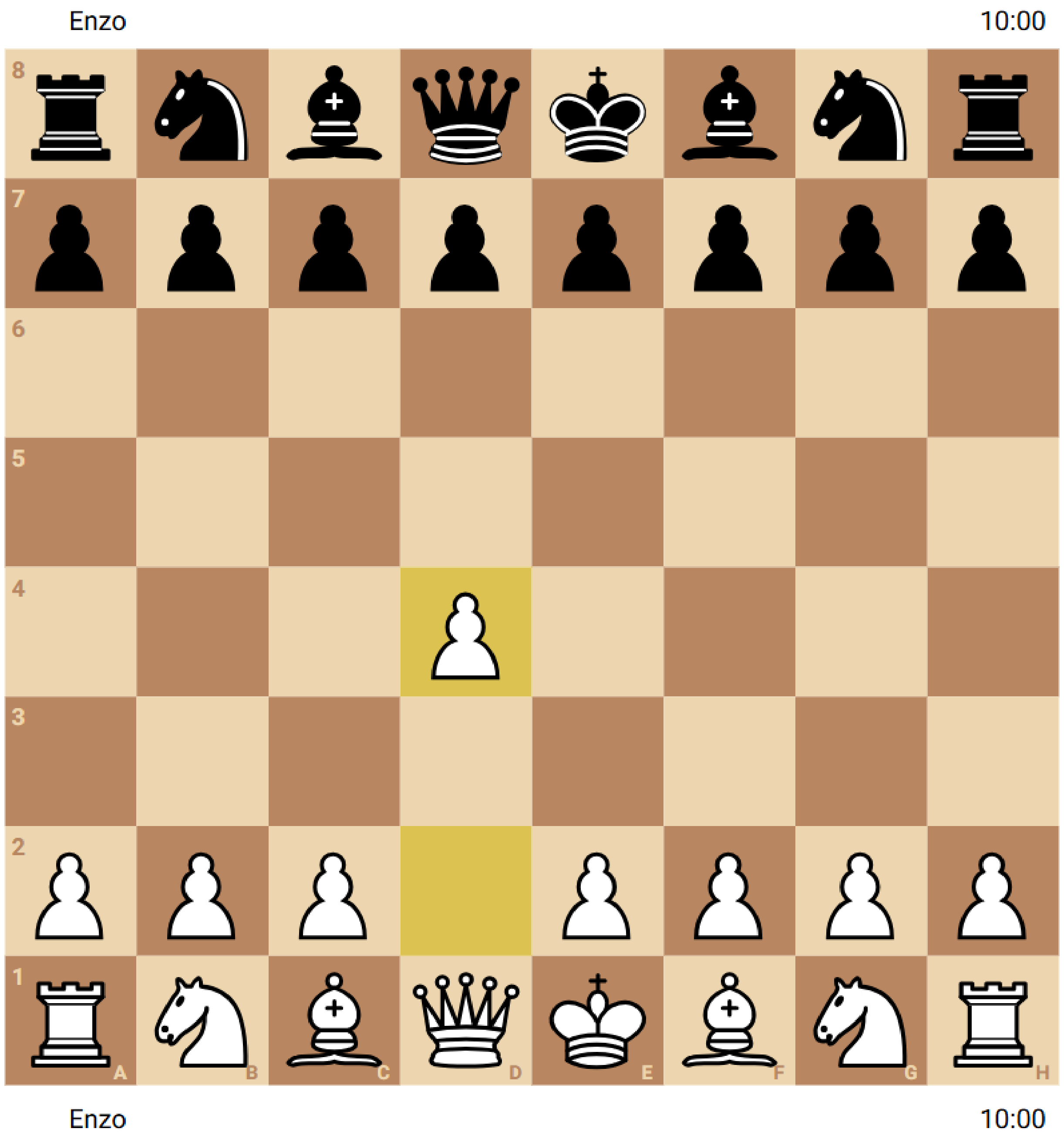
INSTITUT POLYTECHNIQUE DE PARIS

ÉCOLE POLYTECHNIQUE

Checkmate by learning: A modular Reinforcement Learning approach to chess

Enzo Pinchon, Yann Baglin-Bunod, Luis Wiedmann,
Shashwat Sharma, Mathieu Antonopoulos

Environment



Frontend/Backend Architecture

Backend

- Game Logic Engine: Core chess implementation in backend/src/chess handling rules, moves and state management
- Ai model architecture: modular design where agents can inherit from the base engine class
  - We have diverse AI Implementations using techniques from simple heuristics to advanced algorithms
- Communication Layer: socket-based server to interact with frontend

Engine & DeepEngine

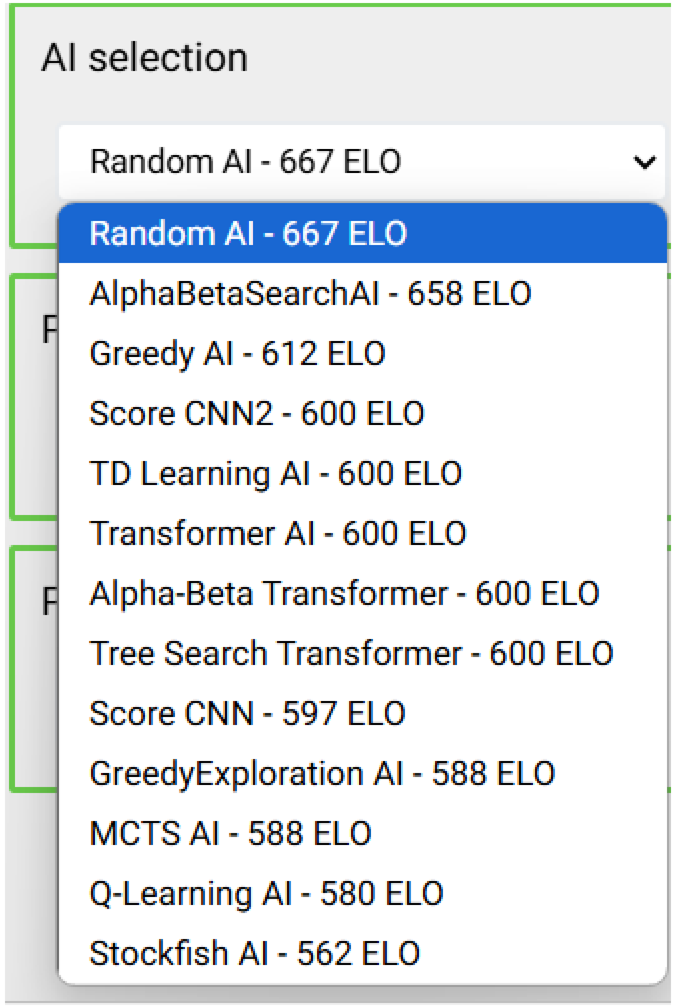
Inheritance of engines – Modular framework to implement engine and deep engines.

```python
with model | generative_head | with_prints | auto_save as env:
    plot_data = env.train(
        epochs=epochs,
        batch_size=batch_size,
        loader=ld_games | ld_puzzles
    )

    env.test(loader=ld_games | ld_puzzles)

    env.plot(plot_data)
```

Frontend Interface



- Frontend allows interactive play and AI benchmarking.
- Supports human vs. AI, AI vs. AI, and human vs. human.
- Visual representation of moves, evaluations, and game states.

