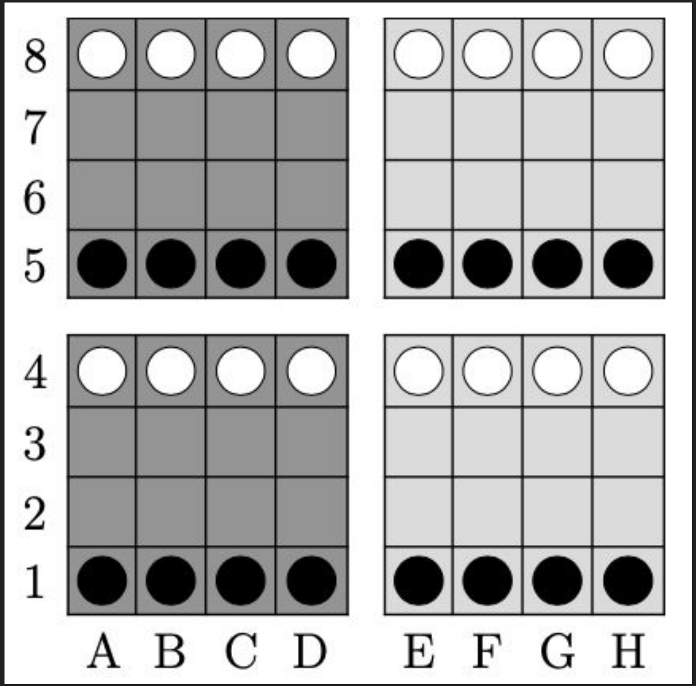
ShabuShabu: Learning Shobu By Self-Play Through Reinforcement Learning

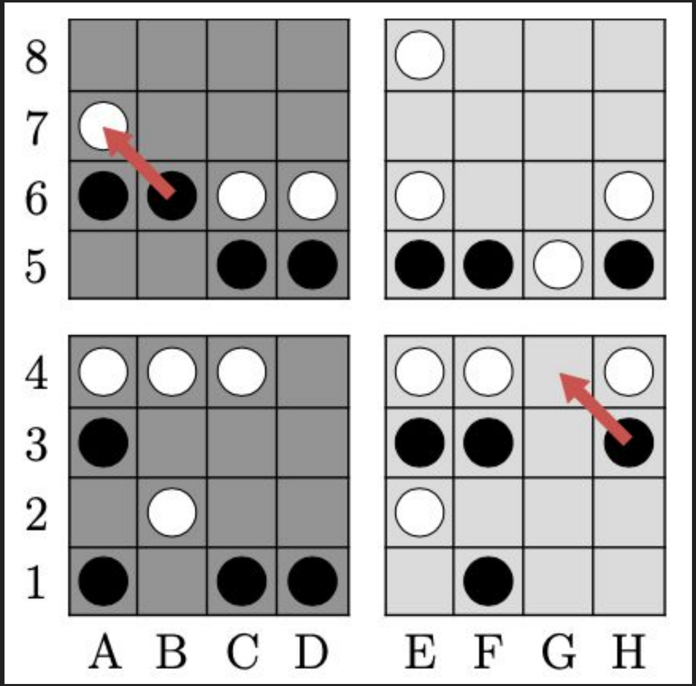
Brandon Gong, Ayushman Choudhury, Seowon Chang
CSCI 1470
30.04.2025

