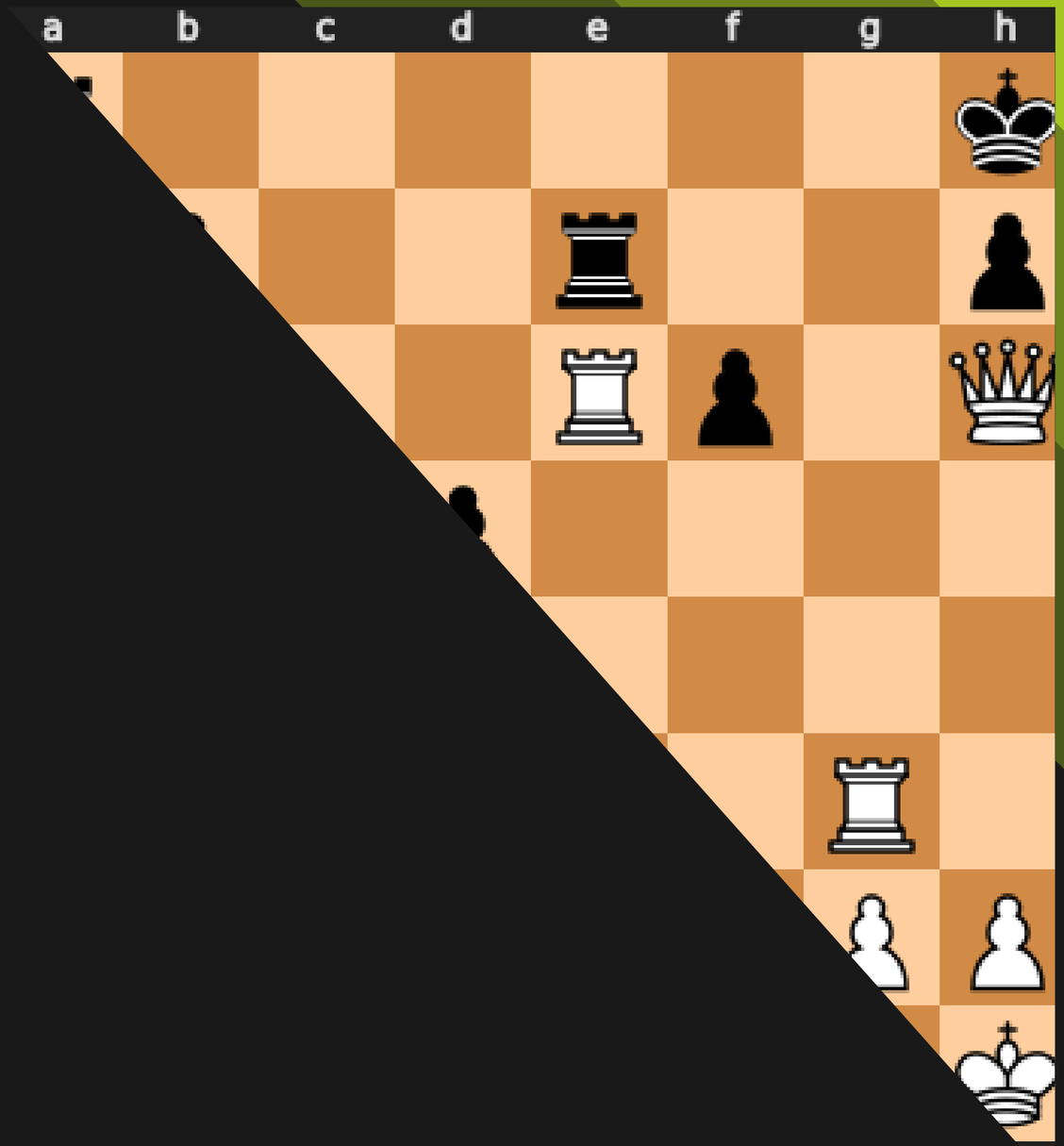
A CNN Chess Engine

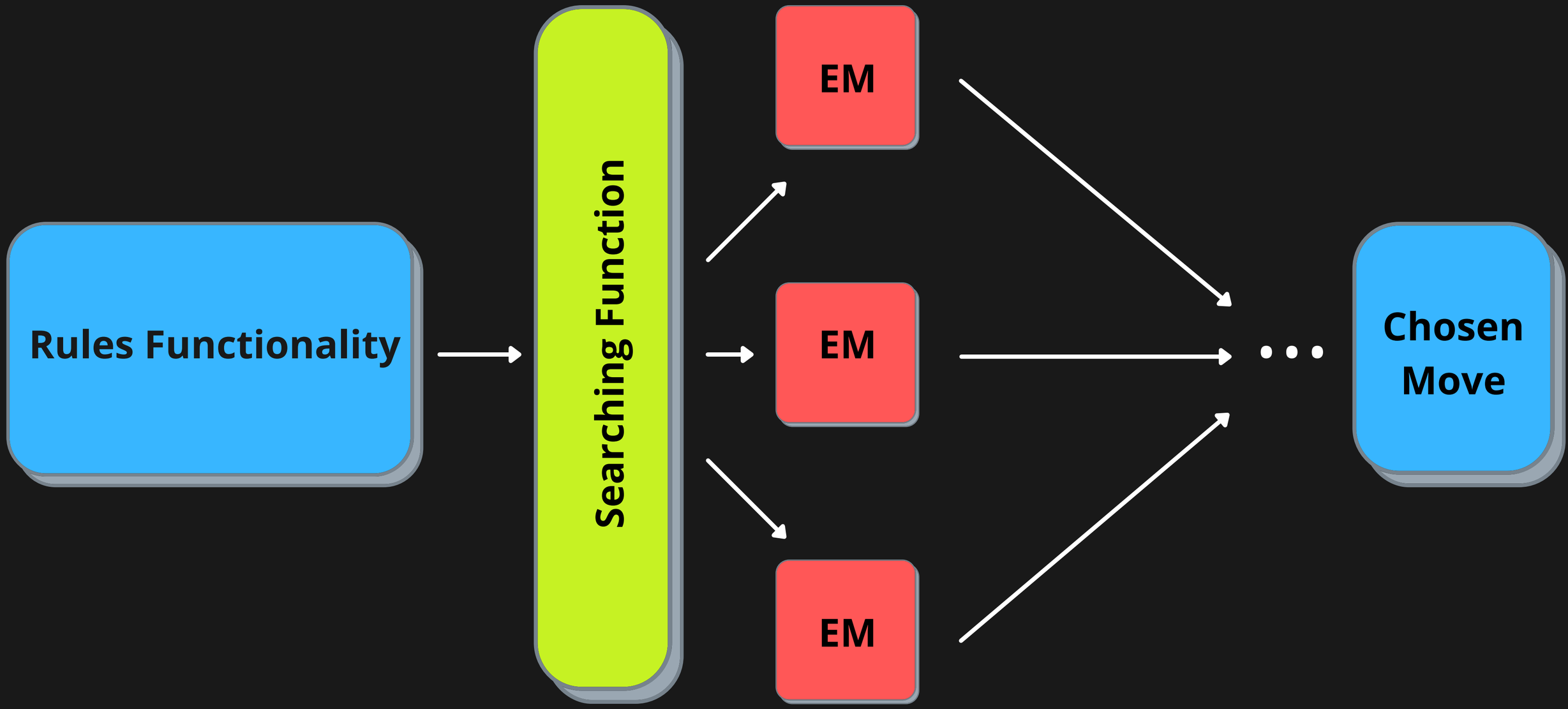
Nathaniel Martin

Agenda

- Engine Architecture
- Data Preparation
- Model Construction
- Performance

Engine Architecture




Rules Functionality
Searching Function
EM
EM
EM
Chosen Move

Data Preparation

Lichess Puzzle Data

~ 2.2 million puzzles

All taken from actual games played.

"Only Move" Puzzles

This lets us label moves easily.

A position has ~ 32 possible moves

This makes the data very imbalanced

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.