Models

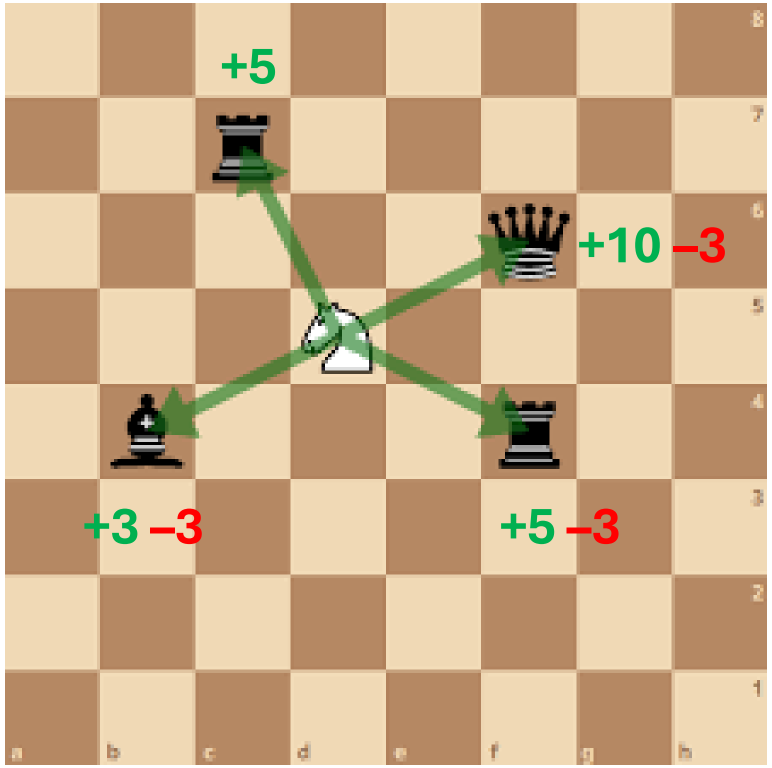
Model Overview

- **Greedy AI**
- **MCTS AI** (Monte Carlo Tree Search)
- **Random AI**
- **Score CNN**
- **Stockfish AI** (Baseline engine for comparison)
- **Transformer AI**
- **AlphaBetaSearch**

Greedy / Greedy Exploration

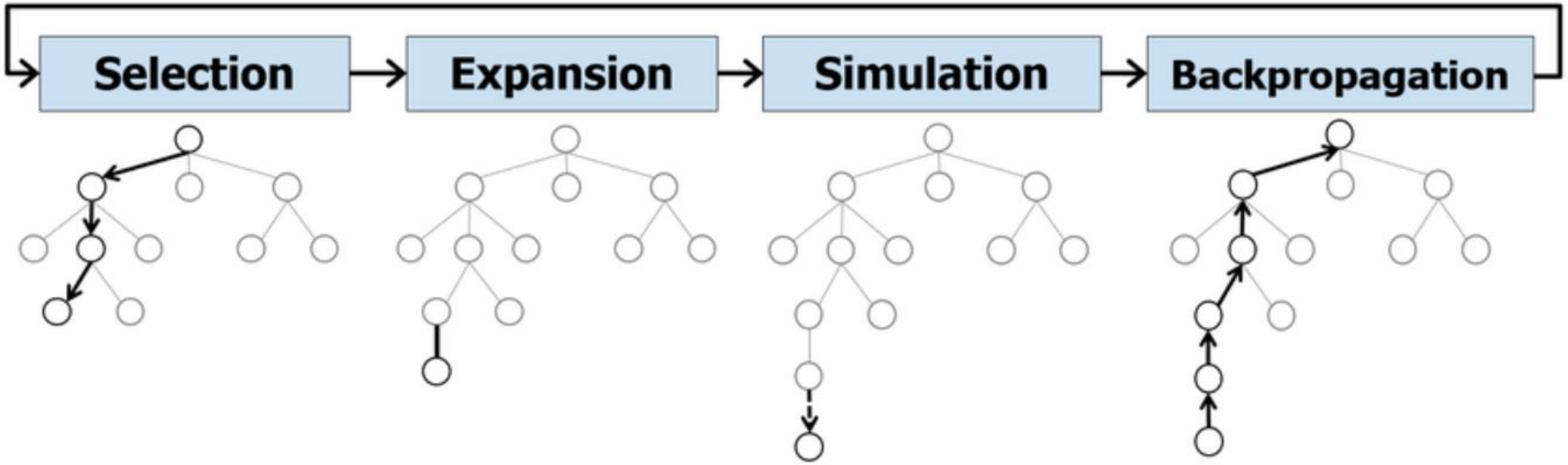
- Optimized Greedy AI that plays as strongly as possible with a single-move evaluation.
- It uses move selection based on piece values (MVV-LVA principle) and positional advantages.

- Strengths:
  - Fast decision-making (no deep search).
  - Simple evaluation based on material and position.
- Limitations:
  - Fails in long-term strategy.
  - Vulnerable to tactical traps.



