Supplementary Material

A On Planning in Prefix Value Trees

In a prefix value (PV) tree, the values of nodes are drawn from the set of integers, with positive values representing wins for the maximizing player (henceforth, Max) and the rest indicating wins for the minimizing player (Min). For the sake of simplicity, we disallow draws. Let $m(v)$ represent the minimax value of a node v. We grow the subtree rooted at v as follows:

- Let $V = \{v_1, v_2, \ldots, v_b\}$ represent the set of children of v, corresponding to action choices $A = \{a_1, a_2, \ldots, a_b\}$.
- Pick an $a_i \in A$ uniformly at random — this is designated to be the optimal action choice at v.
- Assign $m(v_i) = m(v)$. If Max is on move at v, then $m(v_j) = m(v) - k, \forall j \neq i$. If Min is on move v, then $m(v_j) = m(v) + k, \forall j \neq i$.

Here, k is a constant that represents the cost incurred by the player on move for taking a sub-optimal action. The depth of the tree is controlled by the parameter d_{max} and a uniform branching factor of b is assumed.

The PV tree model, is an attractive object of study as despite its relative simplicity, it captures a very rich class of games. However, it has one major drawback: a simple 1-ply lookahead search, using the average outcome of random playout trajectories as a heuristic, achieves very high decision accuracies. We now explore this phenomenon a little deeper.

Without loss of generality, we restrict our attention to trees where Max is on move at the root node n. Moreover, we require that $m(n) = 1$ and that n has exactly one optimal child — this ensures that our search algorithm is faced with a non-trivial decision at the root node. While we focus on the case where $b = 2$ in what follows, extending our results to higher branching factors is straightforward. We denote the left and right children of n by l and r respectively and assume that l is the optimal move. Define $S_d(v)$ to be the sum of the minimax values of the leaf nodes in the subtree of depth d rooted at node v.

Proposition 1. *$S_d(l) - S_d(r) = 2^d$ for all $d \geq 0$.*

Proof. We proceed by induction on d. For the base case, $S_0(l) - S_0(r) = 1 - 0 = 2^0$ by definition. Assume the claim holds for $d = t, t \geq 0$, where t is a Max level. Then, $S_{t+1}(l) - S_{t+1}(r) = (S_t(l) + (S_t(l) - k)) - (S_t(r) + (S_t(r) - k)) = 2(S_t(l) - S_t(r)) = 2 \cdot 2^d = 2^{d+1}$. A symmetric argument can be made for the case where t is a Min level. □

Define $P(v)$ to be the average outcome of random playouts performed from the node v. If the subtree rooted at v has uniform depth d, then $\mathbb{E}[P(v)] = S_d(v)/2^d$. An immediate consequence of Proposition 1 is that $\mathbb{E}[P(l)] - \mathbb{E}[P(r)] = (S_d(l) - S_d(r))/2^d = 1$, for any depth d, i.e., in the limit, the estimated utility of the optimal move l at the root will always be greater than that of r. In other words, the decision accuracy of a 1-ply lookahead search informed by random playouts approaches 100% with increasing number of playouts, independent of the depth of the tree. It is straightforward to extend this result to the case even when k is a random variable, drawn uniformly at random from some set $\{1, \ldots, k_{max}\}$.

B On the Distribution of Leaf Node Values in Critical Win-Loss Games

In this section, we analyze the density of $+1$ nodes in critical win-loss game as a function of its depth (d), branching factor (b) and critical rate (γ). We limit ourselves to the case where b is uniform, $b \geq 2$ and $0 < \gamma \leq 1$.

Without loss of generality, we assume that the root is a maximizing choice node (i.e., has a minimax value of $+1$). Then, the expected number of its $+1$ children is:

$$1 + (1 - \gamma) \cdot (b - 1) = b + \gamma - b\gamma$$

We denote the expected *density* of these $+1$ children (among the possible b children) by k, where:

$$k = \frac{b + \gamma - b\gamma}{b} = 1 - \gamma + \frac{\gamma}{b} \tag{3}$$

We note that by a symmetric argument, this expression also captures the density of the -1 children of a minimizing choice node.

Now consider a critical win-loss game instance rooted at a maximizing choice node. Let f_n denote the $+1$ density at depth n in this tree. We will calculate a closed-form expression for f_n. When n is even (i.e., Max is on move), we have the following recursive relationship due to equation (3):

$$f_{n+1} = f_n \cdot \left(1 - \gamma + \frac{\gamma}{b}\right)$$

Substituting $2d$ for n, we have:

$$f_{2d+1} = f_{2d} \cdot k \tag{4}$$

When n is odd (i.e., Min is on move), the choice nodes have value -1. Once again using equation (3), we derive the following recursive equation for -1 nodes:

$$\begin{aligned} 1 - f_{n+1} &= (1 - f_n) \cdot \left(1 - \gamma + \frac{\gamma}{b}\right) \\ f_{n+1} &= 1 - (1 - f_n)\left(1 - \gamma + \frac{\gamma}{b}\right) \\ f_{n+1} &= f_n \cdot k + 1 - k \end{aligned}$$