Overview

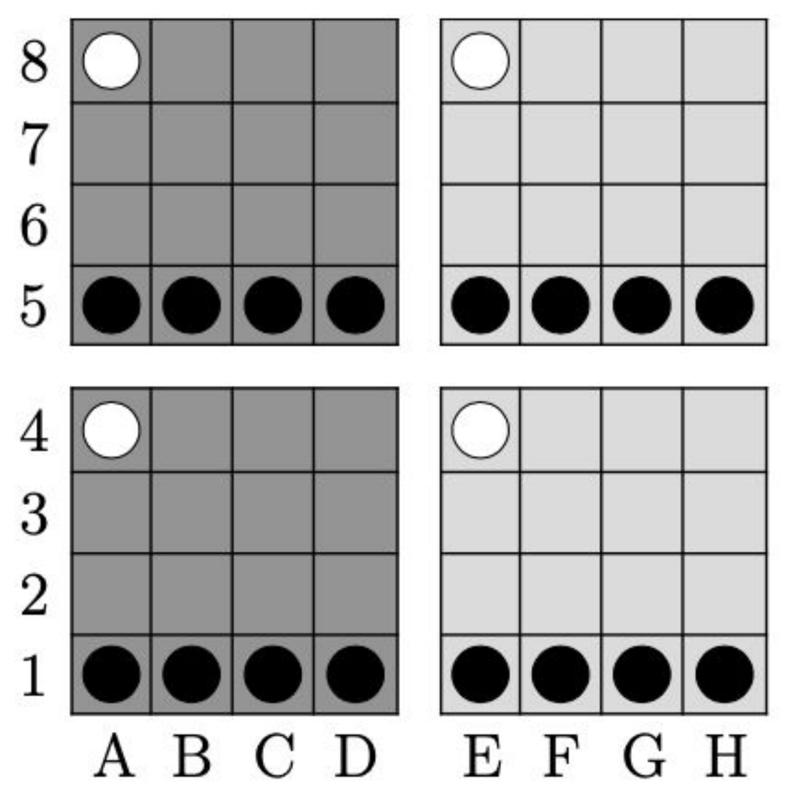
- What is Shobu?
- Methodology
  - Monte Carlo Tree Search
  - ShabuShabu architecture
  - Training
  - Evaluation metrics
  - Challenges
- Results
- Discussion
- Interactive Demo (Experimental)

