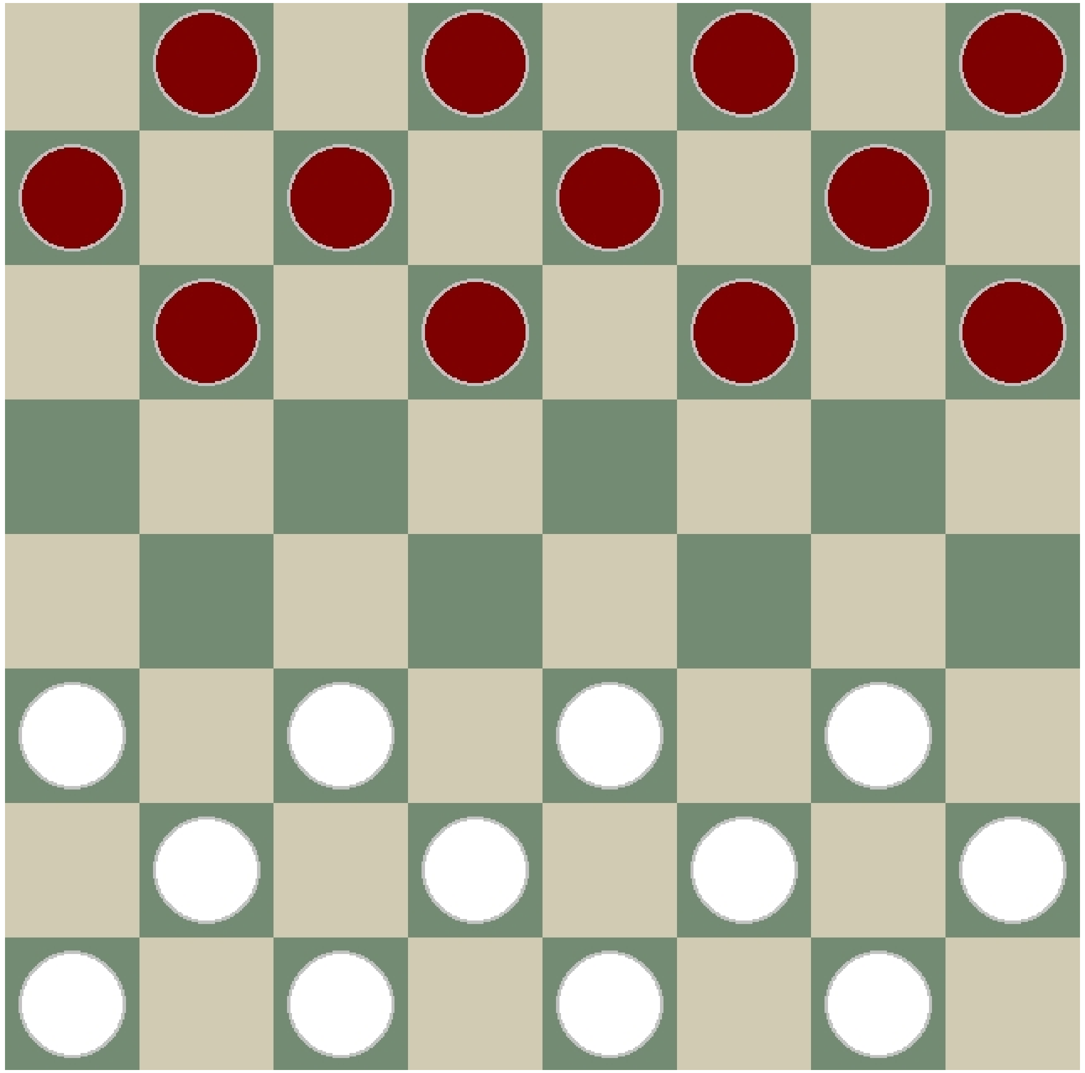
Draught

CS181 Final Project

周守琛 叶柯成 张郅睿 王鹏豪

Settings

- 8*8 grids
- 24 pieces (12 per side)
- 5^32 ≈ 10^22 states



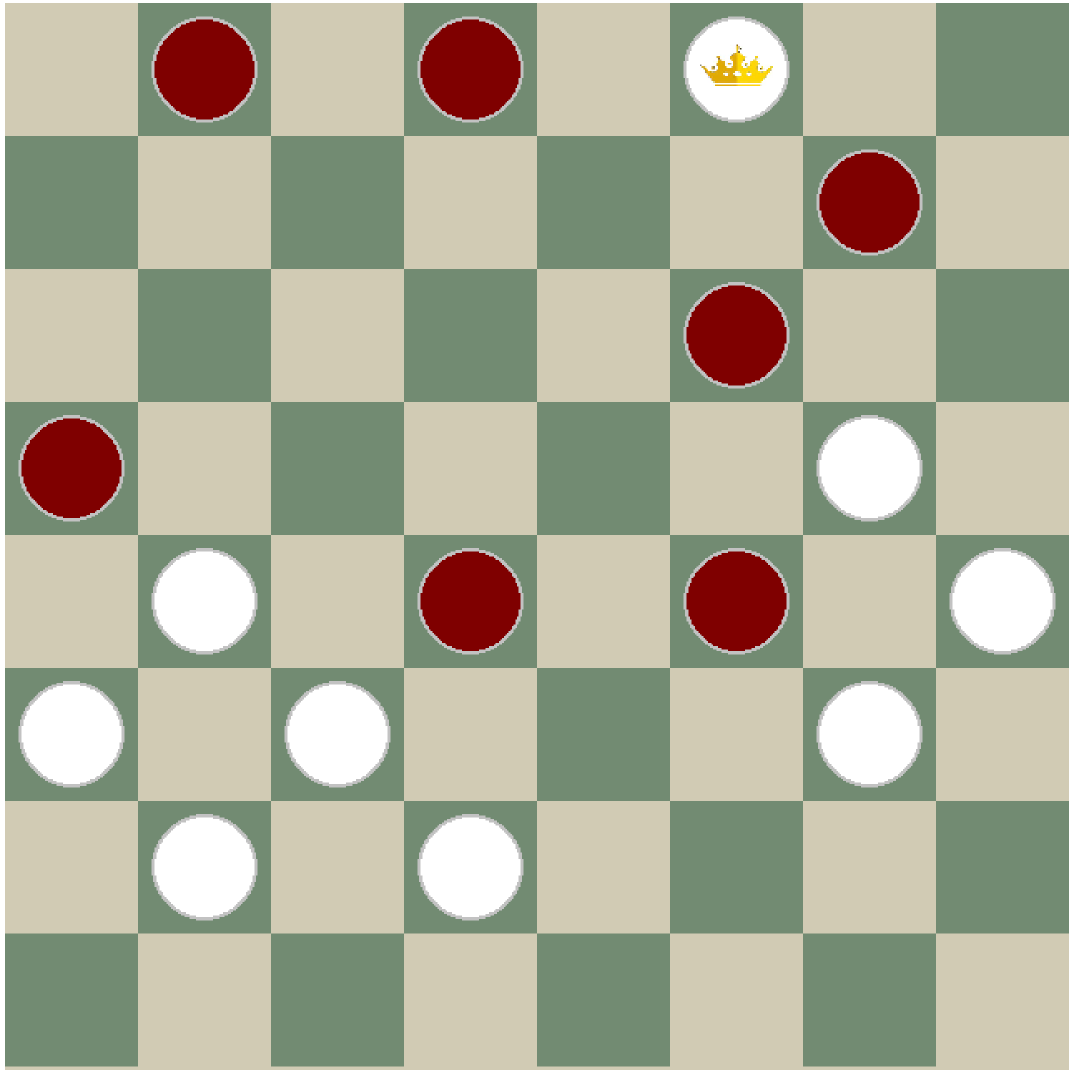
Implement methods

- Random
- Greedy
- Adversarial search (minimax, alpha-beta pruning)
- Reinforcement learning
    - MCTS
    - Q-learning
    - Approximate Q-learning

