendre-Fenchel dual. Due to the strong convexity of the regulariser, the dual function, $\Omega^\star(Q)$ has useful properties: $\nabla\Omega^\star(Q)$ is Lipshitz and is the unique maximiser of the value function (Rockafellar, 1997; Niculae & Blondel, 2017). That is, $V^\star_\Omega(s) = \Omega^\star(Q^\star_\Omega(s,\cdot))$ is the regularised optimal value function and $\pi^\star_\Omega(\cdot|s) = \nabla\Omega^\star(Q^\star_\Omega(s,\cdot))$ is the optimal regularised policy. This formulation allows for efficient policy and value iteration algorithms, but restricts the class of permissible operators to those with strongly convex regularisers.

$$V^\star_\Omega(s) = \max_\pi V^\pi(s) - \Omega(\pi(\cdot|s)) = \max_\pi \langle \pi(\cdot|s), Q^\pi(s,\cdot) \rangle - \Omega(\pi(\cdot|s))$$

$$\left\| \nabla\Omega^\star(Q_1(s,\cdot)) - \nabla\Omega^\star(Q_2(s,\cdot)) \right\|_p \leq L \left\| Q_1(s,\cdot) - Q_2(s,\cdot) \right\|_p \quad \text{(Lipschitz, under p-norm)}$$

2.3 Mellowmax and soft mellowmax Q-value operators

Mellowmax and soft mellowmax are value functions, shown respectively in Equation 1 and Equation 2, designed to reduce the overestimation errors usually present in Q-learning algorithms like deep Q-network (DQN) (Mnih et al., 2015; Asadi & Littman, 2017; Kim et al., 2019; Gan et al., 2021). While Q-value overestimation can be alleviated by using a target network, as in Double DQN, to "decouple the selection from the evaluation (Van Hasselt et al., 2016)" of actions, Asadi & Littman (2017) propose a different approach: smoothing the Q-values through the mellowmax function to explicitly reduce their size. Kim et al. (2019) demonstrated that this operator is powerful enough to achieve competitive performance even without target networks. The soft mellowmax function was subsequently introduced by Gan et al. (2021) to reduce oversmoothing while retaining the benefits of bias reduction.

$$V_{\text{MM}}(s) = \frac{1}{\omega} \log \left(\sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{A}|} \exp(\omega Q(s,a)) \right) \tag{1}$$

Table 1: Probabilistic upper bounds on suboptimal action selection and value error at the root. There exists constants $C, \hat{C} > 0$ for each bound. The δ_a term in UCT is a value estimation error term. While the value bound is not explicitly stated in the UCT paper, it is easily derived. See Kocsis & Szepesvári (2006) and Dam et al. (2021) for full details.

	UCT	**CRTS**
$\mathbb{P}(a_t \neq a^\star)$	$Ct^{-\hat{C}\min_{a \neq a^\star} \delta_a^2}$	$Ct\exp\left(-\frac{t}{\hat{C}\sigma(log(t))^3}\right)$
$\mathbb{P}\left(\lvert\hat{V}_t(s_0) - V^\star(s_0)\rvert > \epsilon\right)$	$C\exp\left(-2t\epsilon^2\right)$	$C\exp\left(-\frac{t\epsilon}{\hat{C}\sigma(log(t))^2}\right)$

$$V_{\text{SMM}}(s) = \frac{1}{\omega}\log\left(\sum_{a\in\mathcal{A}} \text{softmax}(\alpha Q(s,\cdot))\exp(\omega Q(s,a))\right) \tag{2}$$

Note that both mellowmax and soft mellowmax can be achieved via KL regularisation, where the reference policy for mellowmax corresponds to a uniform distribution over the action space. Soft mellowmax is recovered when the reference policy is a softmax over the Q-values.

