

A Neural Network Chess Engine

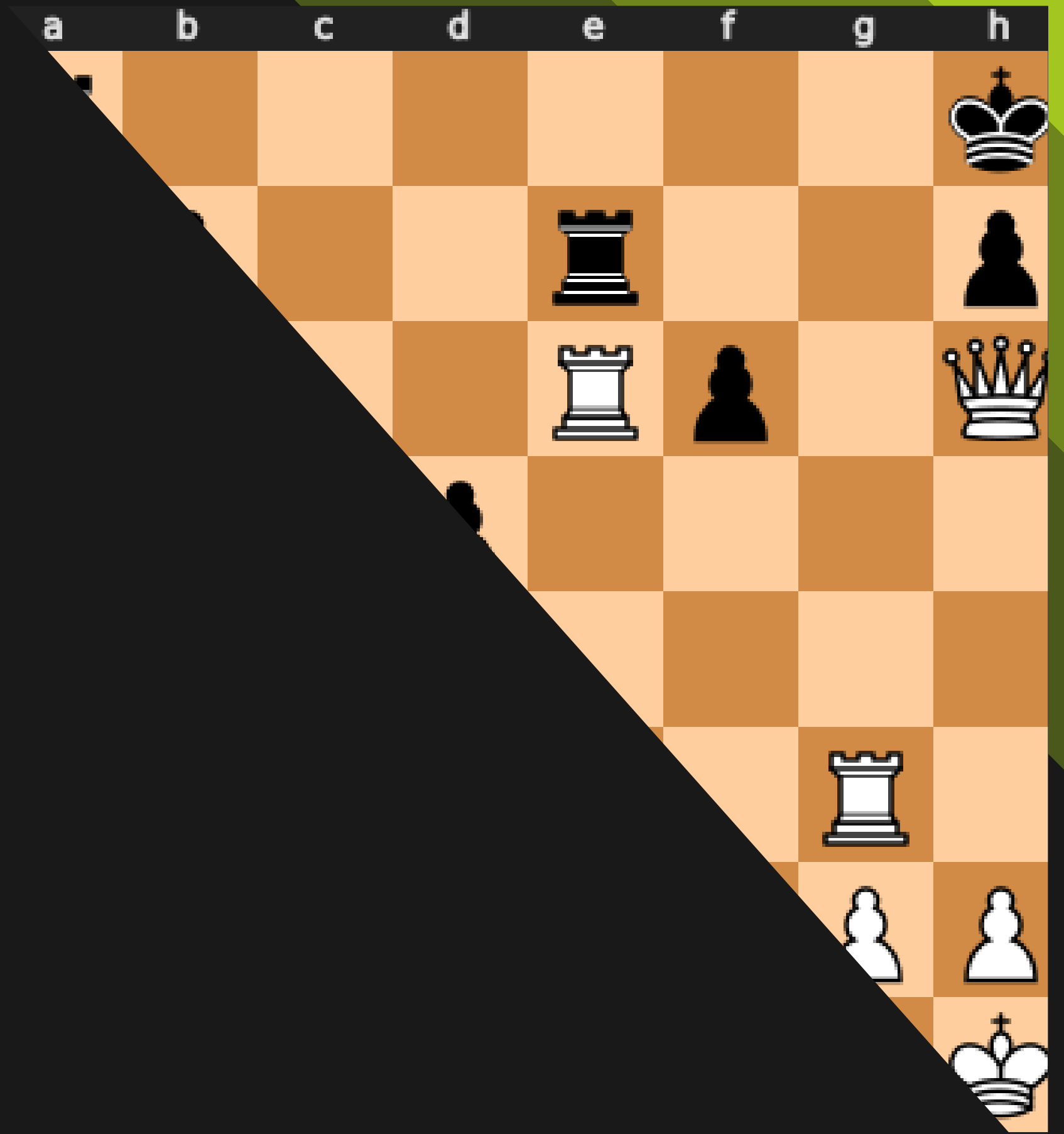
Nathaniel Martin

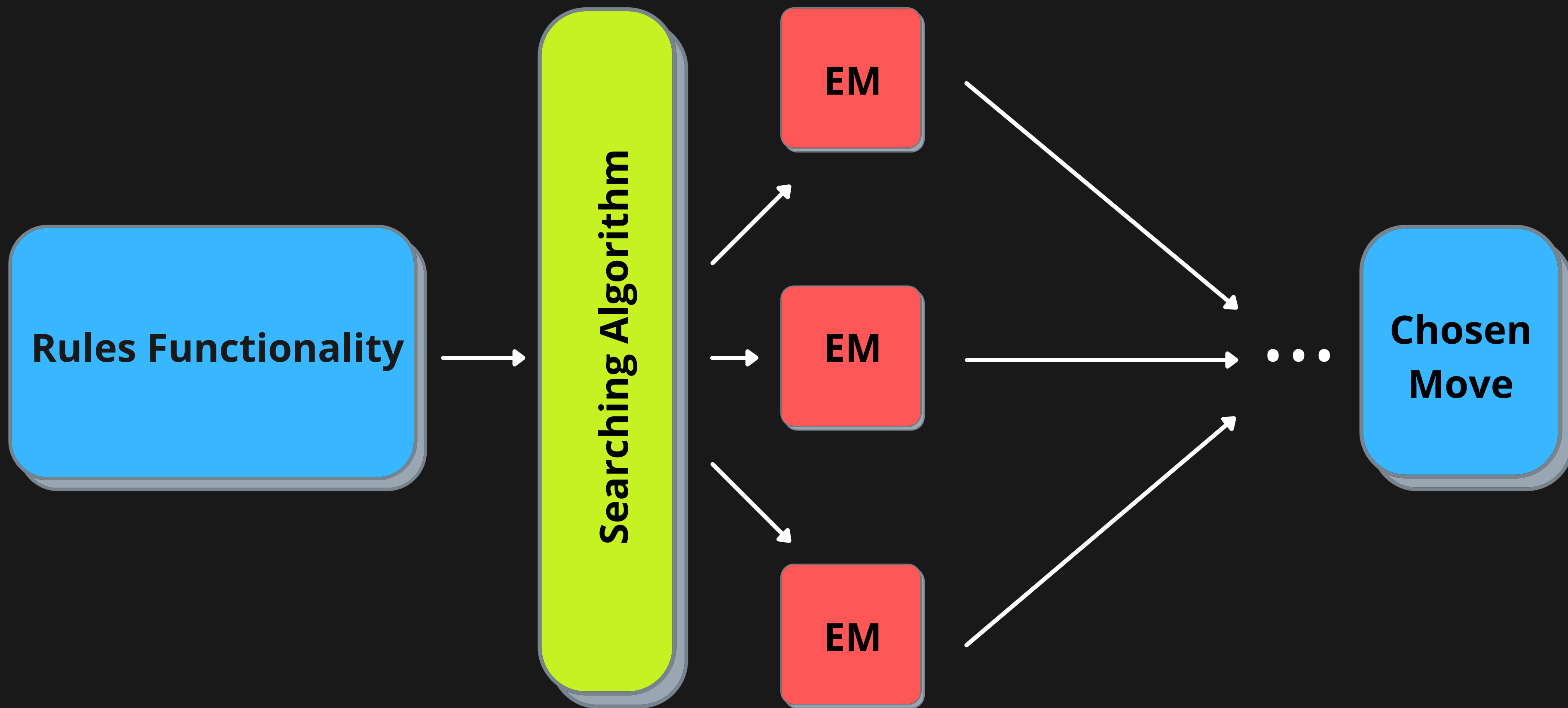


Agenda

- ▶ Engine Architecture
- ▶ Data Preparation
- ▶ Model Construction
- ▶ Performance

Engine Architecture





Data Preparation

Lichess Puzzle Data



~ 2.2 million puzzles

All taken from actual games played.



A position has ~ 32 possible moves

This makes the data very imbalanced



"Only Move" Puzzles

This lets us label moves easily.