Substituting $2d + 1$ for n, we have:

$$f_{2d+2} = f_{2d+1} \cdot k + 1 - k \tag{5}$$

From equations (4) and (5), we have:

$$f_{2d+2} = f_{2d} \cdot k^2 + (1 - k)$$

By induction, we can then derive the following non-recurrent formula for f_{2d}:

$$f_{2d} = f_0 \cdot (k^2)^d + (1 - k) \cdot \frac{1 - (k^2)^{d+1}}{1 - k^2}$$

where $f_0 = 1$. Simplifying, we have:

$$f_{2d} = k^{2d} + \frac{1 - k^{2d+2}}{1 + k}$$

If we now allow $d \to \infty$, we have:

$$\lim_{d \to \infty} f_{2d} = \frac{1 - k}{1 - k^2} = \frac{1}{1 + k} = \frac{1}{2 - \gamma + \frac{\gamma}{b}} \tag{6}$$

and:

$$\lim_{d \to \infty} f_{2d+1} = \lim_{d \to \infty} f_{2d} \cdot k = \frac{k}{1 + k} = \frac{1 - \gamma + \frac{\gamma}{b}}{2 - \gamma + \frac{\gamma}{b}} \tag{7}$$

C Deriving Bounds on Pathological Behavior

We provide a proof of Theorem 1 from Section 4, which is restated below.

Theorem 1. *In a critical win-loss game with $\gamma = 1.0$, UCT with a search budget of N nodes will exhibit lookahead pathology for choices of the exploration parameter $c \geq \sqrt{\frac{N^3}{2 \log N}}$, even with access to a perfect heuristic.*

Proof. The proof consists of two parts. First, we argue that with an appropriately chosen value for the exploration constant c, UCT will build a balanced search tree in a breadth-first fashion. Then, we show that such a tree building strategy will cause UCT's decision accuracy to devolve to random guessing with increased search effort. Consider a node p with b children. Let a_1 and a_2 denote two of these children such that $n(a_1) < n(a_2)$. Our aim is to find a value for the exploration parameter c such that the UCB1 formula will prioritize visiting a_1 over a_2, regardless of the difference in their (bounded) utility estimates. This amounts to UCT building a search tree in a breadth-first fashion.

Without loss of generality, assume p is a maximizing node. For UCT to visit a_1 before a_2 on the next iteration, we must have:

$$\overline{Q}(a_1) + c\sqrt{\frac{\log n(p)}{n(a_1)}} > \overline{Q}(a_2) + c\sqrt{\frac{\log n(p)}{n(a_2)}}$$

Rearranging terms, this is equivalent to the condition:

$$c > \frac{\overline{Q}(a_2) - \overline{Q}(a_1)}{\sqrt{\log n(p)}\left(\sqrt{\frac{1}{n(a_1)}} - \sqrt{\frac{1}{n(a_2)}}\right)} \tag{8}$$

We will now bound the right-hand side of equation 8 in terms of our search budget N. Firstly, since the heuristic estimates of nodes are bounded by $[0, 1]$, we know that $\overline{Q}(a) \in [0, 1]$ for any node a. We can therefore bound the numerator of equation (8) from above as:

$$\overline{Q}(a_2) - \overline{Q}(a_1) \leq 1 \tag{9}$$

Now we turn our attention to the denominator D of equation (8). We have:

$$\begin{aligned}
D &= \sqrt{\log n(p)} \left(\sqrt{\frac{1}{n(a_1)}} - \sqrt{\frac{1}{n(a_2)}} \right) \\
&= \sqrt{\log n(p)} \left[\frac{n(a_2) - n(a_1)}{\sqrt{n(a_1)n(a_2)}(\sqrt{n(a_1)} + \sqrt{n(a_2)})} \right] \\
&\geq \sqrt{\log n(p)} \left[\frac{1}{\sqrt{n(a_1)n(a_2)}(\sqrt{n(a_1)} + \sqrt{n(a_2)})} \right] && \text{since } n(a_2) > n(a_1) \\
&\geq \sqrt{\log n(p)} \left[\frac{1}{\frac{n(a_1)+n(a_2)}{2}(\sqrt{n(a_1)} + \sqrt{n(a_2)})} \right] && \text{by the AM-GM inequality} \\
&\geq \sqrt{\log n(p)} \left[\frac{1}{\frac{n(a_1)+n(a_2)}{2}\sqrt{2(n(a_1) + n(a_2))}} \right] && \text{since } \sqrt{x} + \sqrt{y} \leq \sqrt{2(x+y)} \\
&\geq \sqrt{\log n(p)} \left[\frac{1}{\frac{n(p)}{2}\sqrt{2 \cdot n(p)}} \right] && \text{since } n(a_1) + n(a_2) \leq n(p) \\
&= \sqrt{\frac{2 \log n(p)}{n(p)^3}} \\
&\geq \sqrt{\frac{2 \log N}{N^3}}
\end{aligned}$$