Crucially, V_{SMM} is a difference of convex functions and is not generally convex (Yao & Jiang, 2023). Thus, it cannot be directly applied to the Legendre-Fenchel dual framework used in CRTS (Dam et al., 2021). However, the robustness, stability, and sample efficiency of soft mellowmax suggest it could significantly accelerate planning if successfully adapted to MCTS (Eysenbach & Levine, 2021; Jiralerspong et al., 2025).

2.4 Monte Carlo tree search with various backup operators

Monte Carlo tree search (MCTS) emerged as a successor to Deep Blue when computer-based game-playing shifted in interest from Chess to Go (Campbell et al., 2002; Coulom, 2006). Instead of searching an MDP via brute force tree search with heuristics, Monte Carlo tree search uses Monte Carlo rollouts to efficiently estimate node values, and backup operators to propagate node values up the tree. MCTS can be understood in four phases, where for each simulation, $t \in [1, T]$, the following occurs (Mańdziuk, 2018):

1. **Selection:** Starting from the root, select actions using the tree policy until a leaf is reached.
2. **Expansion:** Select an action at the leaf node and add the resulting node to the tree.
3. **Value Estimation:** Estimate the new node's value via rollouts or a value function.
4. **Backup:** Propagate the value up the tree to update estimates $\hat{Q}_t$ and $\hat{V}_t$.

The tree policy and backup functions are usually derived from bandit algorithms. For instance, the tree search in AlphaZero (a variant of UCT) is based on the upper confidence bound action selection algorithm, and uses value averaging as a backup operator (Auer et al., 2002; Kocsis & Szepesvári, 2006; Silver et al., 2018). While UCT converges to the standard RL value and policy estimate, convex regularised tree search (CRTS) converges to the regularised value and policy functions. CRTS selects actions stochastically with the extended empirical exponential weight (E3W, see Equation 3) action selection algorithm and uses the regularised value function as a backup operator, where $\lambda_t = \epsilon/\log(t+1)$, with exploration hyperparameter ϵ (Xiao et al., 2019; Dam et al., 2021). Table 1 lists the convergence rates, highlighting the theoretical efficiency of regularised search.

$$d_t(a|s) = (1-\lambda_t)\nabla\Omega^\star(\hat{Q}_t(s,\cdot))(a) + \lambda_t/|\mathcal{A}| \tag{3}$$

Different choices of the regulariser Ω yield distinct algorithms for CRTS:

- **MENTS (Maximum Entropy):** Soft RL, uses Shannon entropy to encourage broad exploration (Xiao et al., 2019).
- **TENTS (Tsallis Entropy):** Uses Tsallis entropy (specifically with index $q = 2$), which leads to sparse policies (Dam et al., 2021).
- **RENTS (Relative Entropy):** Uses the KL-divergence relative to a reference policy, not dependent on the current Q-values (Dam et al., 2021). RENTS with a uniform reference policy is equivalent to mellowmax.

3 SOFT MELLOWMAX-BASED MONTE CARLO TREE SEARCH

Soft mellowmax-based tree search (SMM-TS) is proposed as an extension to existing literature in convex regularised tree search. Section 3.1 proposes the tree search architecture for SMM-TS, including assumed reward properties, tree policies, and backup operators. Section 3.2 then proves the convergence of SMM-TS for both optimal action selection and value error minimisation, with probabilistic bounds.

3.1 TREE SEARCH ARCHITECTURE FOR SMM-TS

At initialisation, the search tree contains a single node, the root node, representing the state at which planning occurs in the environment. Each node in the search tree is initialised in the same way: the visitation statistics $N(s,a) = 0$, and the Q-values are set to zero or initialised by some predictor function $\hat{Q}_0(s,a) = 0$ or $Q_\theta(s,a)$. However, new nodes can only be initialised and added to the tree when they are expanded (see Section 2.4). Value estimation is completed at newly expanded nodes, by running Monte Carlo rollouts, or using a value predictor, like a neural network. The backup process is where all node statistics in the trajectory from the root to the leaf, $\tau = s_0, a_0, \ldots, s_{\text{leaf}}, a_{\text{leaf}}$ are updated, $\forall (s_i, a_i) \in \tau$:

$$N(s_i, a_i) \leftarrow N(s_i, a_i) + 1$$

$$\hat{Q}(s_i, a_i) \leftarrow r(s_i, a_i) + \gamma V_{\text{SMM}}(\hat{Q}(s_{i+1}, \cdot))$$