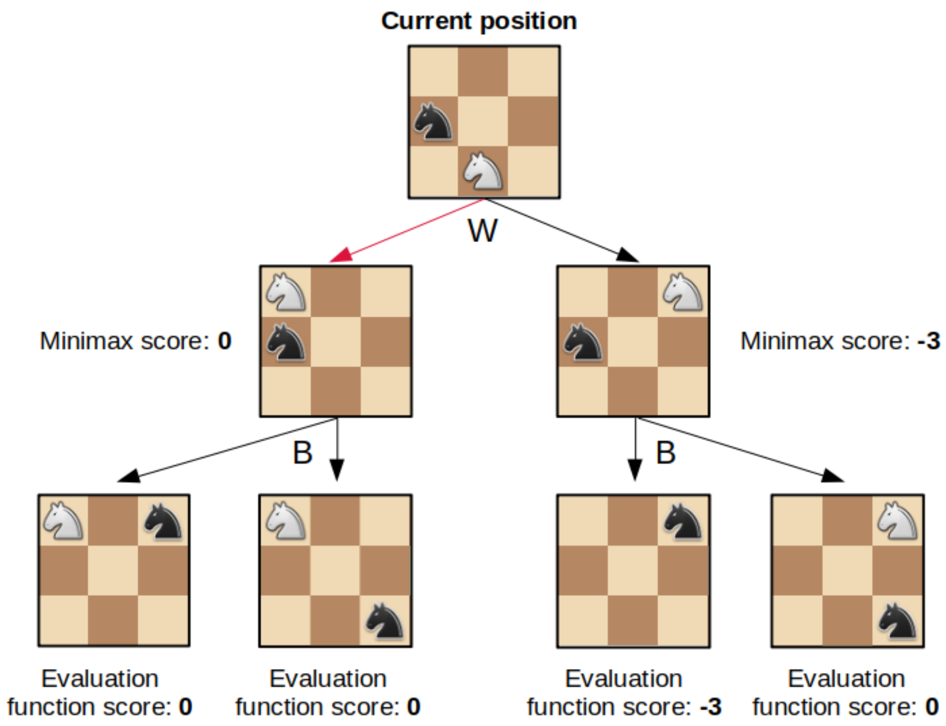
MCTS

- Implemented some heuristics and optimizations that guide the search more effectively.
- It uses move selection based on piece values (MVV-LVA principle) and positional advantages .
- Improved evaluation function, weighted random selection explore/exploit.



Alpha Beta search

- Implemented Alpha-Beta pruning with iterative deepening, inspired by SunFish and Stockfish engines.
- Enhanced pruning efficiency using piece-square tables, move ordering, and time-limited search.
- Evaluation based on material balance, piece mobility, and king safety, without neural networks.



Deep Networks - Playing

1) ChessEmbedding: converts board positions into a high-dimensional latent space.
2) GenerativeHead: generates board reconstructions.
3) BoardEvaluator: outputs a probability distribution for game outcomes.

Implemented with both a CNN and a Transformer

Chess Embedding	One-hot to Latent space
GenerativeHead	Ranking list of best moves
BoardEvaluator:	White: 78% Black:22%

Score CNN

- CNN-based evaluation:
  - Captures spatial patterns in board positions.
  - Predicts win probability from a given state.

- Uses convolutional layers with CBAM & SE attention to extract chessboard patterns and enhance feature importance.
  - Incorporates heatmaps to highlight critical board areas for evaluation and move generation.
  - Less effective at long-term planning than transformers but faster and more efficient for local position analysis