Score function

$$
\begin{aligned}
f(s) = \omega_1 * (N_{\text{our-survived}} - N_{\text{opponent-survived}}) \\
+ \omega_2 * (N_{\text{our-kings}} - N_{\text{opponent-kings}}) \\
+ \omega_3 * \sum_{\text{our normal pieces}} \frac{1}{L_{\text{dis-to-bottom}}} \\
+ \omega_4 * \sum_{\text{our pieces}} \frac{1}{min(L_{\text{dis-to-left}}, L_{\text{dis-to-right}}) + 1}
\end{aligned}
$$

$$\omega_1 = 1, \omega_2 = 2, \omega_3 = 1, \omega_4 = 0.5$$



Basic methods

- Random
- Greedy
- Adversarial search (minimax, alpha-beta pruning)

Monte-Carlo Tree Search

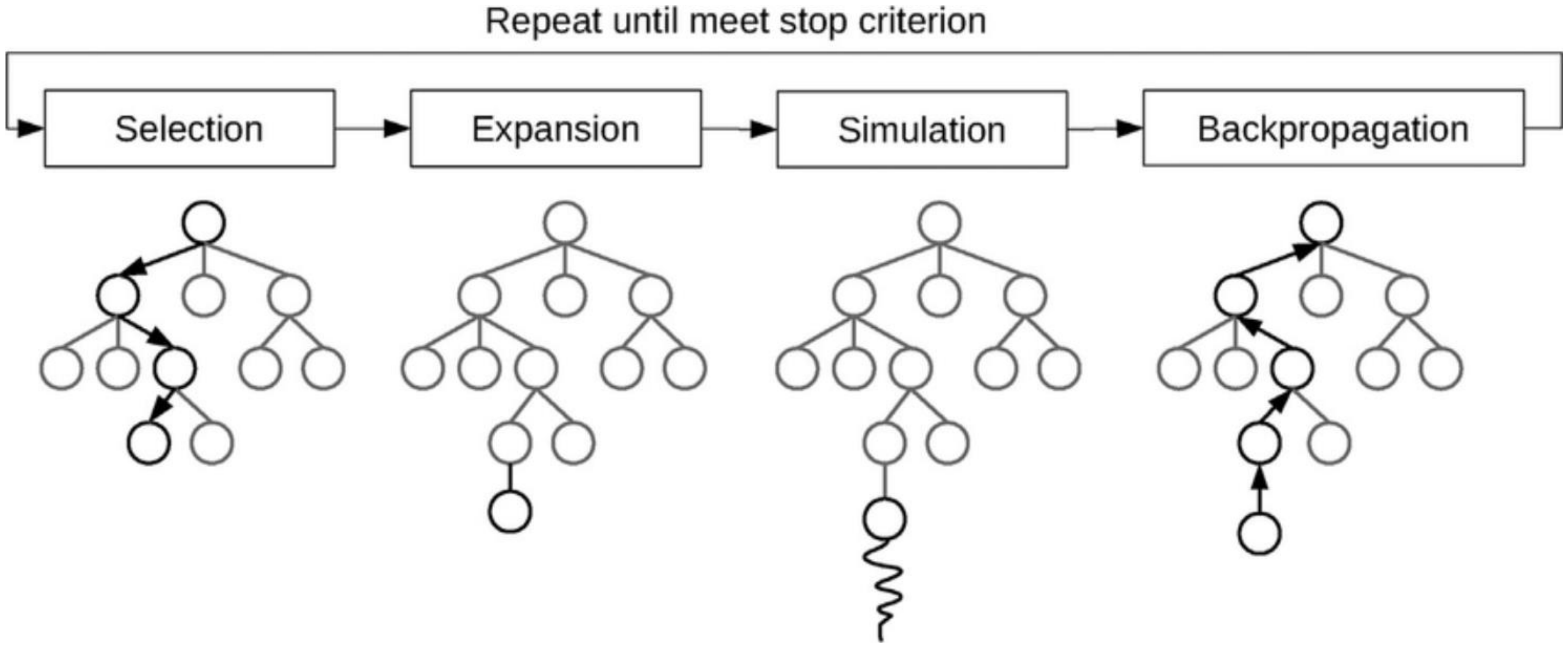
UCT

$$\underset{v' \in \text{children of } v}{\arg\max} \frac{Q(v')}{N(v')} + c\sqrt{\frac{2 \ln N(v)}{N(v')}}$$