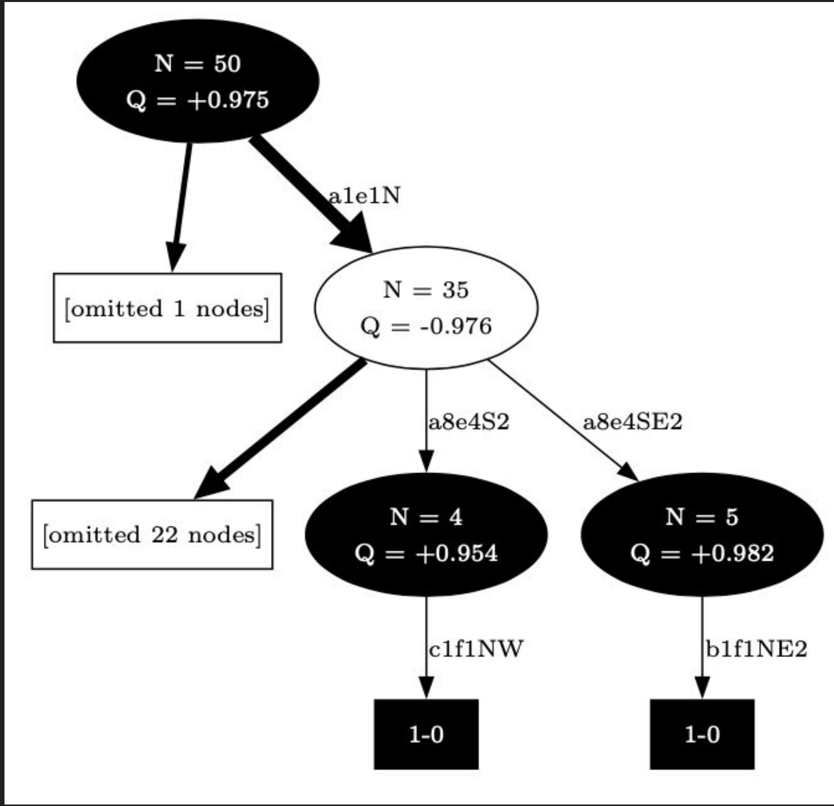
Shobu



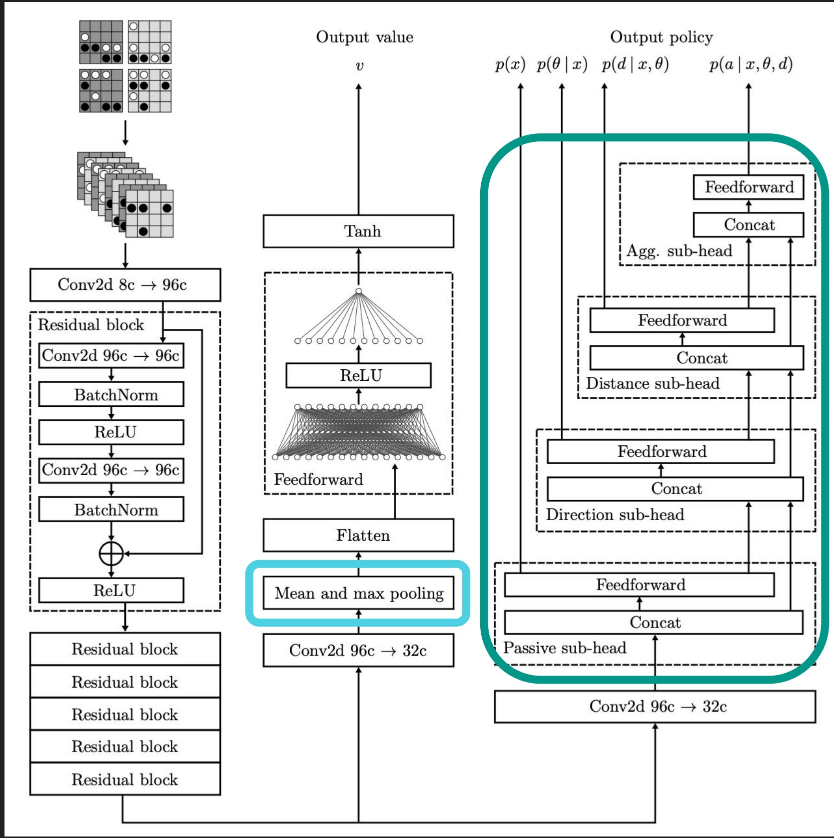


Monte Carlo Tree Search





ShabuShabu Architecture



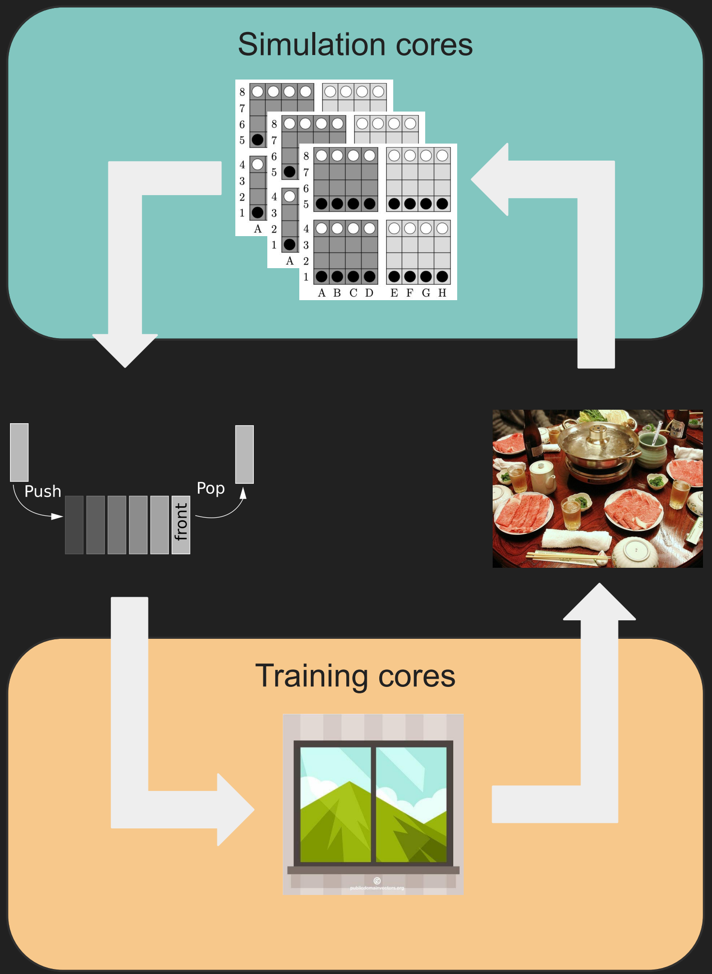
Loss Function

$$L(s) = \boxed{\beta_{value} * (\hat{v}(s) - r(s))^2} \boxed{- \sum_m \pi(m) \log(\hat{\pi}(m))} + \boxed{\beta_{L2} ||\theta||^2}$$

- Value loss: predicting the expected game outcome of a state
- Policy loss: predicting the probable “good moves” of a state
- L2 penalty

Training Optimizations

- AlphaZero: 5,064 TPUs. Us: 64 CPUs.
- Parallelized game generation and training
  - Games are generated in parallel with training, where training data is sampled from the 50K most recent game states