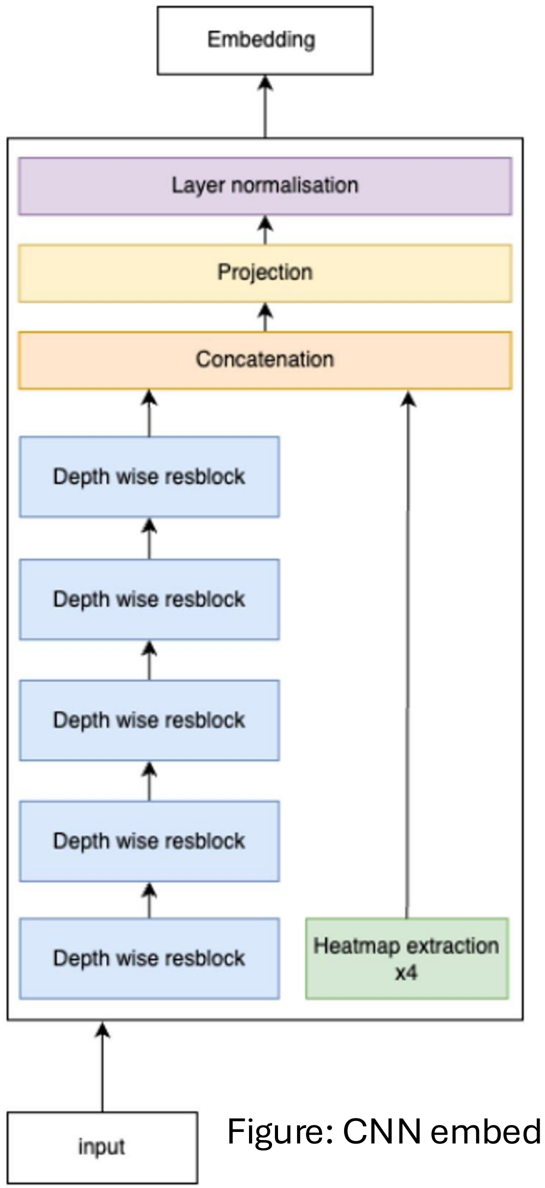
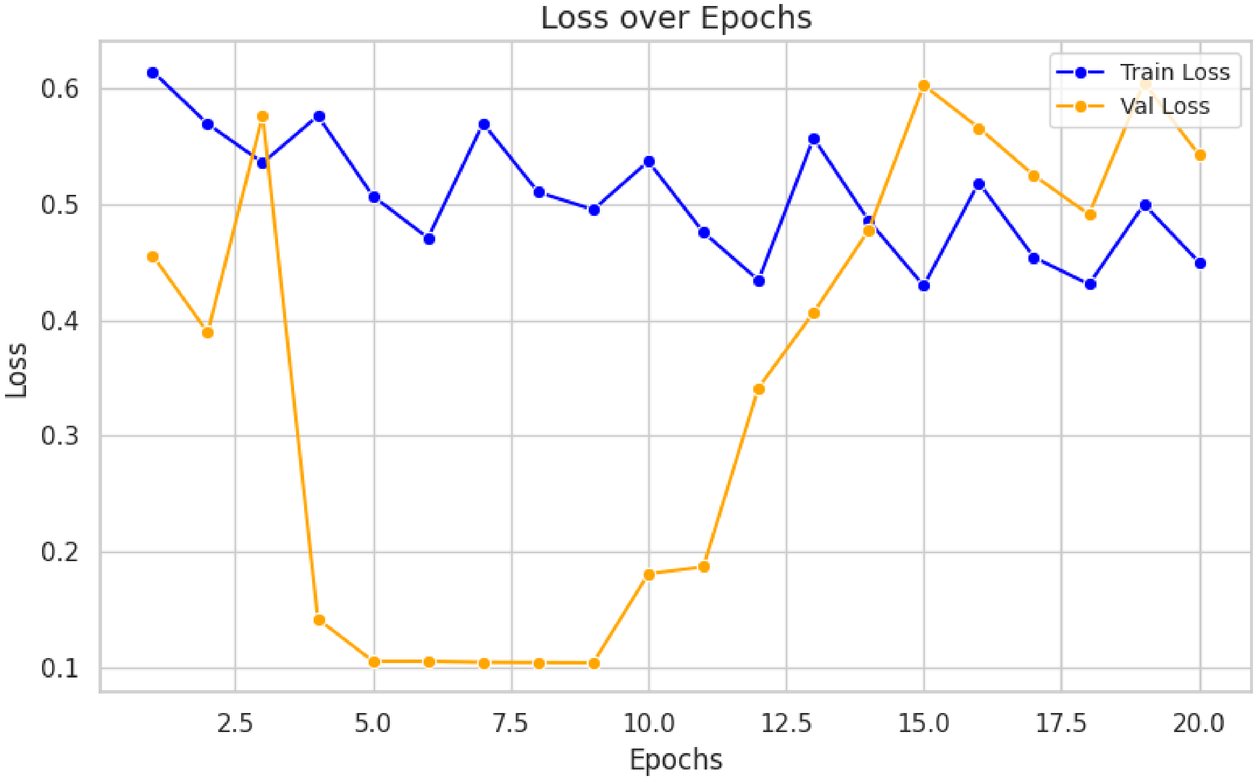
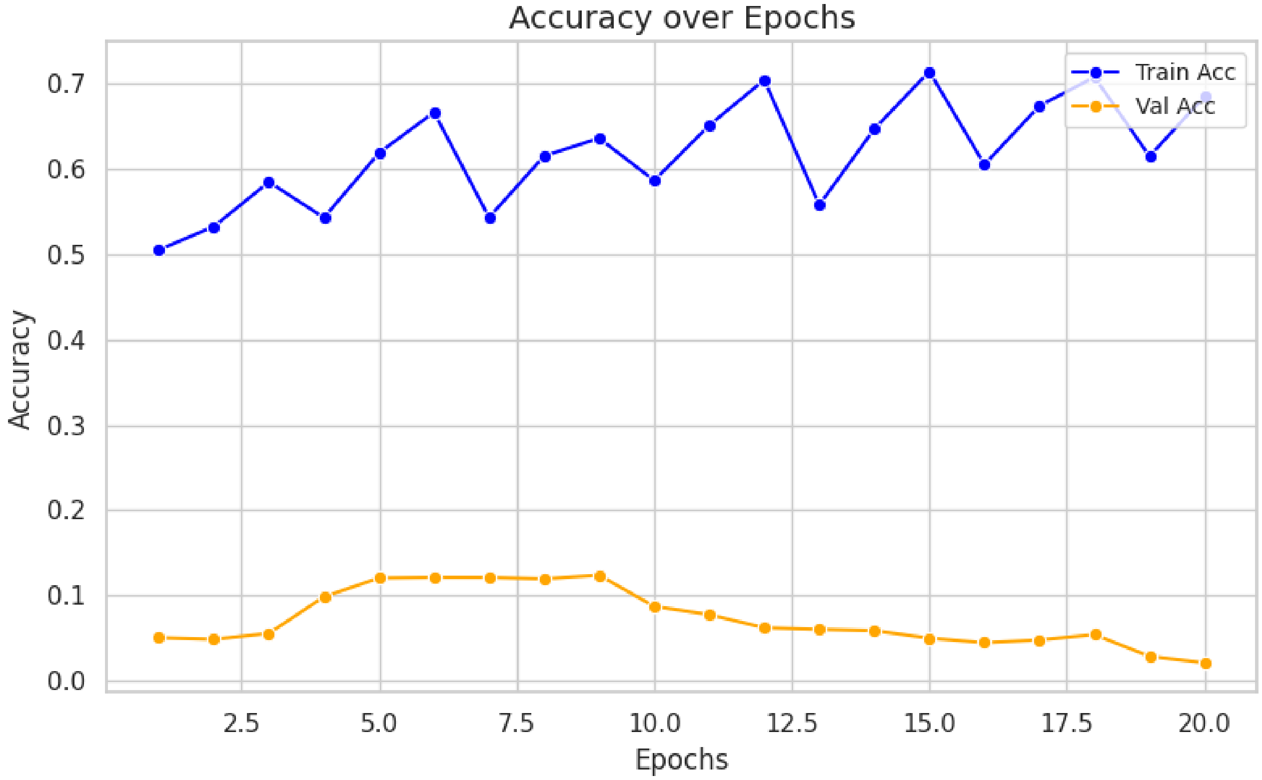