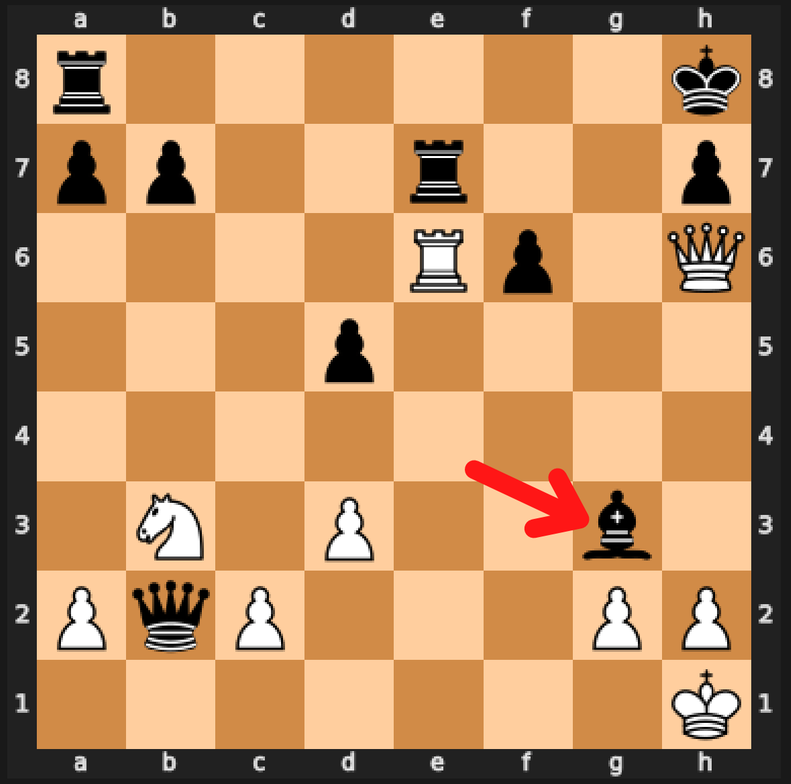
Puzzle Pieces

r6k/pp2r2p/4Rp1Q/3p4/8/1N1P2R1/PqP2bPP/7K

f2g3

UCI Moves

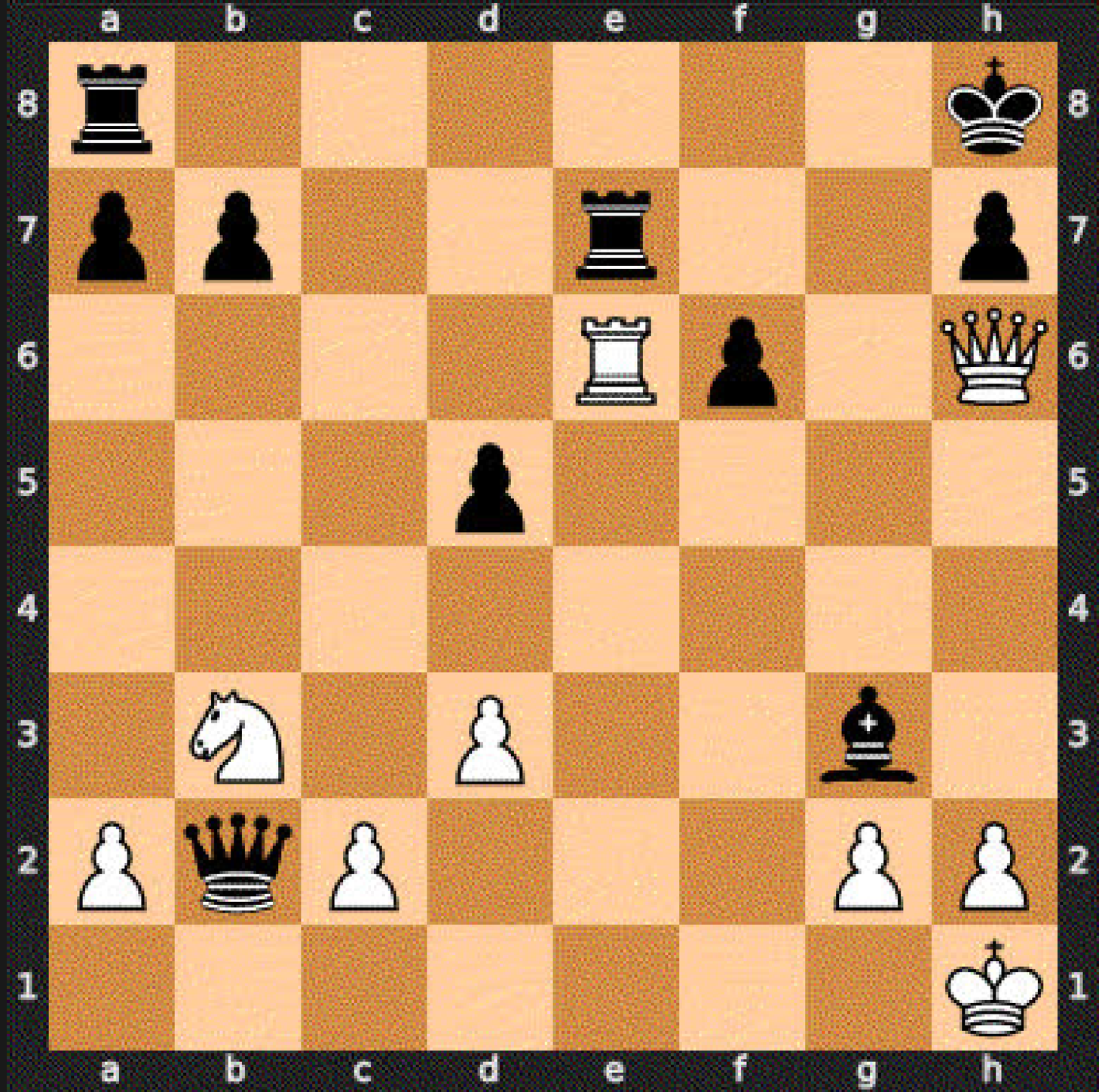
The first move is a prep move to get to the actual puzzle position. The second move is our target.