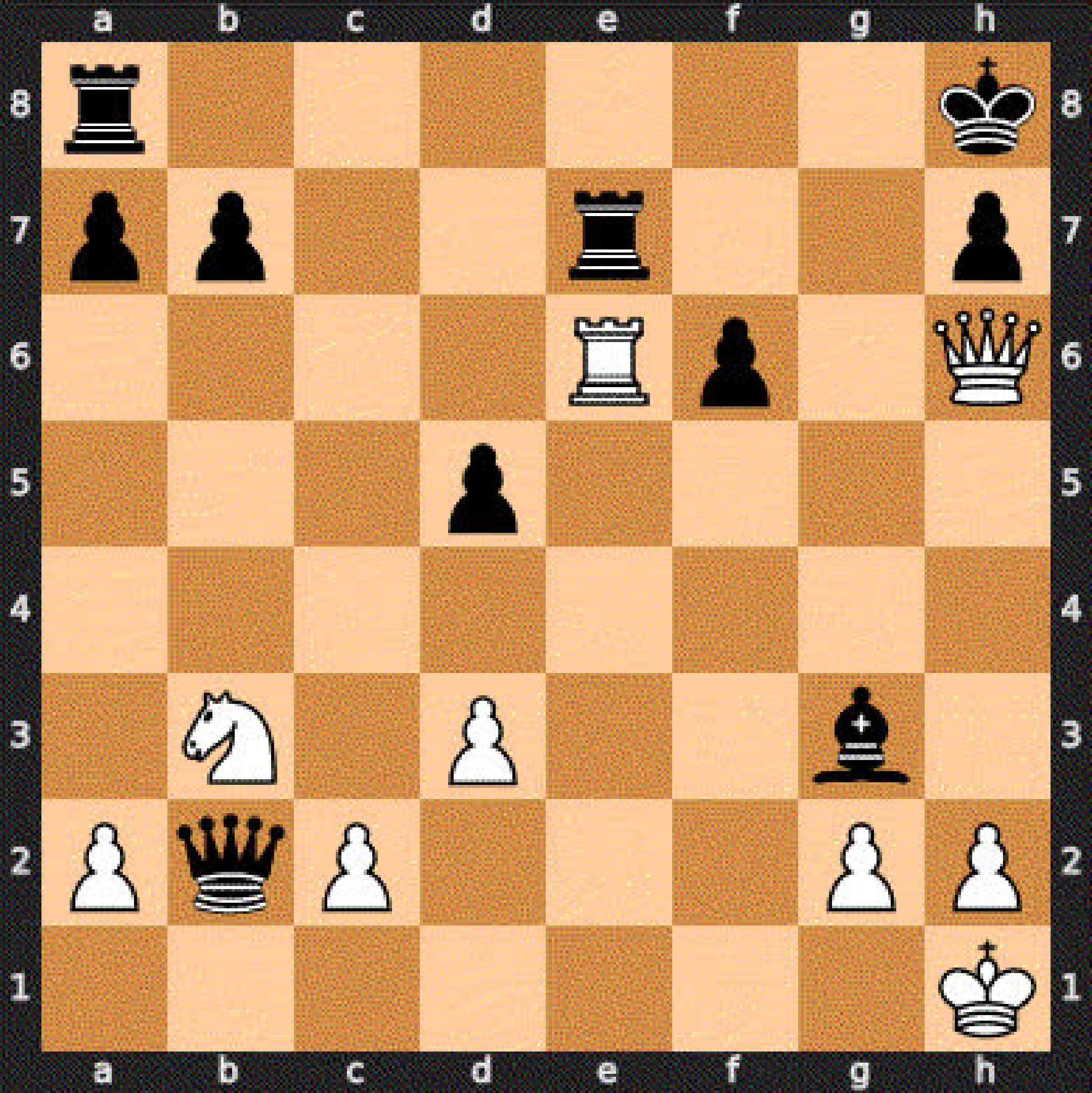


FEN positions and UCI moves

Each entry has a starting FEN and move list. The puzzle position is actually the starting FEN advanced by the first move.

What the Network Gets

- Each "slice" of the board has one piece type.
- This allows each slice to be represented by an 8 x 8 grid of ones and zeros.