Combining this bound with equation (9), we conclude that:

$$\frac{\overline{Q}(a_2) - \overline{Q}(a_1)}{\sqrt{\log n(p)} \left(\sqrt{\frac{1}{n(a_1)}} - \sqrt{\frac{1}{n(a_2)}} \right)} \leq \sqrt{\frac{N^3}{2 \log N}}$$

Thus, choosing a value for the exploration constant c that is larger than this quantity, as per equation (8), will force UCT to build a search tree in a breadth-first fashion.

We now conclude by arguing why such a node expansion strategy will lead to lookahead pathology. Without loss of generality, consider a game tree rooted at a minimizing choice node p (i.e., a minimizing node with minimax value -1). Let s^* denote an optimal child of p and let s denote a sub-optimal child. This means that s^* is a maximizing forced node and s is a maximizing choice node. The density of winning nodes from the maximizing perspective at depth $2d$ from s is then given by equation (1). Since s^* is a forced node, we cannot directly use equations (1) or (2). However, we observe that all the children of s^* are minimizing choice nodes, and thus, the density of winning moves from the minimizing perspective at depth $2d$ from s^* is given by f_{2d-1}. We can use the negamax transformation to recast this as the density of winning nodes from the maximizing perspective at depth $2d$ from s^*: this is given by $1 - f_{2d-1}$. For large values of d, we know from equations (6) and (7) that $f_{2d} + f_{2d-1} = 1$, or $f_{2d} = 1 - f_{2d-1}$. In other words, the density of $+1$ leaf nodes at depth $2d$ in the subtrees rooted at s^* and s approach the same value, for sufficiently large d. Since the average of the utilities of these leaves at level $2d$ will dominate the average of *all* the leaves in the respective subtrees, we conclude that in large enough search trees, UCT's estimate of $\overline{Q}(s^*)$ will approach its estimate of $\overline{Q}(s)$. In other words, the algorithm will not be able to tell apart optimal and sub-optimal moves as it builds deeper trees, even though we have access to the true minimax value of each node in this setting, leading to the emergence of pathological behavior. □

D Game Engine Settings

We collected our Chess data using the Stockfish 13 engine, with the following configuration:

```
"Debug Log File": "",
"Contempt": "24",
"Threads": "1",
"Hash": "16",
"Clear Hash Ponder": "false",
"MultiPV": "1",
"Skill Level": "20",
"Move Overhead": "10",
"Slow Mover": "100",
"nodestime": "0",
"UCI_Chess960": "false",
"UCI_AnalyseMode": "false",
"UCI_LimitStrength": "false",
"UCI_Elo": "1350",
"UCI_ShowWDL": "false",
"SyzygyPath": "",
"SyzygyProbeDepth": "1",
"Syzygy50MoveRule": "true",
"SyzygyProbeLimit": "7",
"Use NNUE": "false",
"EvalFile": "nn-62ef826d1a6d.nnue"
```

We collected our Othello data using the Edax 4.4 engine[1] with the default settings. Edax is freely available online under a GNU GPL v3.0 license.