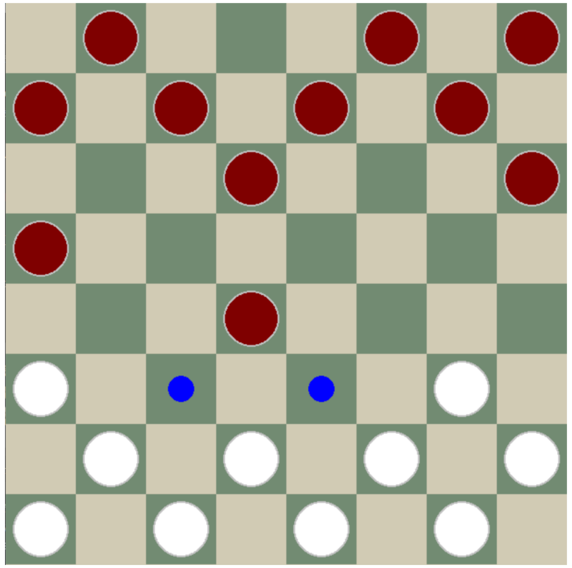
Exploitation

Exploration

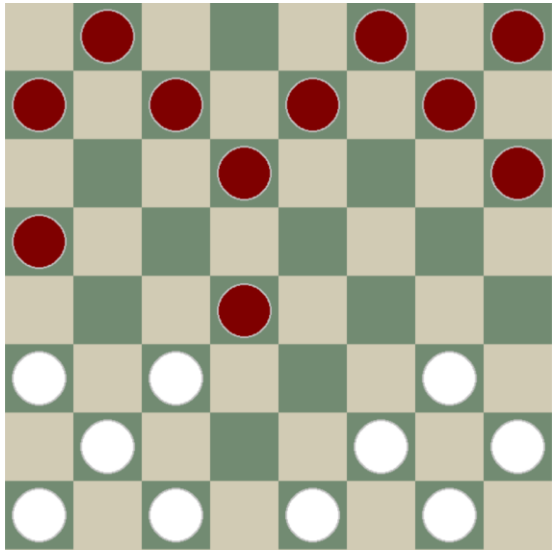
Monte-Carlo Tree Search



Q-learning



S



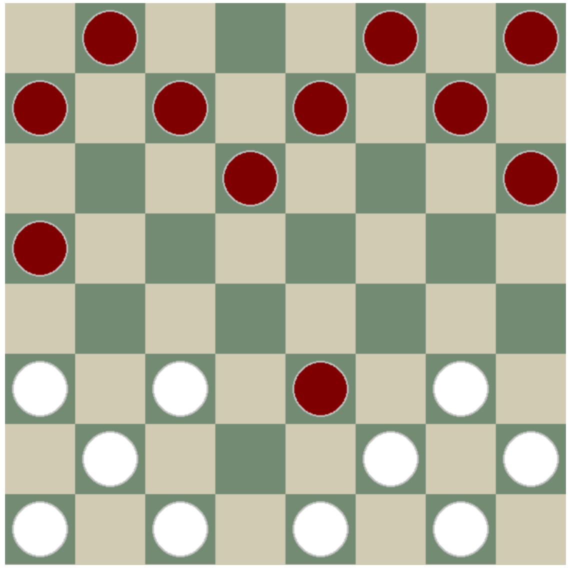
S'