Unlike UCT, which averages returns, SMM-TS replaces Q-values using the soft mellowmax operator V_{SMM} (or a value estimator at the leaf). Actions within the tree are selected using the E3W algorithm, Equation 3, based on the most recent Q-value estimates.

3.2 CONVERGENCE RESULTS FOR SMM-TS

Existing convergence proofs for regularised tree search (Xiao et al., 2019; Dam et al., 2021) rely on two properties: 1) the contraction of the value function, and 2) the smoothness of the Legendre-Fenchel dual. The latter is derived from the strong convexity of the regulariser via dual smoothness (Niculae & Blondel, 2017). These properties ensure that value and policy estimation at a parent node can be related to Q-value estimation at a child node, allowing for inductive proofs of convergence by stacking bandit problems. However, because V_{SMM} is a difference of convex functions (and thus not generally convex), we cannot rely on the dual formulation to establish smoothness (Yao & Jiang, 2023; Rockafellar, 1997; Niculae & Blondel, 2017). To prove convergence for SMM-TS, we must instead prove the smoothness of the operator directly.

Lemma 1 (Smoothness of the soft mellowmax value function)**.** *The soft mellowmax value function, $V_{SMM}(s)$ is $\left(\frac{(\alpha+\omega)^2+\alpha^2}{2\omega}\right)$-smooth.*

Proof of Lemma 1. Let $\text{LSE}(\cdot)$ denote the log-sum-exp function. We write the soft mellowmax value function as $V_s(Q) = \frac{1}{\omega}\left[\text{LSE}((\alpha+\omega)Q) - \text{LSE}(\alpha Q)\right]$ to make the dependence on Q explicit. By the triangle inequality and the known smoothness of log-sum-exp, $V_s(Q)$ is smooth:

$$\begin{aligned}
&\|\nabla V_s(Q_1) - \nabla V_s(Q_2)\|_2 \\
&= \frac{1}{\omega}\left\|\nabla\left[\text{LSE}((\alpha+\omega)Q_1) - \text{LSE}(\alpha Q_1) - \text{LSE}((\alpha+\omega)Q_2) - \text{LSE}(\alpha Q_2)\right]\right\|_2 \\
&\leq \frac{1}{\omega}\left\|\nabla\left[\text{LSE}((\alpha+\omega)Q_1) - \text{LSE}((\alpha+\omega)Q_2)\right]\right\|_2 + \frac{1}{\omega}\left\|\nabla\left[\text{LSE}(\alpha Q_1) - \text{LSE}(\alpha Q_2)\right]\right\|_2 \\
&\leq \frac{(\alpha+\omega)^2+\alpha^2}{2\omega}\|Q_1 - Q_2\|_2
\end{aligned}$$

The final bound is derived via the mean value theorem for vectors, solving for the maximum spectral norm of the Hessian of $V_s(Q)$ via the Gershgorin disk theorem (Axler, 2024). □

Gan et al. (2021) previously established that V_{SMM} is a contraction. Combined with the smoothness proved in Lemma 1, the inductive proofs of Dam et al. (2021) hold for SMM-TS. Consequently, SMM-TS inherits the convergence rates for value and action selection error shown in Table 1.

4 EXPERIMENTS

Soft mellowmax tree search is compared to regularised tree search algorithms and AlphaZero's UCT in synthetic tree environments and MinAtar arcade games. These experiments are designed to showcase two essential properties for scalable recursive self-improvement: the efficiency of the planning loop, and the ability to perform well under different initialisations.