[1] https://github.com/abulmo/edax-reversi

E Critical Rates in Othello

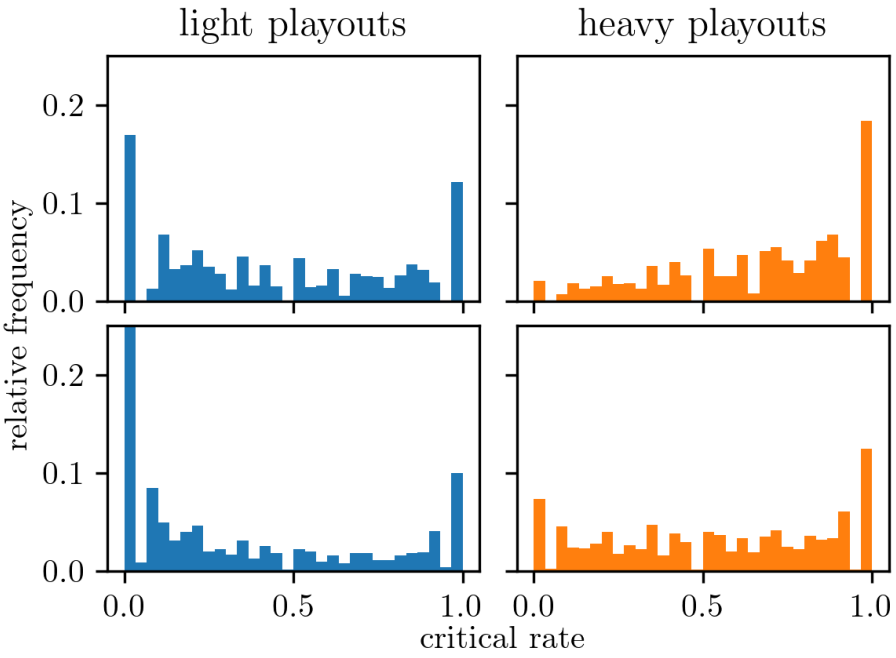

Figure 6: Histograms of empirical critical rates ($\tilde{\gamma}$) for Othello positions sampled $p = 10$ (top row) and $p = 36$ (bottom row) plies deep into the game. We sample the positions using both light playouts (left column) and heavy playouts (right column).

F Impact of Maximum Tree Depth on Pathology

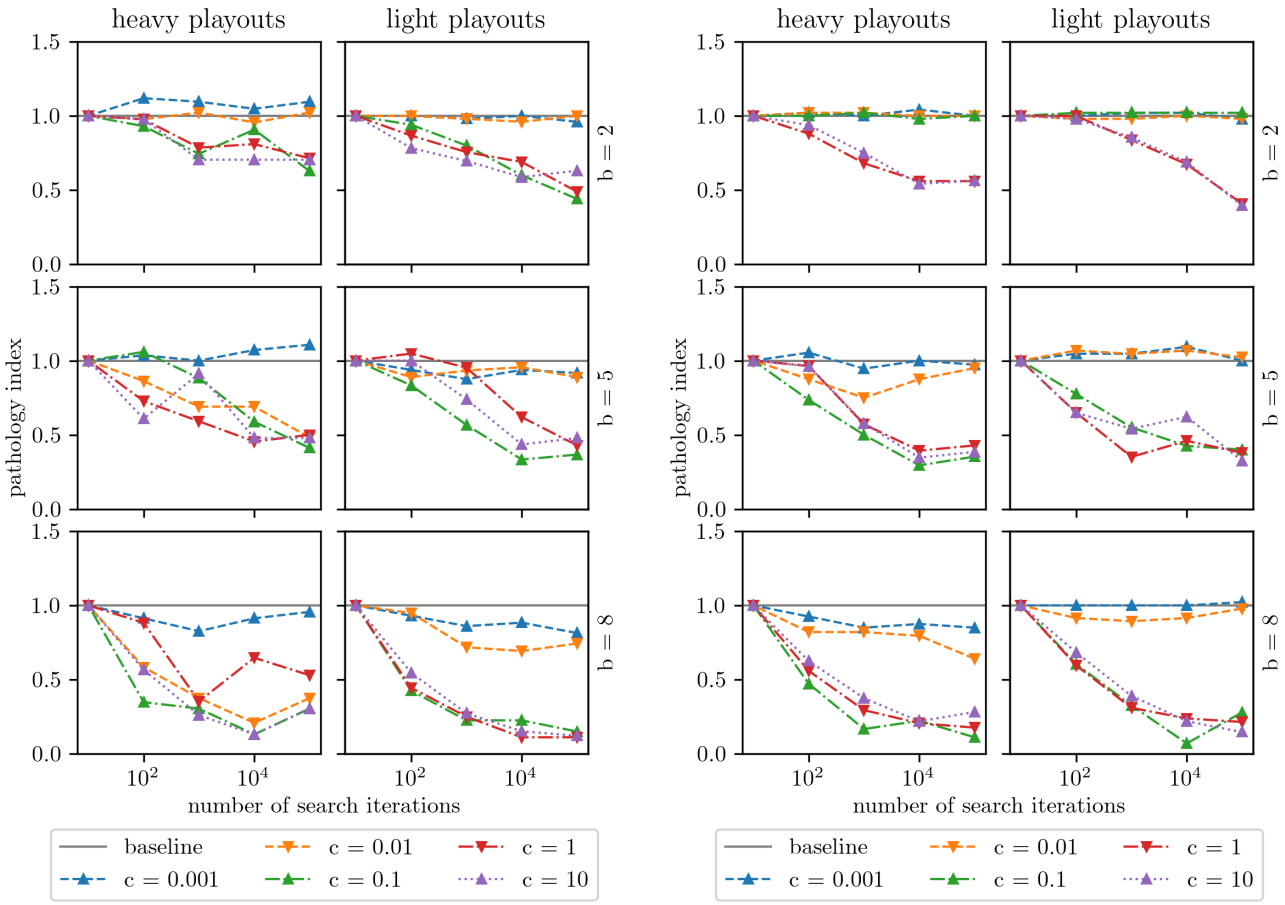

Figure 7: Measuring pathological behavior in UCT on critical win-loss games of depth 200 with $\gamma = 1.0$. The pair of plots on the left correspond to using a heuristic constructed from histograms of Stockfish evaluations of Chess positions sampled at depth 10, using both light and heavy playouts. The pair of plots on the right correspond to using a heuristic constructed from histograms of Edax evaluations of Othello positions sampled at depth 10, using both light and heavy playouts. Each colored line corresponds to an instantiation of UCT with a different exploration constant. The x-axis is plotted on a log-scale. We note the continued persistence of lookahead pathology.