Figure: CNN embedding architecture

CNN training without attention

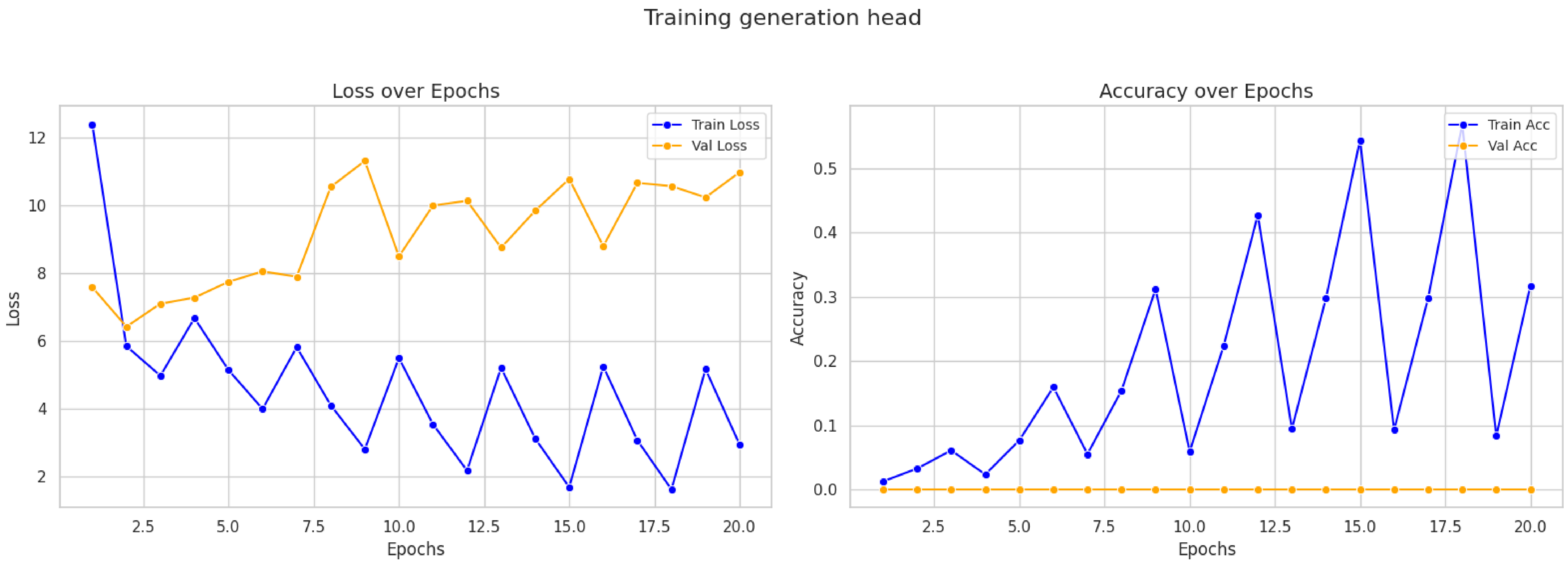
Training board evaluation head



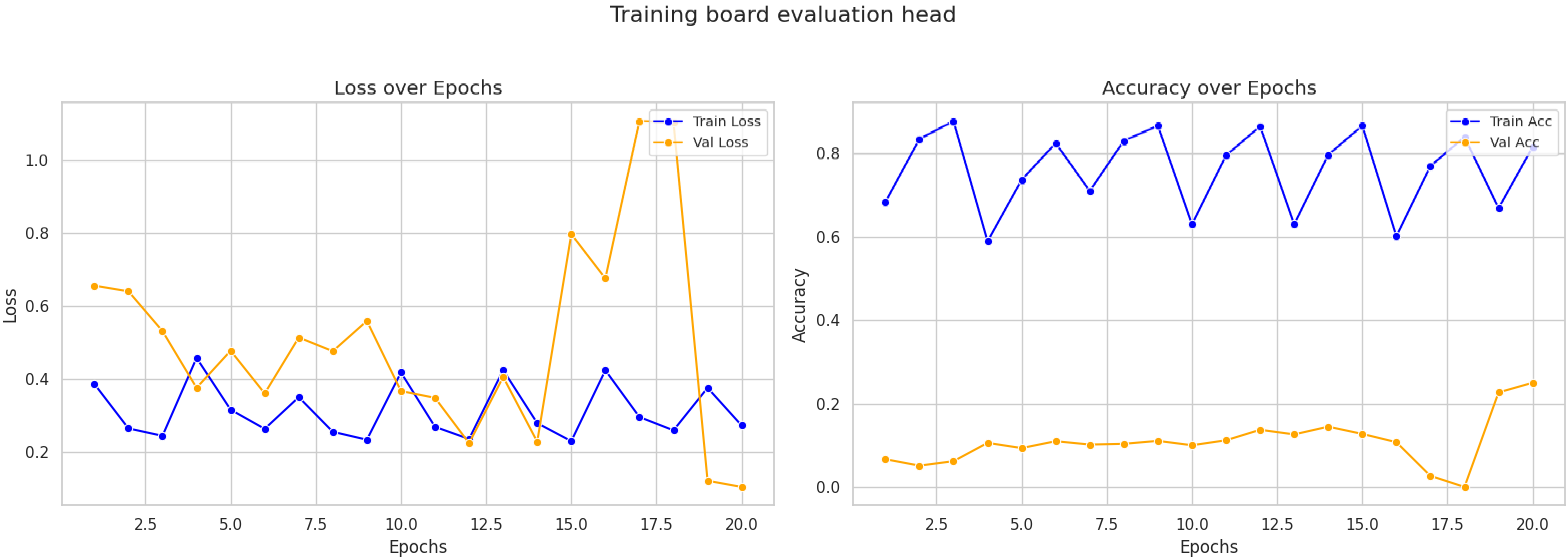


CNN training without attention

CNN training with CBAM attention



CNN training with CBAM attention



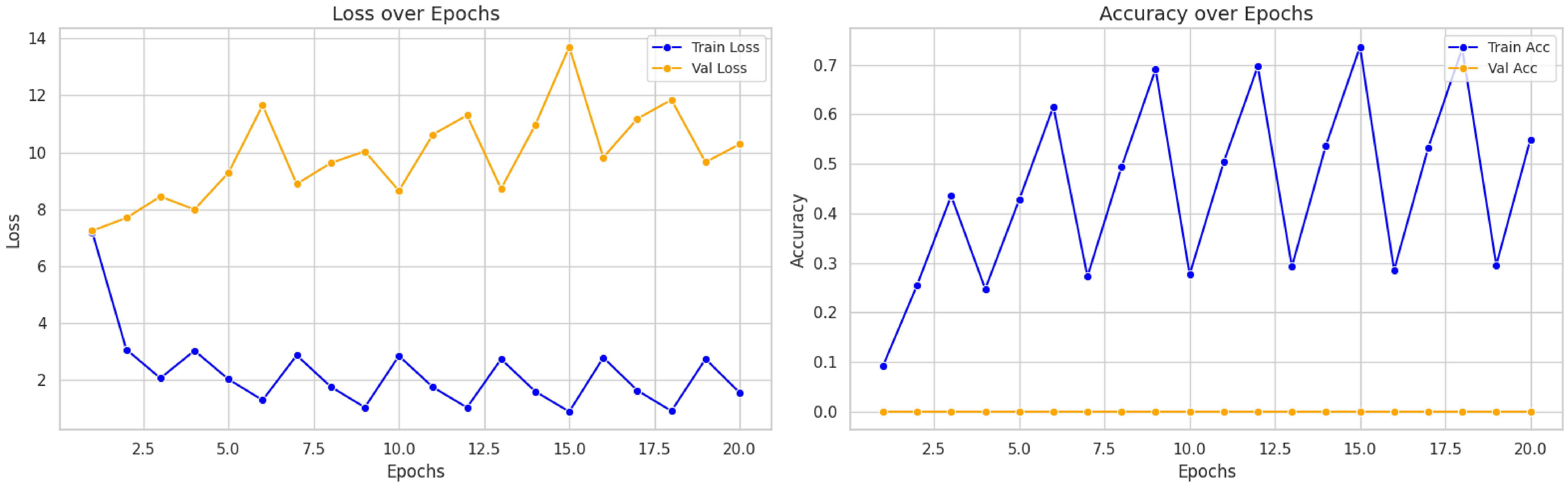
Transformer AI

- Transformer-based evaluation:
  - Uses self-attention to evaluate chess positions.
  - Captures long-range dependencies between pieces.
  - Requires large training data & high computation.