4.1 SYNTHETIC TREE EXPERIMENTS

The synthetic tree experiments introduced by Xiao et al. (2019) and Dam et al. (2024) are useful for measuring the convergence speed of tree search methods to their optimal values. In particular, the value estimation error $\left|V^{\star}_{\Omega}(s) - \hat{V}_{t,\Omega}(s)\right|^2$ is measured for each tree search algorithm in synthetic trees of various depths, d, and action space sizes, k. Each synthetic tree experiment is seeded with a new random number, which is used to generate both the mean reward of a node, $\mu_n \sim \mathcal{N}(0, 1)$, and stochastic samples of the rewards $X_{n,i} \sim \mathcal{N}(|\mu_n|, \sigma^2)$, where i is the index of the i-th sampling of a node, and n is the index of the node. The transition function is deterministic, so stochasticity emerges solely due to the rewards and action selection algorithm. Over increasing simulations, it is expected that the value estimation error of all methods will decrease.

We examine performance in two regimes: (1) **Uniform Initialisation** (simulating a "cold start" without priors), and (2) **Perturbed Initialisation** (simulating a learned but imperfect policy prior). The initial Q-value, value, and policy of a tree can have drastic effects on the speed of convergence. As was demonstrated in Rosin (2011) and Silver et al. (2018) for UCB-based bandits and trees, using policy and value predictors can improve the speed of convergence considerably, by biasing action selection towards favourable actions.

Hyperparameter settings for the synthetic tree experiments Each tree search algorithm is run with at least 50 random seeds, for budgets $T \in [100, 30000]$. We set $\omega = 1$ for regularised methods, $\alpha = 1$ for SMM-TS, and exploration $\epsilon = 0.1$ (Dirichlet fraction for UCT), following standard baselines (Dam et al., 2021; DeepMind et al., 2020; Silver et al., 2018).

4.1.1 VALUE ERROR CONVERGENCE UNDER UNIFORM INITIALISATION

Under uniform initialisation ($\hat{Q}_0(s, a) = 1, \hat{\pi}_0(a|s) = 1/|\mathcal{A}|; \forall (s, a) \in \mathcal{S} \times \mathcal{A}$), we can fairly compare each tree search method without useful prior value or policy estimates to bias action selection. Figure 1 demonstrates that SMM-TS converges significantly faster than UCT, MENTS, RENTS and TENTS across all tree sizes.

Relevance to self-improvement In a recursive loop, a tree search generates policy and value estimates that can be used to train neural networks, or in inference to improve policy performance. The slow convergence of UCT, for instance, implies a high computational cost to generate low-error estimates compared to SMM-TS. As such, the fast convergence speed of SMM-TS is highly desirable for realistic planning scenarios, allowing for more iterations within a fixed compute budget, or more simulations per iteration for a fixed iteration budget.

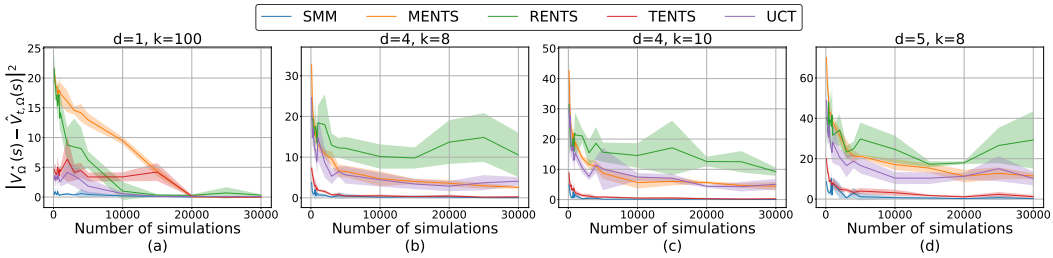

Figure 1: Value error convergence under uniform initialisation. SMM-TS minimizes value error fastest, indicating high sample efficiency when priors are unavailable.