r6k/pp2r2p/4Rp1Q/3p4/8/1N1P2b1/PqP3PP/7K

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

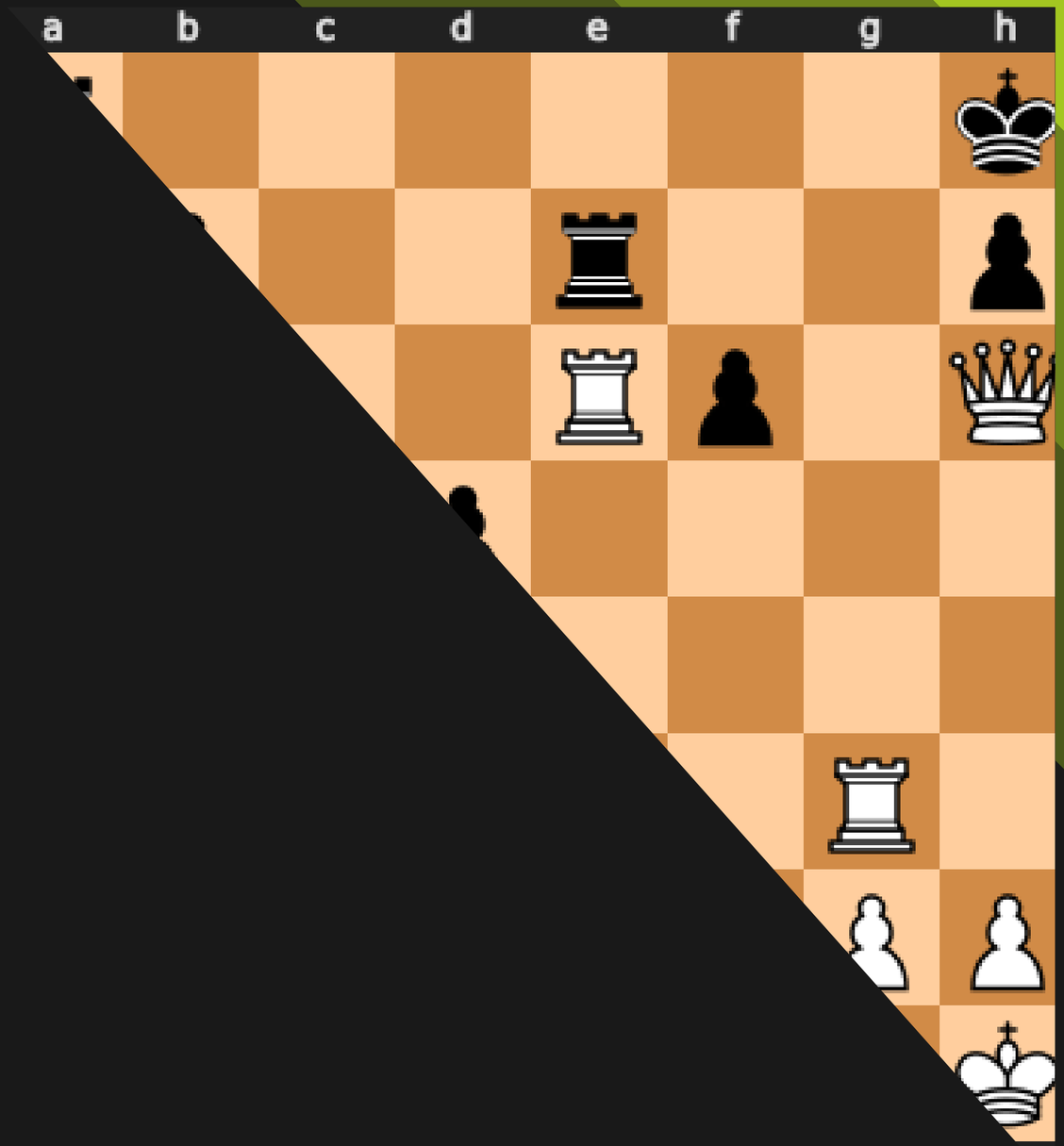
Relatively Small

Just under 55,000 parameters

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: ~88%

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

To see more...

Play the bot on Lichess:
https://lichess.org/@/PuzzledBot

Tensorboard:

Thank you!

Linkedin: www.linkedin.com/in/nathaniel-martin-73b037227
Github: https://github.com/UpGoerFive