$$r1(s,a,W) = f(s,W) - f(s',W)$$

$$r2(s,a,W) = [f(s,W) - f(s',W)] - [f(s',B) - f(s'',B)]$$

$$sample = r1(s,a,W) + \gamma \max_{a'} Q(s',a')$$

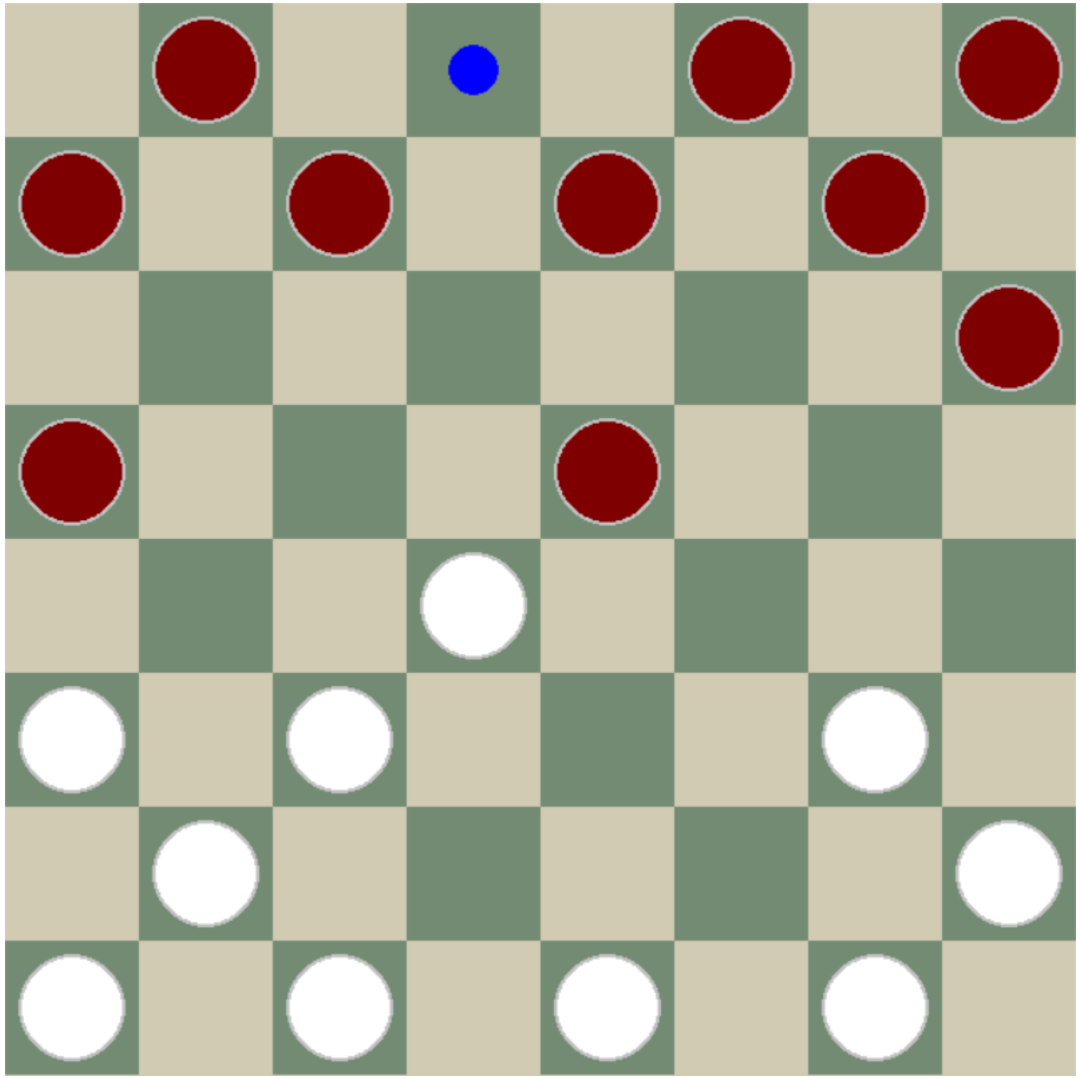
$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(sample)$$



S''