4.1.2 VALUE ERROR CONVERGENCE UNDER INITIALISATION ERROR

Initialising a tree with perturbed optimal values, Q-values and policies allows for an understanding of how better priors can achieve better performance under planning. We initialise the tree Q-values with perturbed optimal Q-values: $\hat{Q}_0(s,\cdot) = Q^\star(s,\cdot) + \delta z / \|z\|_2$, where $z \sim \mathcal{N}(0, I)$. Predictor error of size δ refers to a Q-value estimate whose error is δ under an L2 norm: $\left\|\hat{Q}_0(s,\cdot) - Q^\star(s,\cdot)\right\|_2 = \delta$. This experiment simulates a more realistic machine learning setting, where a prior (*e.g.*, a DQN agent's Q-value function) is available, and may improve over time.

Figure 2 illustrates the importance of good initialisation in tree search: the value prediction error for each method drops drastically compared to Figure 1 for low predictor error $\delta \in 0.2, 0.5, 1$. Interestingly, SMM-TS, TENTS, and UCT perform similarly in this setting, with UCT improving the most from the previous experiment. Finally, it is important to observe that value convergence is also a function of tree initialisation and predictor error. This relation is not reflected by the existing theory in CRTS, *e.g.*, Xiao et al. (2019); Dam et al. (2021); Painter et al. (2023).

Relevance to self-improvement This result highlights an important fact with self-improving agents: the improvement of an agent over time is compounded by the planning algorithm. Clearly, the lower the predictor error, the lower the tree search value error. This is important since it suggests that for all tested methods, if prior information is available, it should be used to initialise the tree, so long as the predictor error is not too high.

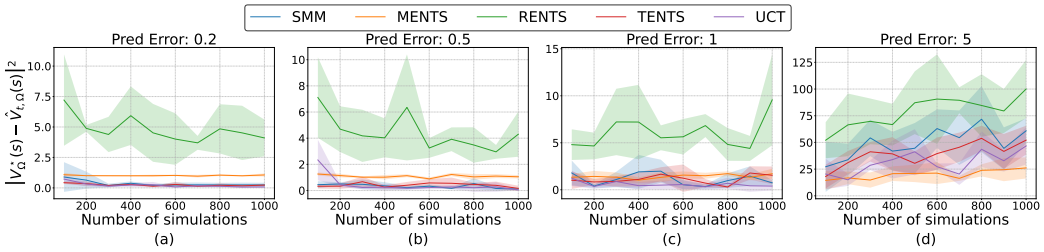

Figure 2: Comparing the performance of regularised tree search methods on a synthetic tree of depth 5, with 8 actions per node, under worsening initialisation error.

4.2 MINATAR EXPERIMENTS

MinAtar is a set of miniaturised Atari 2600 environments designed to efficiently test game-playing algorithms (Young & Tian, 2019; Koyamada et al., 2023).[1] Of the five MinAtar environments, Seaquest is excluded from the experiments because it was difficult to train the baseline agents to convergence.

The MinAtar experiments are designed similarly to Xiao et al. (2019) and Dam et al. (2021). Three baseline agents are trained to provide initialisation values to the tree searches: DQN, soft DQN (SDQN) and soft mellowmax DQN (SM). The latter two agents are trained by generating value and Q-value estimates via the regularised Bellman operators, and policies via the regularised optimal policy functions. Note that, in contrast to Xiao et al. (2019) and Dam et al. (2021), tree Q-values are initialised directly from the baseline agent: $Q_0(s,a) = Q_\theta^{\text{agent}}(s,a)$.

Setup We sweep inverse temperature $\omega \in [1, 20]$ and $\alpha \in \{0, 1, 5\}$. Experiments use 100 random seeds with 512 simulations per step. The exploration hyperparameter is set similarly to above: $\epsilon = 0.1$ (which is the Dirichlet fraction for UCT). UCT's hyperparameters are set as above.