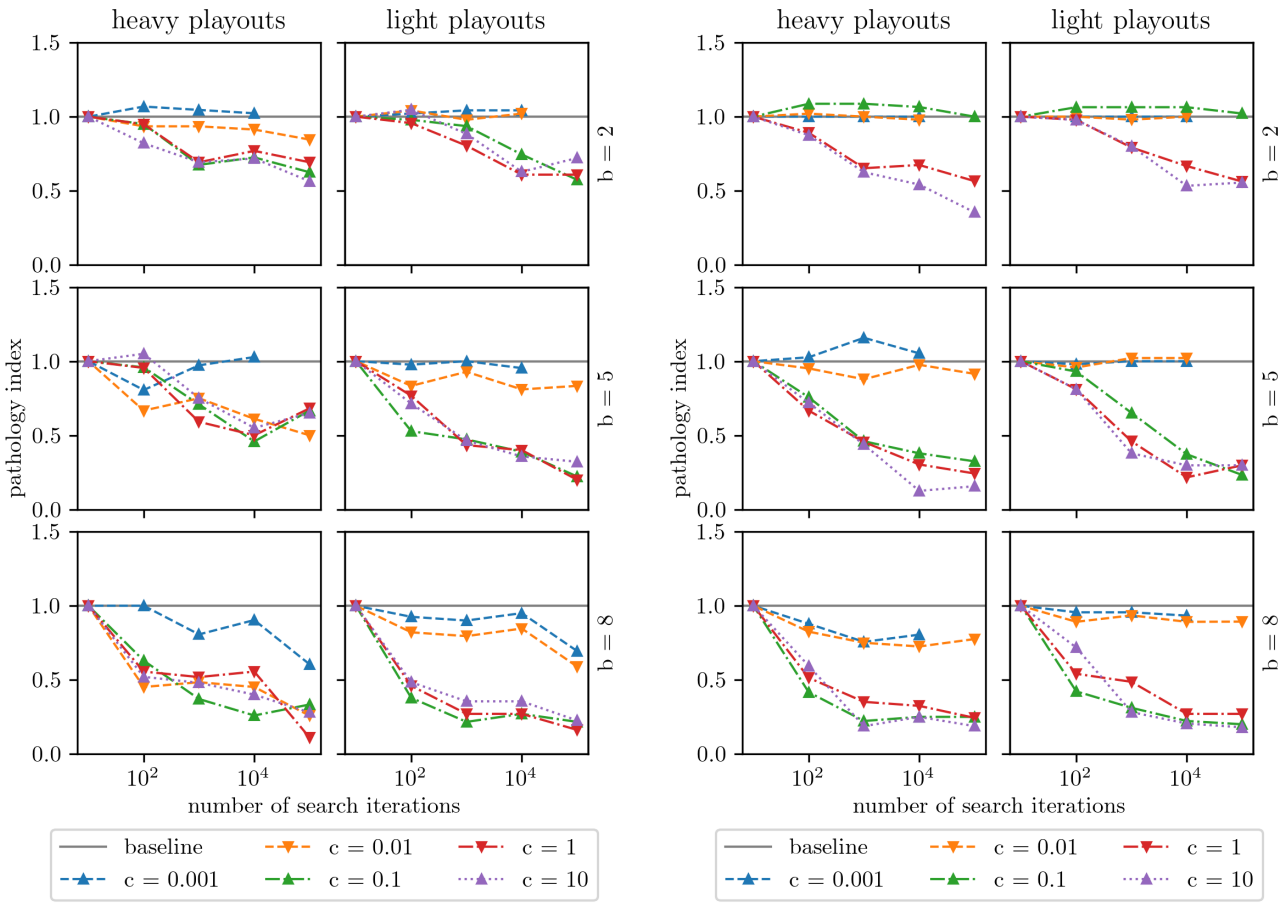

Figure 8: Measuring pathological behavior in UCT on critical win-loss games of depth 2000 with $\gamma = 1.0$. The pair of plots on the left correspond to using a heuristic constructed from histograms of Stockfish evaluations of Chess positions sampled at depth 10, using both light and heavy playouts. The pair of plots on the right correspond to using a heuristic constructed from histograms of Edax evaluations of Othello positions sampled at depth 10, using both light and heavy playouts. Each colored line corresponds to an instantiation of UCT with a different exploration constant. The x-axis is plotted on a log-scale. We note the continued persistence of lookahead pathology.