Model Construction



Convolutional Neural Network

Typically used in Image Classification

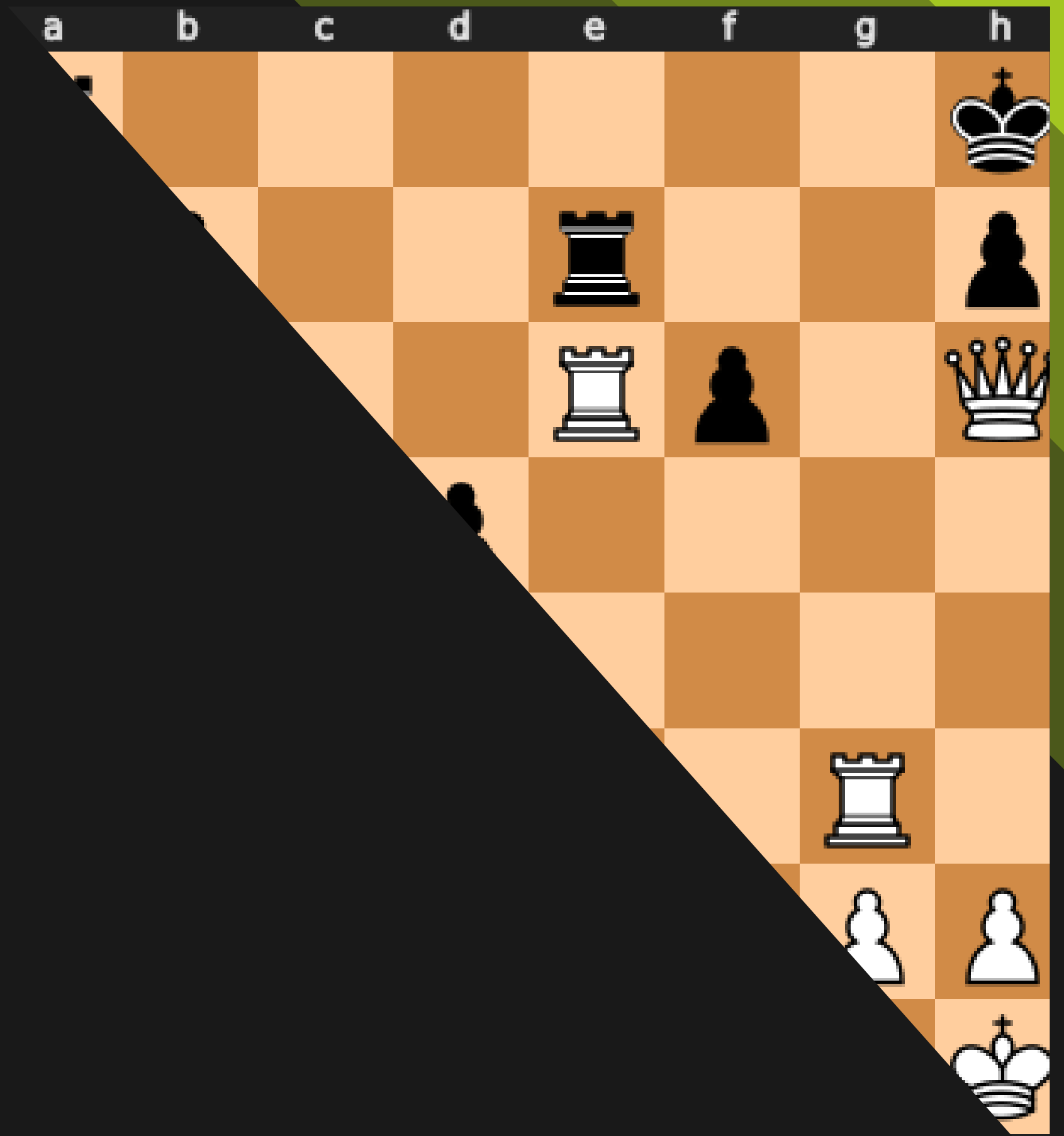
Large Filter

Kernel size of 8

Binary Classification

Thanks to our puzzle data

Performance



Typical Metrics



Accuracy

Evaluation metrics need to give as accurate as possible scores to the search function.



Naive Imbalanced: ~96%

Notably this is slightly *worse* than guessing that every move is bad, given 32 moves on average.



Longer Training, and Imbalance change: ~90%

With 1 in 5 positions being good moves, this is doing better than a dumb model.