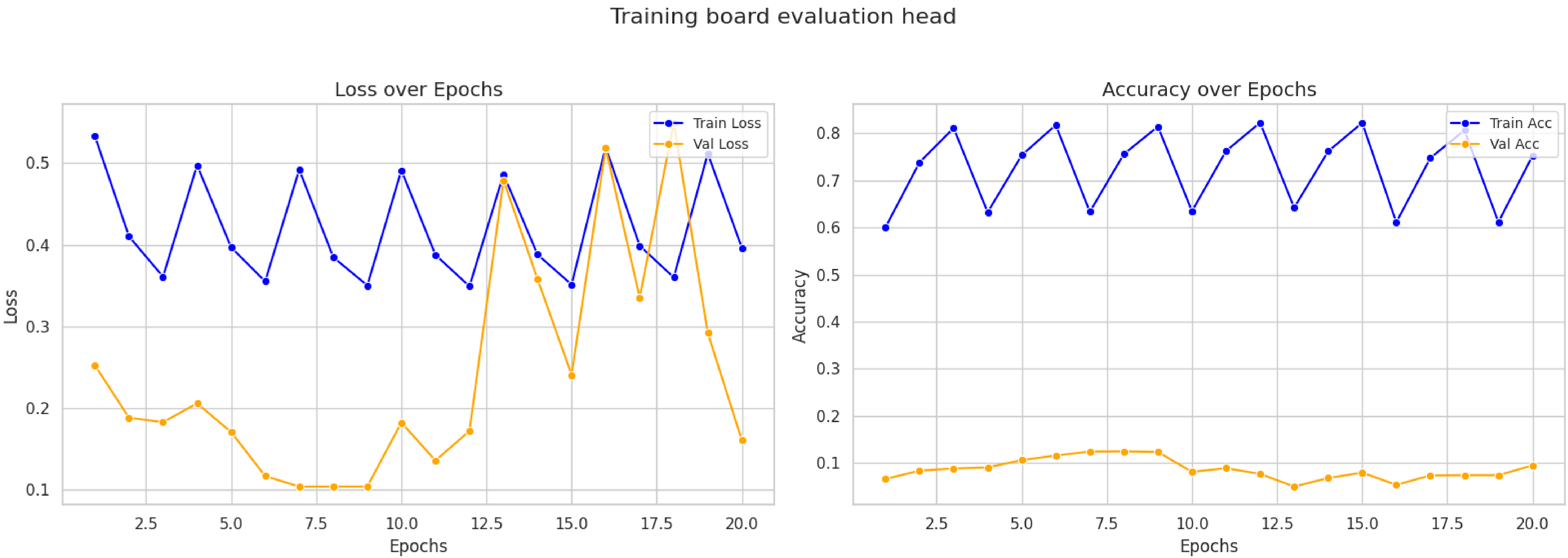
- Architecture: Processes 8×8×13 board state through a Chess Transformer Encoder (piece + positional embeddings, CNN for local features, transformer blocks for global context).