  - Shobu games last between 15-25 full-ply moves, so this is roughly 1K-1.6K most recent games



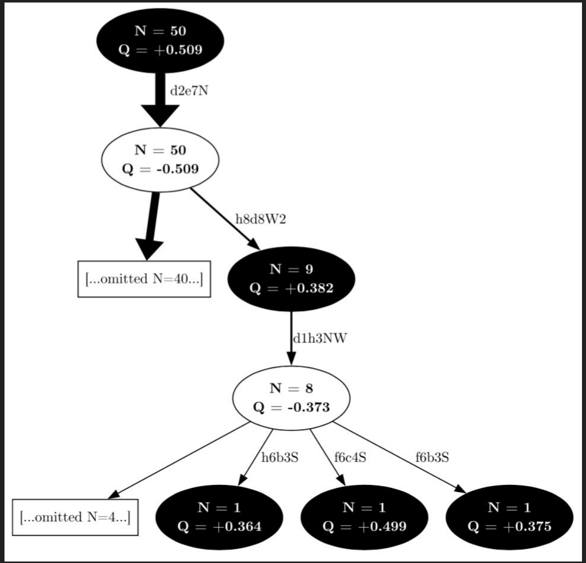
Training Optimizations (cont'd)

- Playout cap randomization
  - Randomly perform shorter depth searches to generate more games
  - Allows for more game results for the value head to train on
- Starting games from random positions
  - Allows the model to see more varied positions
  - Also allows for more game results since these games will usually finish quicker

Evaluation Metrics: Qualitative

`explorer`: interactive REPL to run MCTS and view model outputs

- Adjust hyperparameters during training to induce/reduce exploration
- Evaluate model outputs and MCTS behavior on test board states