Results and Analysis Table 2 presents one sub-table for each baseline agent. Bolding indicates that an algorithm's mean return was not statistically significantly different from the highest mean

[1]For a visualisation of the MinAtar environments, see https://github.com/kenjyoung/MinAtar.

in the column, using a t-test with $p > 0.05$ (Dam et al., 2021). We further evaluate the best-performance of the agents using Nash averaging, where the results of each column are standardised by subtracting the `min` and dividing by the (new) `max` (Balduzzi et al., 2018). The resulting maximum-entropy Nash equilibria give interesting interpretations on the performance of the tree search algorithms. While high variance prevents us from identifying a single best search algorithm for each task, some interesting patterns emerge that allow us to comment on the benefits of regularised RL and MCTS more generally.

- **Planning universally improves agent performance**: The baseline performance of each agent is significantly improved when using a planning algorithm, with mean performance increasing by 2-3 times in most environments. This is reflected by the Nash average of the non-tree search agents being nearly zero, indicating that they are dominated by tree search.
- **The dominance of KL-regularised agents**: RENTS is regularised with KL-divergence, using a temperature=1 softmax of the agent's Q-values as the reference policy. It is clear that this static reference policy achieves high performance in all games, across all agents. Conversely, SMM-TS uses the tree's current Q-values to generate a reference policy. While SMM-TS achieves high performance in many environments, it could be that its dynamic reference policy holds back performance, as tree Q-values may be highly erroneous. However, for all agents, it is clear that RENTS or SMM-TS achieve non-zero weights in the Nash equilibria, indicating that they are non-dominated, and non-redundant.
- **The importance of regularised search**: What is clear from these results is that UCT is almost universally outperformed by regularised search methods, with all types of agents. This suggests that the policies generated by regularised search may be more robust to the stochasticity of the environment, and can plan around this stochasticity better than reward-maximising agents.

Table 2: Comparison of best tree search results across a sweep of inverse temperatures for MinAtar environments. Results are presented as mean $\pm$ 95% confidence interval. Nash equilibria are rounded to 3 decimal places.

Agent: DQN

Search	Asterix	Breakout	Freeway	Space Invaders	Nash Eq.
NONE	15.78 ± 2.70	19.23 ± 2.25	21.59 ± 0.56	4.78 ± 1.16	0
MENTS	$\mathbf{46.46 \pm 4.43}$	$\mathbf{38.82 \pm 4.09}$	26.19 ± 0.25	$\mathbf{153.56 \pm 7.25}$	0.267
RENTS	$\mathbf{51.33 \pm 4.52}$	$\mathbf{36.24 \pm 3.33}$	$\mathbf{26.58 \pm 0.23}$	$\mathbf{160.13 \pm 4.92}$	0.551
SMM	44.05 ± 4.86	$\mathbf{35.32 \pm 3.76}$	$\mathbf{26.58 \pm 0.26}$	$\mathbf{159.96 \pm 4.98}$	0.043
TENTS	42.30 ± 4.72	$\mathbf{39.56 \pm 4.25}$	$\mathbf{26.59 \pm 0.26}$	151.23 ± 6.19	0.126
UCT	42.89 ± 4.40	$\mathbf{33.87 \pm 3.76}$	$\mathbf{26.46 \pm 0.24}$	151.03 ± 5.76	0.013

Agent: SDQN

Search	Asterix	Breakout	Freeway	Space Invaders	Nash Eq.
NONE	16.93 ± 2.93	14.79 ± 2.75	23.07 ± 0.35	29.64 ± 5.27	0
MENTS	42.47 ± 4.57	37.23 ± 3.79	26.43 ± 0.25	$\mathbf{148.93 \pm 7.17}$	0.01
RENTS	$\mathbf{50.08 \pm 4.78}$	$\mathbf{43.04 \pm 4.30}$	$\mathbf{26.78 \pm 0.23}$	$\mathbf{154.92 \pm 6.61}$	0.933
SMM	40.11 ± 4.58	33.74 ± 3.78	$\mathbf{26.68 \pm 0.28}$	$\mathbf{150.97 \pm 6.46}$	0.002
TENTS	$\mathbf{45.14 \pm 4.62}$	$\mathbf{37.75 \pm 3.81}$	$\mathbf{26.76 \pm 0.24}$	$\mathbf{147.17 \pm 7.76}$	0.033
UCT	42.10 ± 4.50	$\mathbf{38.96 \pm 3.89}$	$\mathbf{26.74 \pm 0.26}$	$\mathbf{151.74 \pm 4.11}$	0.022