Transformer training

Training generation head

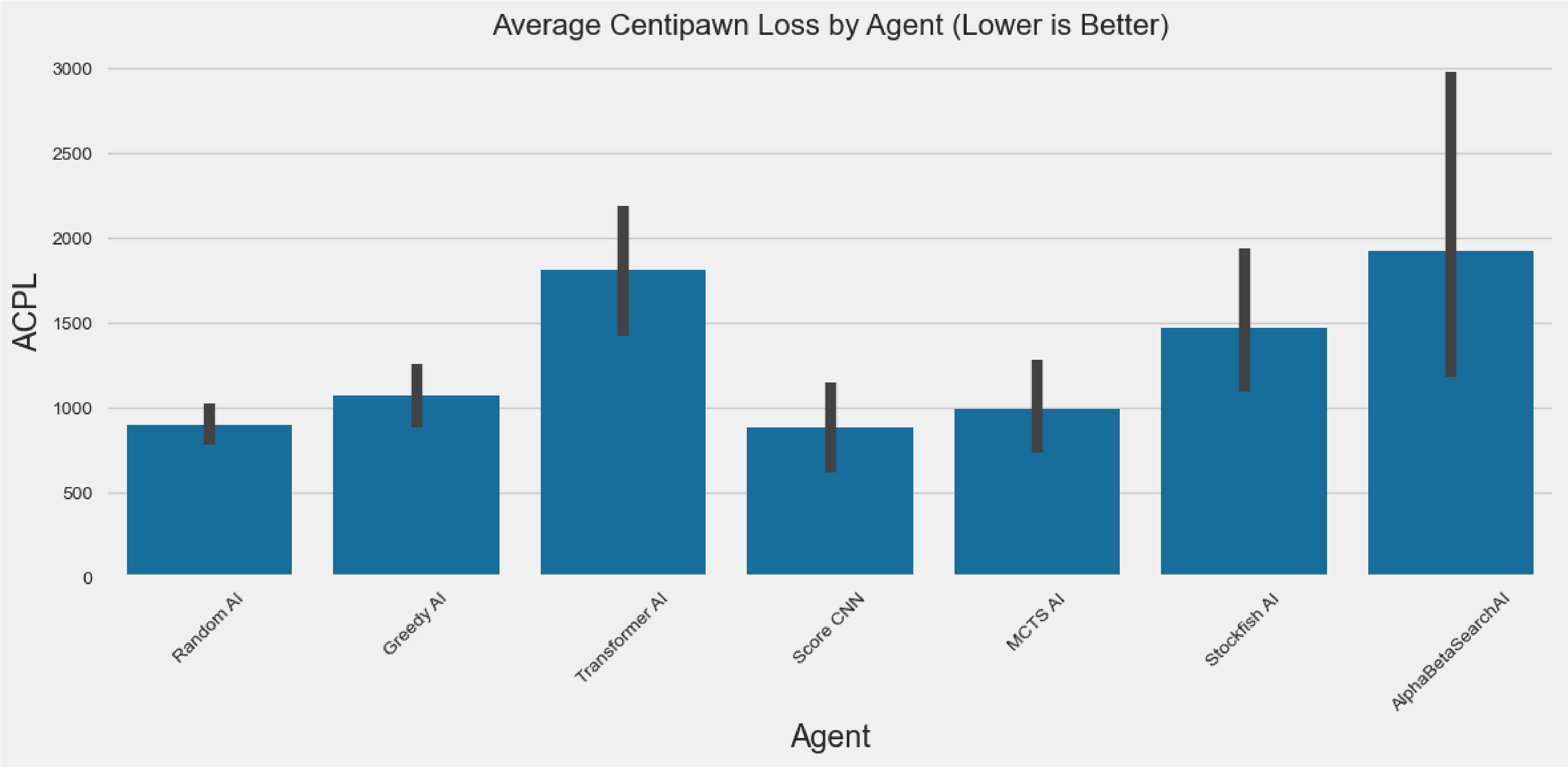


Transformer training



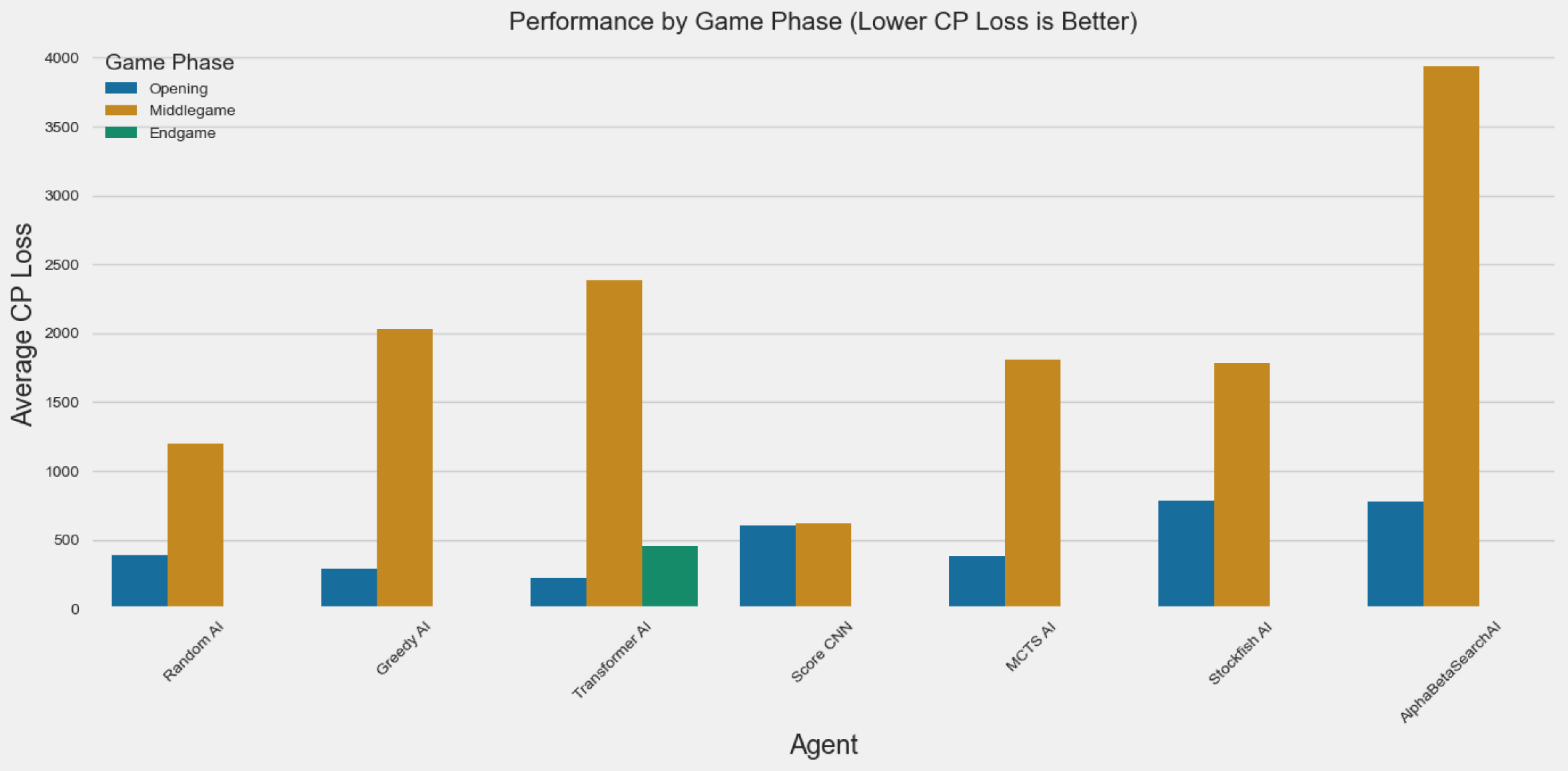
Evaluation

Evaluation



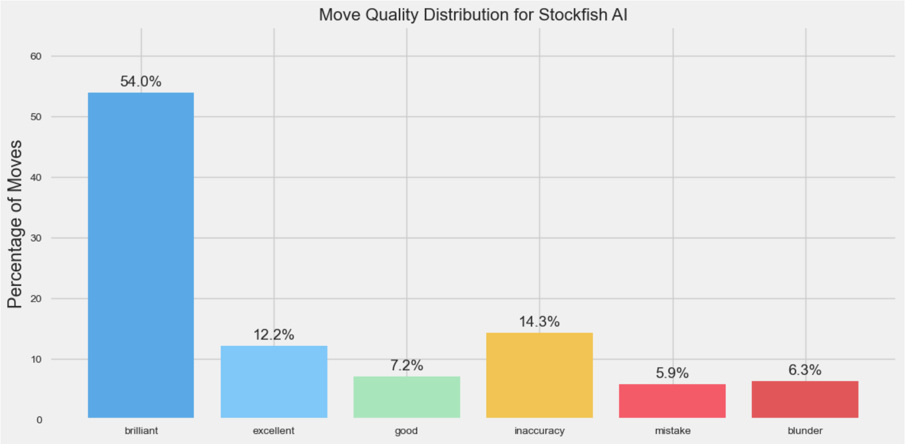
Average Centipawn Loss by Agent (Lower is Better)
ACPL
3000
2500
2000
1500
1000
500
0
Random AI
Greedy AI
Transformer AI
Score CNN
MCTS AI
Stockfish AI
AlphaBetaSearchAI
Agent