Evaluation Metrics: Quantitative

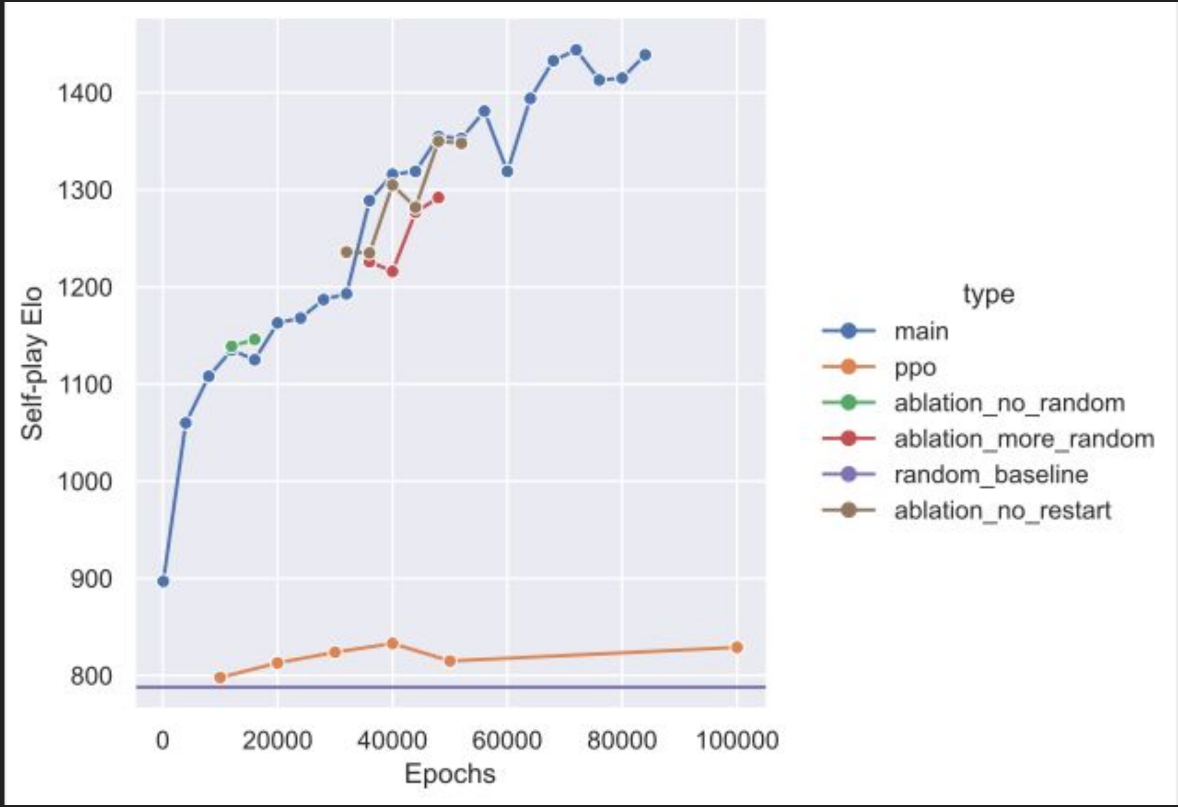
We employ two different quantitative metrics:

- **Elo**: metric that measures relative strength between players
- Win-draw-loss (**WDL**) from matches between two players

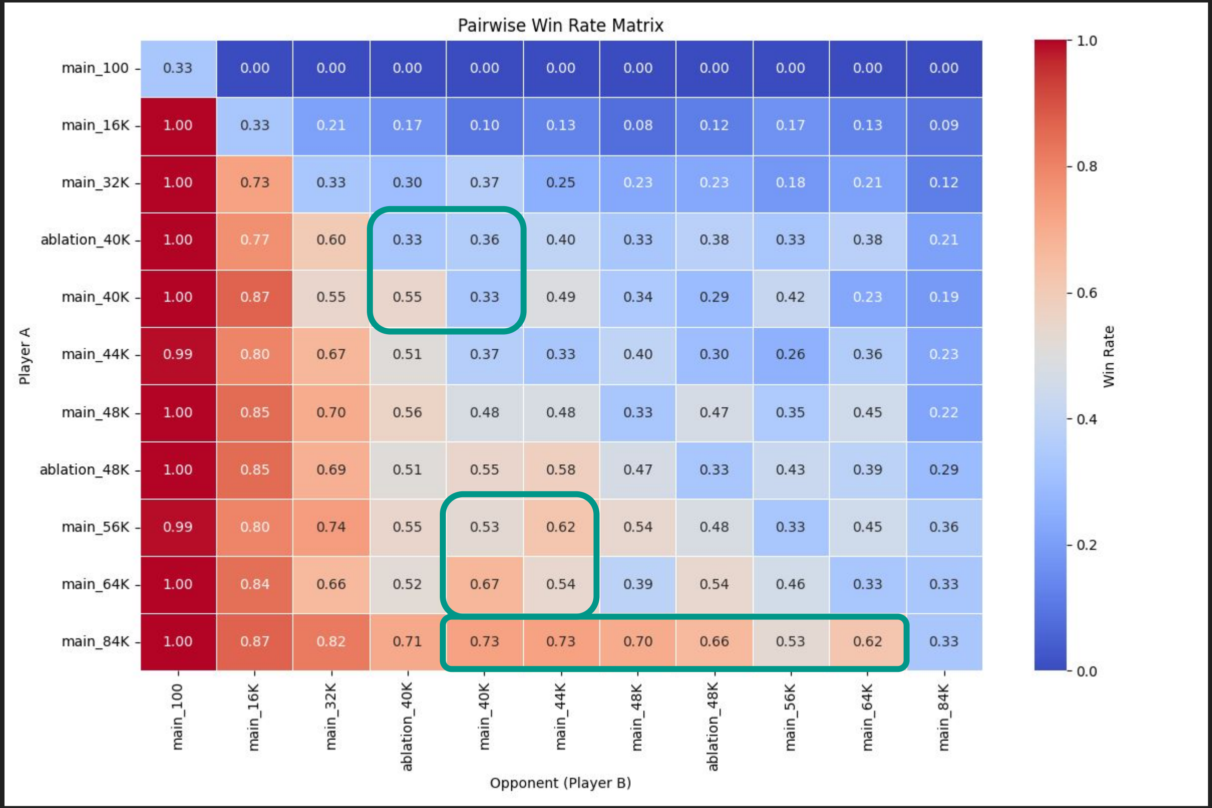
We perform the following experiments:

- We have different model checkpoints play each other with randomized openings to calculate Elos across training time
- We observe the direct WDL between agents to identify more subtle behaviors

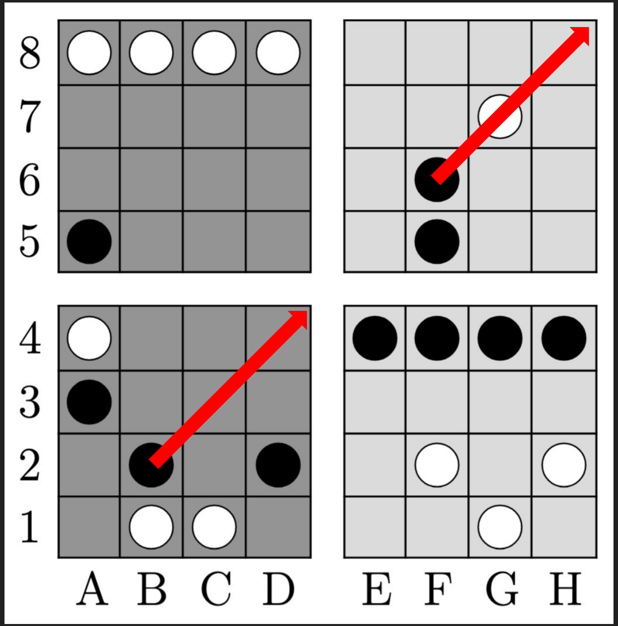
Results: Self-play Elo Across Training



Results: H2H performance



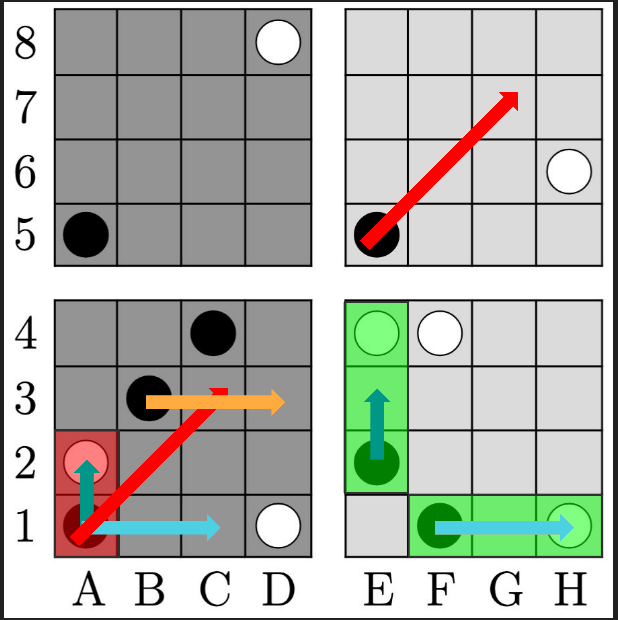
Results: individual examples of `explorer`