G Impact of Low Critical Rate on Pathology

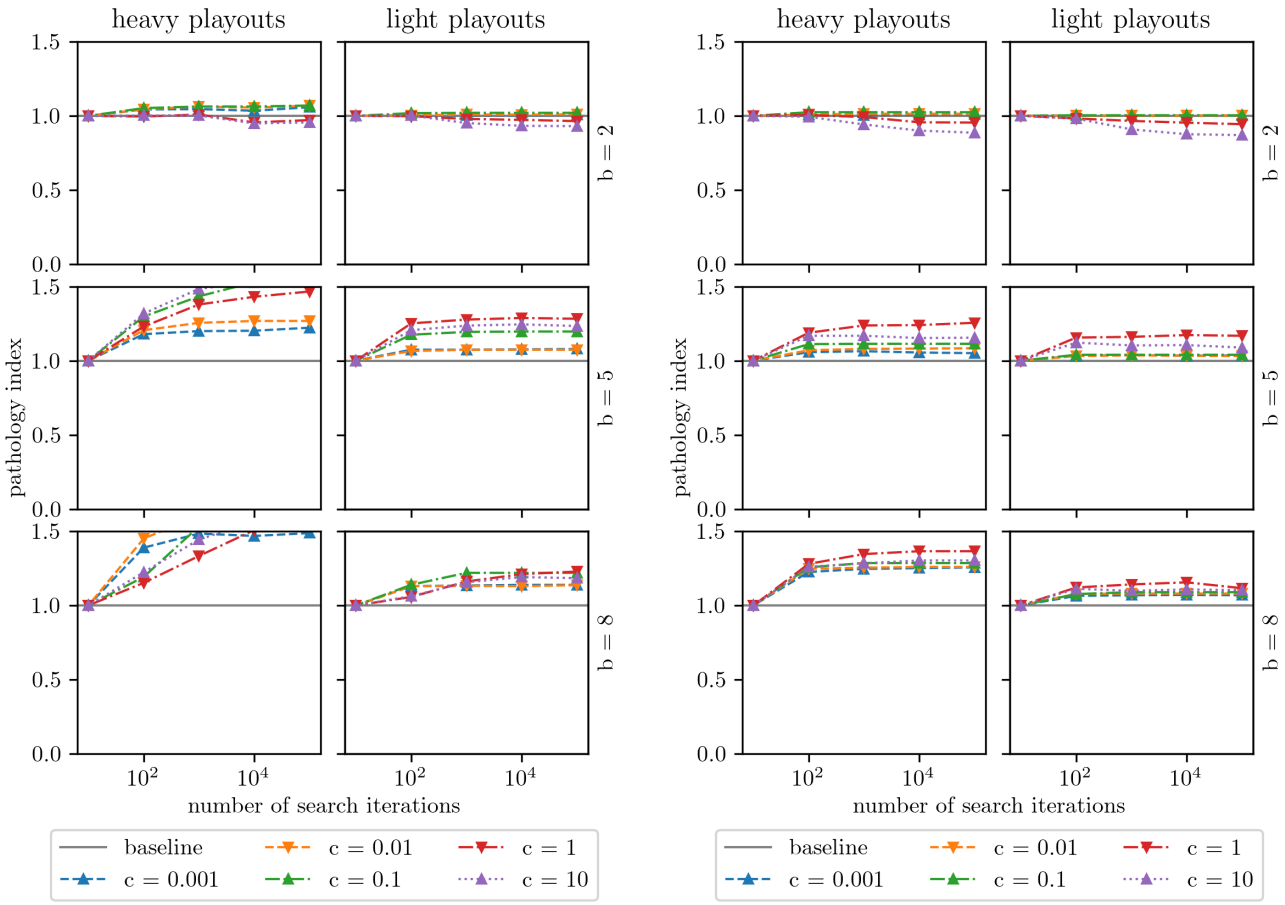

Figure 9: Measuring pathological behavior in UCT on critical win-loss games of depth 50 with $\gamma = 0.5$. The pair of plots on the left correspond to using a heuristic constructed from histograms of Stockfish evaluations of Chess positions sampled at depth 10, using both light and heavy playouts. The pair of plots on the right correspond to using a heuristic constructed from histograms of Edax evaluations of Othello positions sampled at depth 10, using both light and heavy playouts. Each colored line corresponds to an instantiation of UCT with a different exploration constant. The x-axis is plotted on a log-scale. We note the near complete absence of lookahead pathology in this low γ regime.