Agent: SM

Search	Asterix	Breakout	Freeway	Space Invaders	Nash Eq.
NONE	17.56 ± 3.04	14.40 ± 1.76	17.30 ± 0.64	62.52 ± 8.84	0
MENTS	$\mathbf{46.07 \pm 4.65}$	$\mathbf{43.66 \pm 4.37}$	26.21 ± 0.23	136.73 ± 9.84	0.085
RENTS	$\mathbf{43.79 \pm 4.71}$	$\mathbf{47.13 \pm 4.33}$	$\mathbf{26.61 \pm 0.27}$	$\mathbf{159.01 \pm 5.76}$	0.754
SMM	$\mathbf{42.88 \pm 5.07}$	40.33 ± 4.76	$\mathbf{26.59 \pm 0.23}$	$\mathbf{154.12 \pm 7.55}$	0.083
TENTS	$\mathbf{44.38 \pm 5.12}$	36.05 ± 4.59	$\mathbf{26.40 \pm 0.28}$	140.01 ± 8.04	0.013
UCT	$\mathbf{40.62 \pm 4.72}$	$\mathbf{45.70 \pm 4.42}$	$\mathbf{26.36 \pm 0.25}$	142.54 ± 8.73	0.064

Despite sweeping α, the highest means for SMM-TS were universally achieved with $\alpha \in \{0, 1\}$. This indicates that either mellowmax (uniform reference policy) or soft mellowmax with a softer reference policy, were the best backup operators for SMM-TS. With regards to the inverse temperature ω, it seems that the optimal setting is environment-dependent, since no distinct pattern emerges across the experiments for any of the tree search algorithms.

4.3 Commentary on the experimental results

There is a discordance between the synthetic tree search experiments of Section 4.1 and the MinAtar experiments of Section 4.2. In the synthetic environments, RENTS had universally higher value estimation error than any other method, but in the MinAtar environments it was consistently amongst the best-performing search algorithms. TENTS, on the other hand, failed to match previously reported success in arcade games, despite its exceptional value error curves (Dam et al., 2021). SMM-TS was somewhere in the middle, demonstrating quick value convergence and robustness to predictor error, but not achieving high performance in all environments. These differences in performance could be a result of instabilities caused by the static versus dynamic reference policies of RENTS and SMM-TS, respectively. An interesting direction for future work would be to mix the static Q-values of the predictor and dynamic Q-values of the tree, ideally ensuring lower-error reference policies, more stable convergence, and better empirical performance.

Conclusion

Soft mellowmax tree search is proposed as an extension of the convex regularised tree search literature. SMM-TS is shown to converge theoretically and empirically in tree value estimates. Synthetic tree experiments demonstrate that the quality of the initialisation is important in value error convergence speed. Despite the poor performance of RENTS in the synthetic tree experiments, in MinAtar RENTS was always amongst the best-performing methods. Experiments in MinAtar also demonstrate that SMM-TS is capable of matching the performance of other tree search algorithms in these simple tasks. It would be interesting to apply regularised tree search algorithms to tasks necessitating diversity, such as scientific discovery. This may better demonstrate the utility of regularised policies due to their natural diversity, and robustness to mis-specified rewards and environmental perturbations.

References

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pp. 243–252. PMLR, 2017.

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

Sheldon Jay Axler. *Linear Algebra Done Right*, pp. 170–171. Springer, New York, 2024. URL `http://linear.axler.net/`.

David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. *Advances in Neural Information Processing Systems*, 31, 2018.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134 (1-2):57–83, 2002.