Can you spot the mate?

```
> search 1000
> children
✅ b2f6NE2: W(visits: 988, avg_reward: −1.000, prior: +0.063, value: None)
  a3f5E2: W(visits:   4, avg_reward: −0.280, prior: +0.100, value: +0.403)+
 b2f5NE2: W(visits:   3, avg_reward: −0.073, prior: +0.096, value: +0.505)+
   b2f6N: W(visits:   1, avg_reward: +0.590, prior: +0.007, value: +0.590)+
  b2f5NE: W(visits:   1, avg_reward: +0.582, prior: +0.030, value: +0.582)+
  a3f5NE: W(visits:   1, avg_reward: +0.354, prior: +0.042, value: +0.354)+
   a3f5E: W(visits:   1, avg_reward: +0.424, prior: +0.031, value: +0.424)+
  a3f6E2: W(visits:   1, avg_reward: +0.514, prior: +0.066, value: +0.514)+
   d2f6N: W(visits:   0, avg_reward: +0.000, prior: +0.008, value: None)
  b2f6N2: W(visits:   0, avg_reward: +0.000, prior: +0.024, value: None)
```

Results: individual examples of `explorer`



What would you do?

```
> c
a1e5NE2: W(visits: 902, avg_reward: -0.688, prior: +0.057, value: -0.480)+
 e2a5NE: W(visits:  31, avg_reward: -0.571, prior: +0.045, value: -0.480)+
e2a5NE2: W(visits:  31, avg_reward: -0.564, prior: +0.100, value: +0.305)+
f1a5NE2: W(visits:  12, avg_reward: -0.418, prior: +0.093, value: +0.012)+
 f1a5NE: W(visits:   6, avg_reward: -0.446, prior: +0.042, value: -0.112)+
 a1e5NE: W(visits:   5, avg_reward: -0.532, prior: +0.026, value: -0.317)+
 a1f1E2: W(visits:   2, avg_reward: +0.025, prior: +0.031, value: -0.181)+
 e2a5E2: W(visits:   2, avg_reward: +0.113, prior: +0.069, value: +0.191)+
  b3f1N: W(visits:   1, avg_reward: +0.566, prior: +0.004, value: +0.566)+
 f1a5N2: W(visits:   1, avg_reward: +0.465, prior: +0.030, value: +0.465)+
```

Discussion

- Challenges/Limitations
  - Working with limited compute
  - Diagnosing/preventing model overfitting
- Future Work
  - Increase model complexity
  - Predicting win/draw/loss probability rather a scalar
  - Remove passive move dependency on aggressive moves
  - Does incorporating past moves or additional state information improve learning?
  - Training data augmentation

References

[1] Arpad Elo. The rating of chessplayers, past and present, 1986.
[2] Center for Computation and Brown University Visualization. Oscar. https://docs.ccv.brown.edu/oscar.
[3] Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization-an empirical study on continuous control. In International Conference on Learning Representations, 2020.
[4] Jamie Sajdak and Manolis Vranas. Shobu. https://www.smirkanddagger.com/product-page/shobu, 2019. Accessed April 14, 2025.
[5] A. L. Samuel. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3):210–229, 1959.
[6] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. CoRR, abs/1707.06347, 2017.
[7] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den 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, 529:484–503, 2016.
[8] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.
[9] David J. Wu. Accelerating self-play learning in go, 2020.

(Highly experimental) Try playing against ShabuShabu!

- Try Random Bot out to learn how the game works
- Smaller number = less training = weaker model
- Let us know if you can win against ShabuShabu-84k – we can't!

bgong.local:1470

bgong.local:1470

Questions?