H Investigating Pathology with Othello-Derived Heuristics

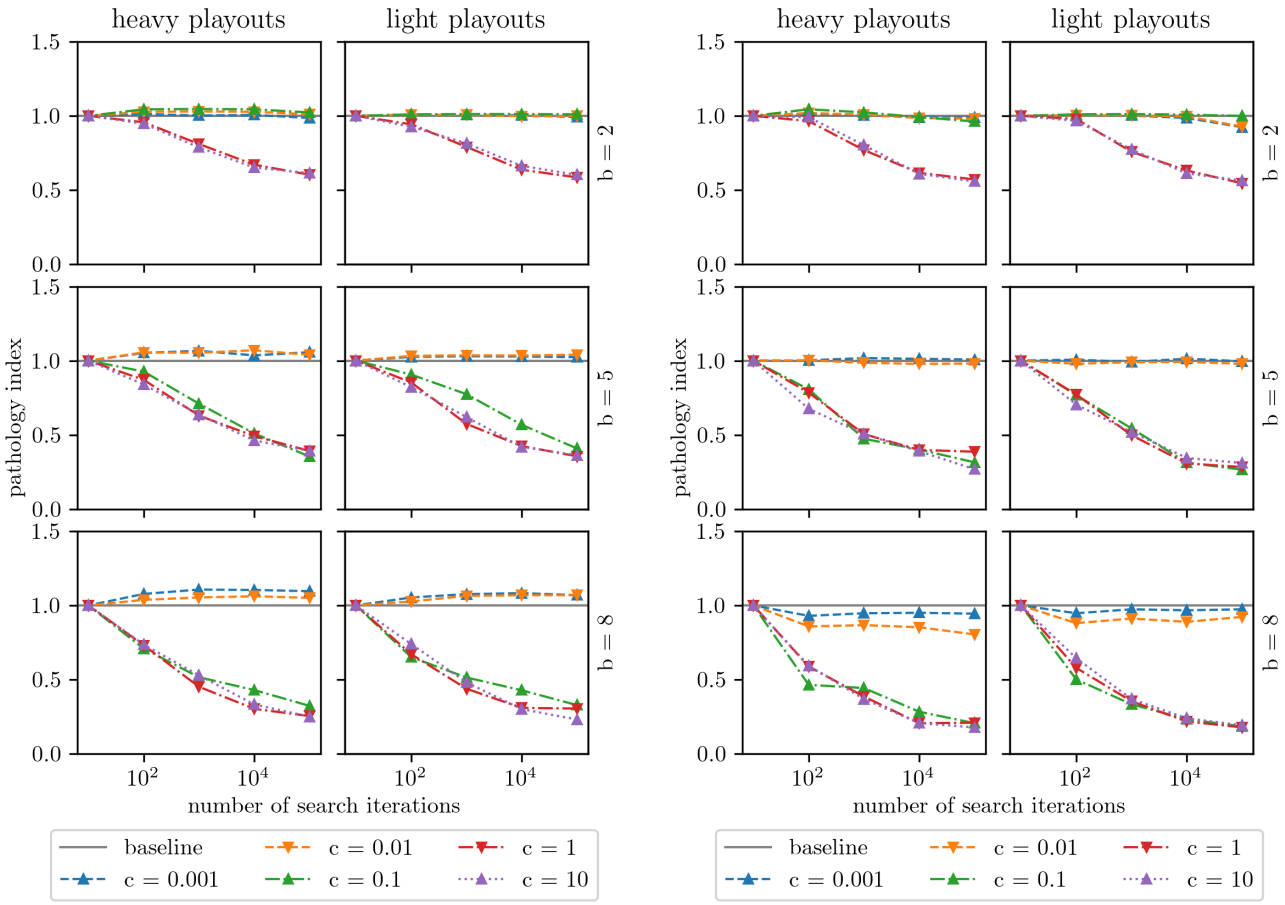

Figure 10: Measuring pathological behavior in UCT on critical win-loss games of depth 50 with $\gamma = 0.9$ (left) and $\gamma = 1$ (right). The heuristic to guide UCT is constructed from histograms of Edax evaluations of Othello positions sampled at depth 10, using both light and heavy playouts. Each colored line corresponds to an instantiation of UCT with a different exploration constant. The x-axis is plotted on a log-scale. We note that aside from some minor exceptions, pathological behavior generally persists even when the heuristic is sourced from a different domain.