Q-learning

- $\gamma = 0.8, 0.9, 1.0, 1.1$?
- Unvisited Q-value → 0



Approximate Q-learning

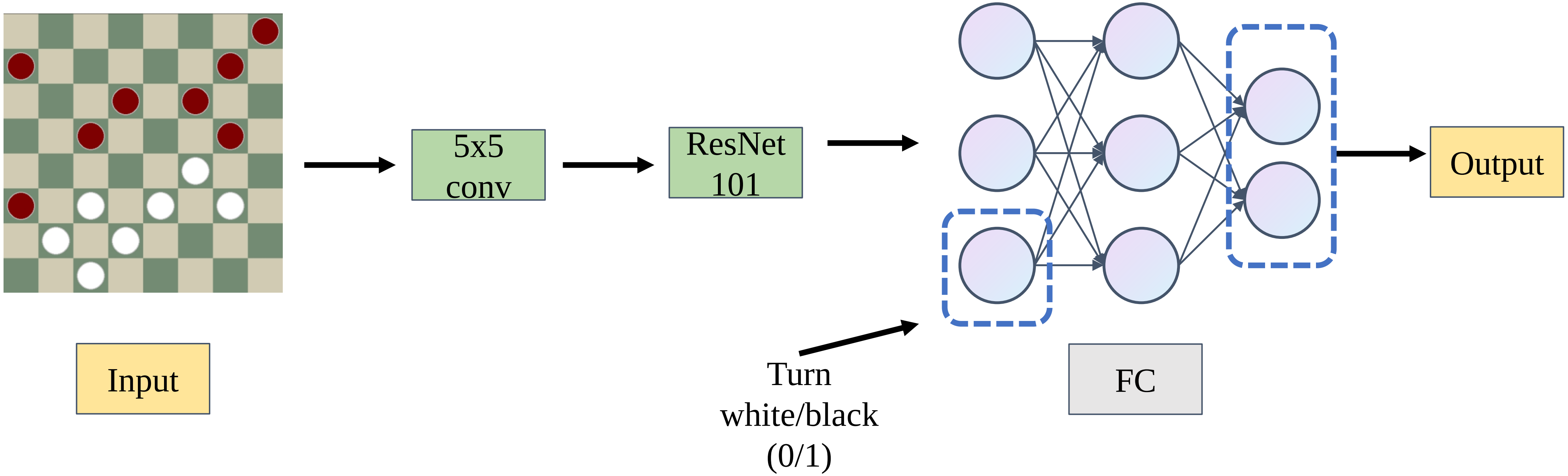
$$
\begin{aligned}
f(s) = \omega_1 * (N_{\text{our-survived}} - N_{\text{opponent-survived}}) & \qquad \longrightarrow \qquad f_1 \\
+ \omega_2 * (N_{\text{our-kings}} - N_{\text{opponent-kings}}) & \qquad \longrightarrow \qquad f_2 \\
+ \omega_3 * \sum_{\text{our normal pieces}} \frac{1}{L_{\text{dis-to-bottom}}} & \qquad \longrightarrow \qquad f_3 \\
+ \omega_4 * \sum_{\text{our pieces}} \frac{1}{min(L_{\text{dis-to-left}}, L_{\text{dis-to-right}}) + 1} & \qquad \longrightarrow \qquad f_4
\end{aligned}
$$

$$difference = sample - Q(s, a)$$

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha(sample)$$

$$w_i \leftarrow w_i + \alpha(difference) f_i(s, a)$$

Approximate Q-learning



5x5 conv

ResNet 101

Input

Turn white/black (0/1)