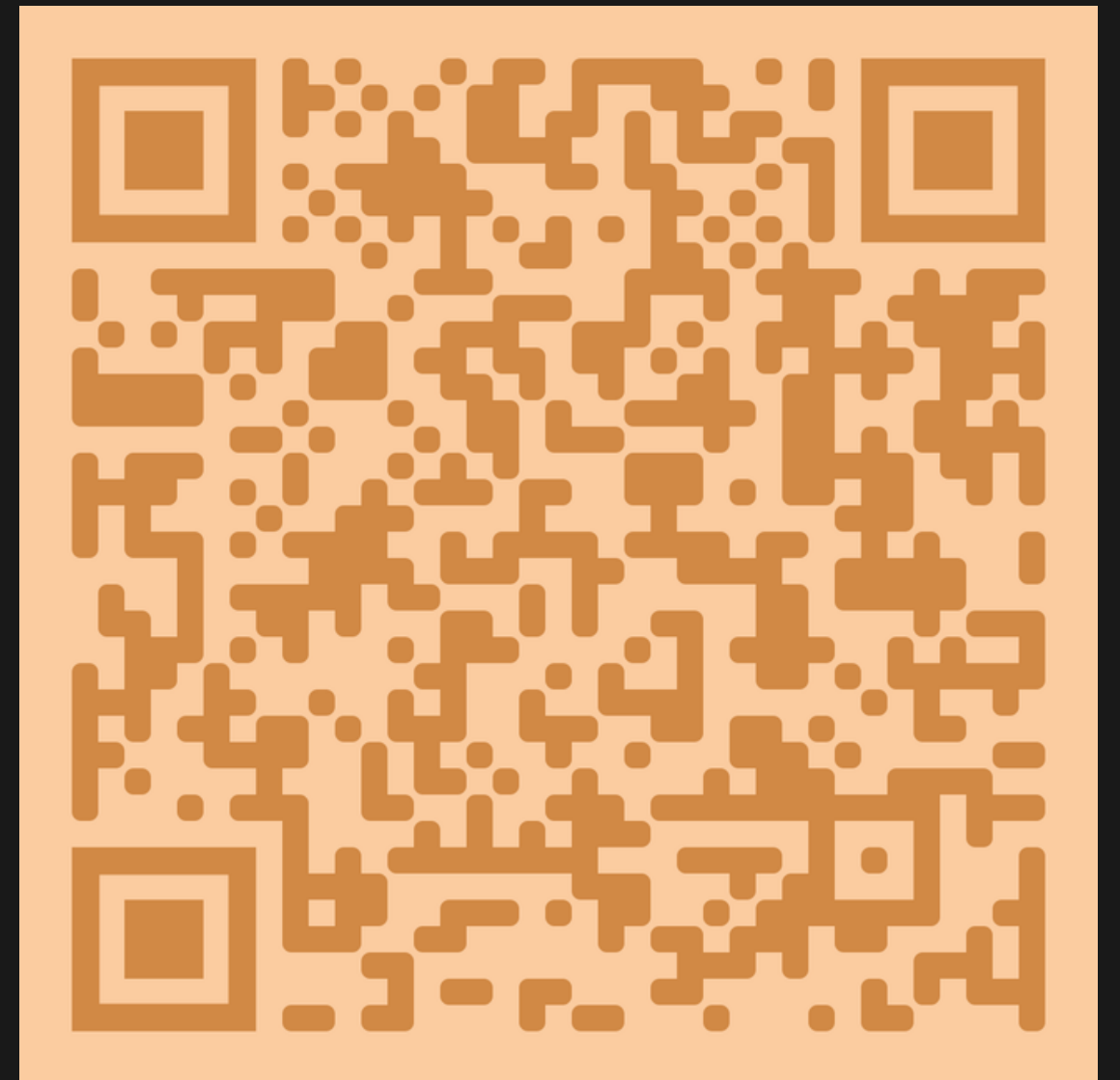
Thank You!



View the project
Tensorboard:



Play the bot on Lichess:



www.linkedin.com/in/nathaniel-martin-73b037227

<https://github.com/UpGoerFive>