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

Tuan Dam, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. A unified perspective on value backup and exploration in monte-carlo tree search. *Journal of Artificial Intelligence Research*, 81:511–577, 2024.

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

Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with gumbel. In *International Conference on Learning Representations*, 2022.

DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem. Technical report, Google Deepmind, 2020. URL `http://github.com/deepmind`.

Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.

Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. In *Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=PtSAD3caaA2`.

Yaozhong Gan, Zhe Zhang, and Xiaoyang Tan. Stabilizing q learning via soft mellowmax operator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35-9, pp. 7501–7509, 2021.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International conference on machine learning*, pp. 2160–2169. PMLR, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 2018.

Hisham Husain, Kamil Ciosek, and Ryota Tomioka. Regularized policies are reward robust. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 64–72. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/husain21a.html`.

Marco Jiralerspong, Esther Derman, Danilo Vucetic, Nikolay Malkin, Bilun Sun, Tianyu Zhang, Pierre-Luc Bacon, and Gauthier Gidel. Robust reinforcement learning for discrete compositional generation via general soft operators. *arXiv preprint arXiv:2506.17007*, 2025.

Seungchan Kim, Kavosh Asadi, Michael Littman, and George Konidaris. Deepmellow: removing the need for a target network in deep q-learning. In *Proceedings of the twenty eighth international joint conference on artificial intelligence*, 2019.

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

Sotetsu Koyamada, Shinri Okano, Soichiro Nishimori, Yu Murata, Keigo Habara, Haruka Kita, and Shin Ishii. Pgx: Hardware-accelerated parallel game simulators for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45716–45743, 2023.

Jacek Mańdziuk. *MCTS/UCT in Solving Real-Life Problems*, pp. 277–292. Springer International Publishing, Cham, 2018. ISBN 978-3-319-67946-4. doi: 10.1007/978-3-319-67946-4_11. URL `https://doi.org/10.1007/978-3-319-67946-4_11`.

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.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. *Advances in neural information processing systems*, 30, 2017.

Michael Painter, Mohamed Baioumy, Nick Hawes, and Bruno Lacerda. Monte carlo tree search with boltzmann exploration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NG4DaApavi.

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, pp. 74–118. John Wiley & Sons, Inc., 1994a.

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, pp. 153–154. John Wiley & Sons, Inc., 1994b.

R.T. Rockafellar. *Convex Analysis*, chapter 12. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1997. ISBN 9780691015866. URL https://books.google.ca/books?id=1TiOka9bx3sC.

Christopher D. Rosin. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230, March 2011. ISSN 1012-2443. doi: 10.1007/s10472-011-9258-6. URL https://doi.org/10.1007/s10472-011-9258-6.

David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George 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*, 529:484–489, 01 2016. doi: 10.1038/nature16961.

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 with deep neural networks and tree search. *Nature*, 550:354–359, 10 2017. doi: 10.1038/nature24270.

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. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404. URL https://www.science.org/doi/abs/10.1126/science.aar6404.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, pp. 47–72. The MIT Press, Cambridge, MA, 2018.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30-1, 2016.

Chenjun Xiao, Ruitong Huang, Jincheng Mei, Dale Schuurmans, and Martin Müller. Maximum entropy monte-carlo planning. *Advances in Neural Information Processing Systems*, 32, 2019.

Chaorui Yao and Xin Jiang. A globally convergent difference-of-convex algorithmic framework and application to log-determinant optimization problems. *arXiv preprint arXiv:2306.02001*, 2023.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv preprint arXiv:1903.03176*, 2019.

Peng Zhai, Jie Luo, Zhiyan Dong, Lihua Zhang, Shunli Wang, and Dingkang Yang. Robust adversarial reinforcement learning with dissipation inequation constraint. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5431–5439, Jun. 2022. doi: 10.1609/aaai.v36i5.20481. URL https://ojs.aaai.org/index.php/AAAI/article/view/20481.