I Investigating Pathology when Using True Node Utilities

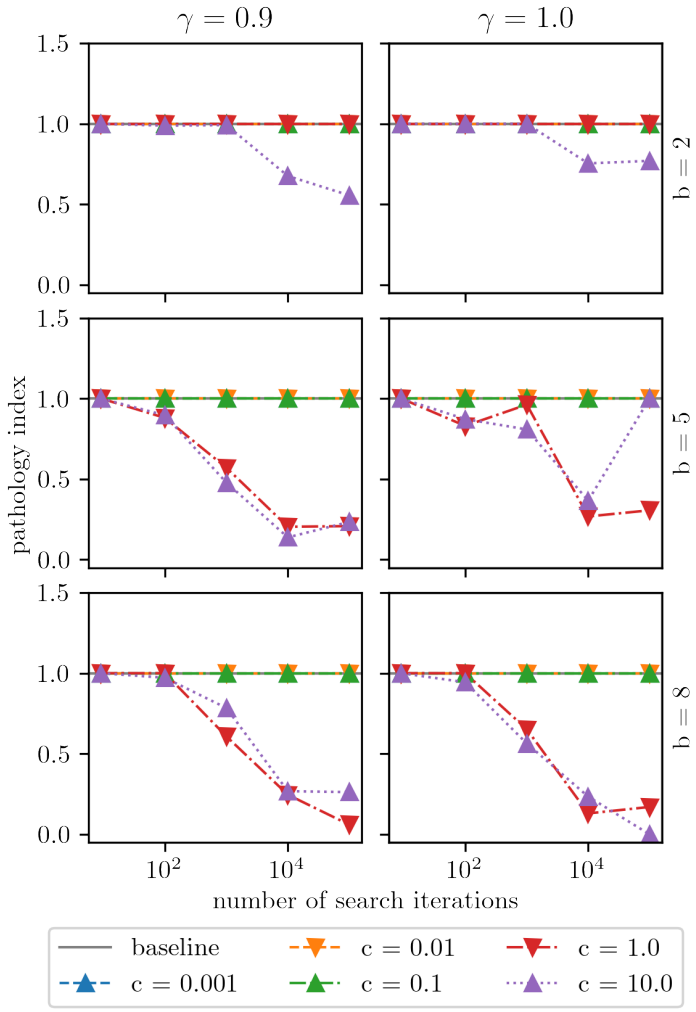

Figure 11: Measuring pathological behavior in UCT on critical win-loss games of depth 10^9 with $\gamma = 0.9$ (left) and $\gamma = 1$ (right). We do not use a heuristic to guide UCT in this instance, instead relying on the true utility of each node. Each colored line corresponds to an instantiation of UCT with a different exploration constant. The x-axis is plotted on a log-scale. We note that lookahead pathology arises in the high-exploration regime even with access to the true game-theoretic values of nodes.

J Investigating Pathology in Alpha-Beta Search

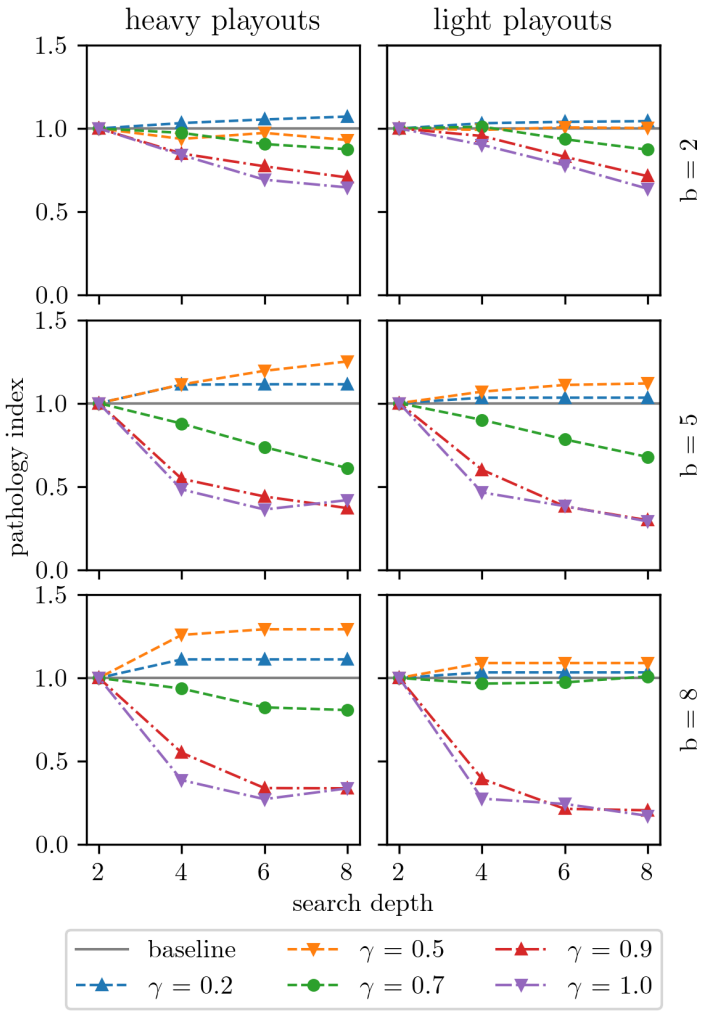

Figure 12: Measuring pathological behavior in alpha-beta search on critical win-loss games. The heuristic to guide the search constructed from histograms of Stockfish evaluations of Chess positions sampled at depth 10, using both light and heavy playouts. The x-axis indicates the depth of the search tree. Each colored line corresponds to a different choice of γ. We note that pathology occurs in a wide range of parameterizations, but is most pronounced for larger branching factors and larger values of γ.