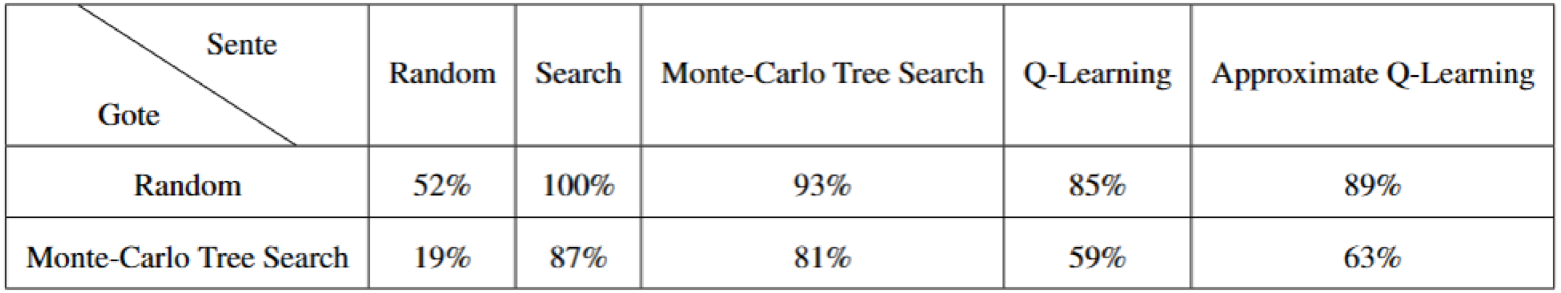
FC

Output

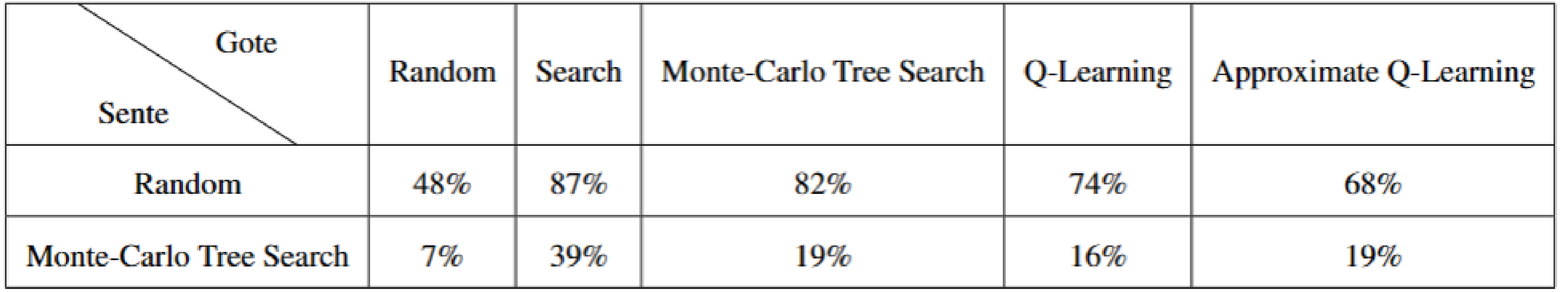
f_5 : 2-classification {+1,-1}

Results

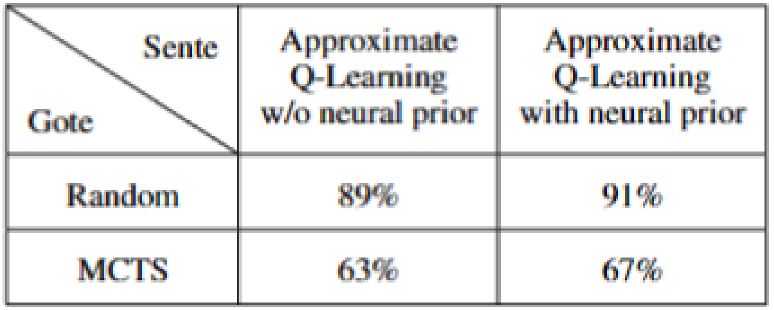
Sente:先手
Gote :后手



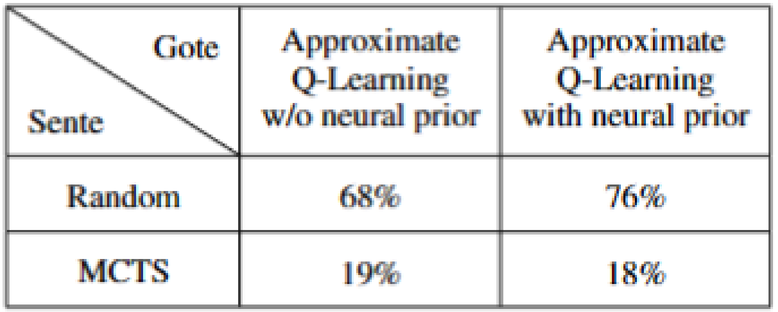
Sente Gote	Random	Search	Monte-Carlo Tree Search	Q-Learning	Approximate Q-Learning
Random	52%	100%	93%	85%	89%
Monte-Carlo Tree Search	19%	87%	81%	59%	63%



Gote Sente	Random	Search	Monte-Carlo Tree Search	Q-Learning	Approximate Q-Learning
Random	48%	87%	82%	74%	68%
Monte-Carlo Tree Search	7%	39%	19%	16%	19%



Sente Gote	Approximate Q-Learning w/o neural prior	Approximate Q-Learning with neural prior
Random	89%	91%
MCTS	63%	67%



Gote Sente	Approximate Q-Learning w/o neural prior	Approximate Q-Learning with neural prior
Random	68%	76%
MCTS	19%	18%

Thank you