Evaluation

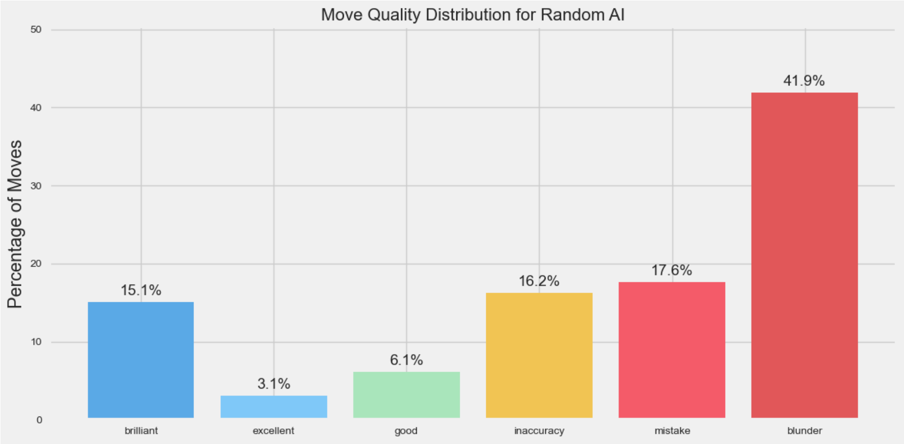


Performance by Game Phase (Lower CP Loss is Better)
Game Phase
Opening
Middlegame
Endgame
Average CP Loss
4000
3500
3000
2500
2000
1500
1000
500
0
Random AI
Greedy AI
Transformer AI
Score CNN
MCTS AI
Stockfish AI
AlphaBetaSearchAI
Agent

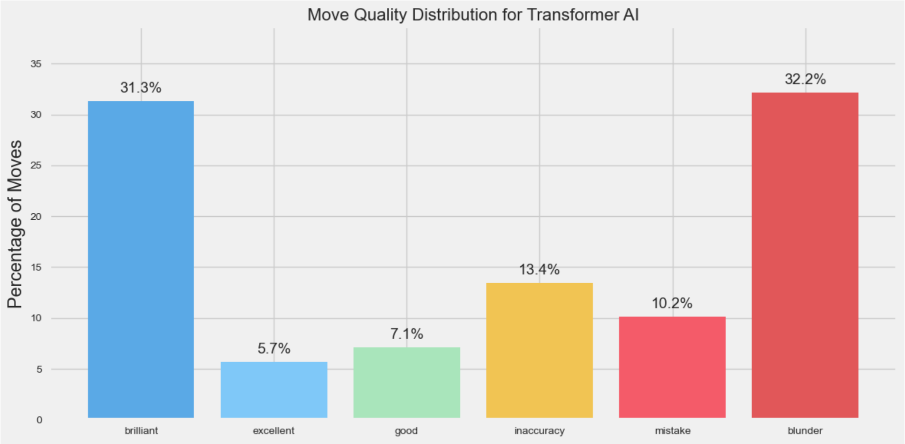
Evaluation



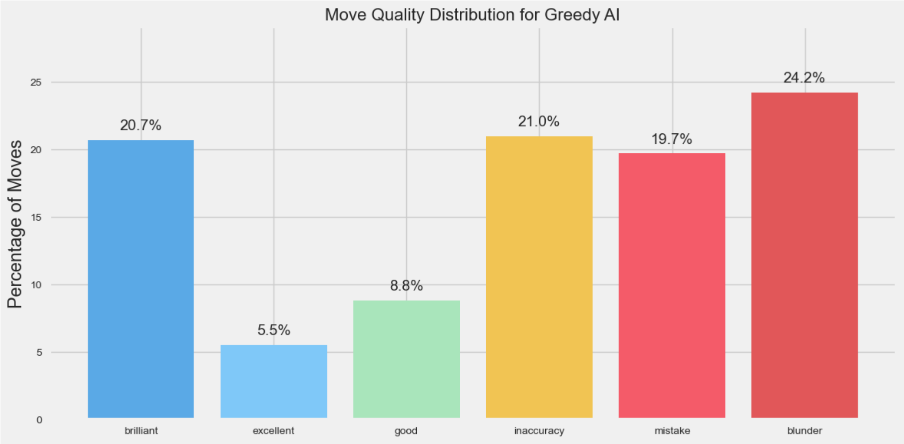
Move Quality Distribution for Stockfish AI
Percentage of Moves
54.0%
12.2%
7.2%
14.3%
5.9%
6.3%
brilliant
excellent
good
inaccuracy
mistake
blunder



Move Quality Distribution for Random AI
Percentage of Moves
15.1%
3.1%
6.1%
16.2%
17.6%
41.9%
brilliant
excellent
good
inaccuracy
mistake
blunder

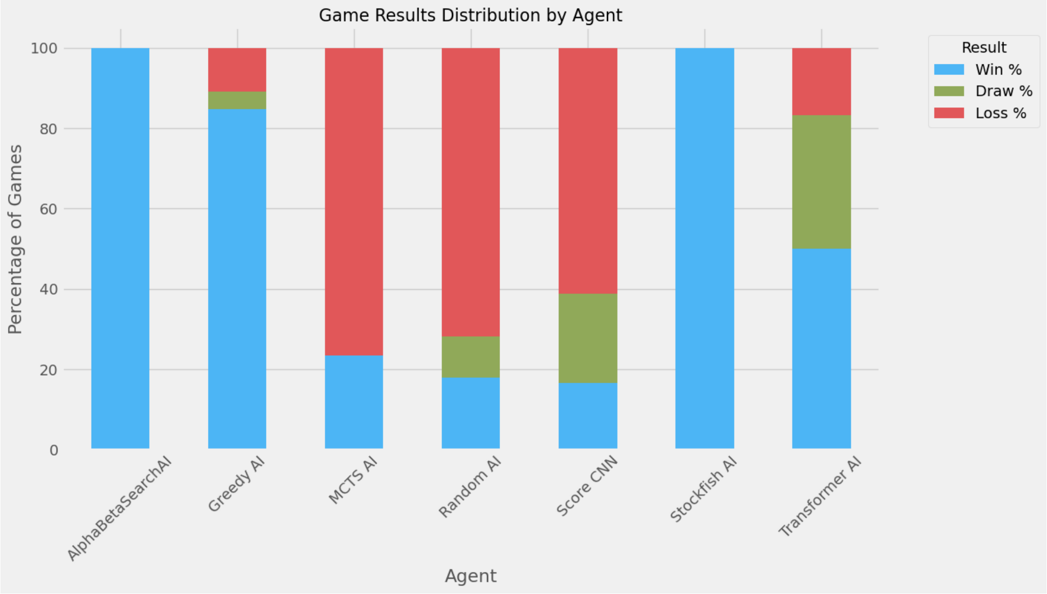


Move Quality Distribution for Transformer AI
Percentage of Moves
31.3%
5.7%
7.1%
13.4%
10.2%
32.2%
brilliant
excellent
good
inaccuracy
mistake
blunder

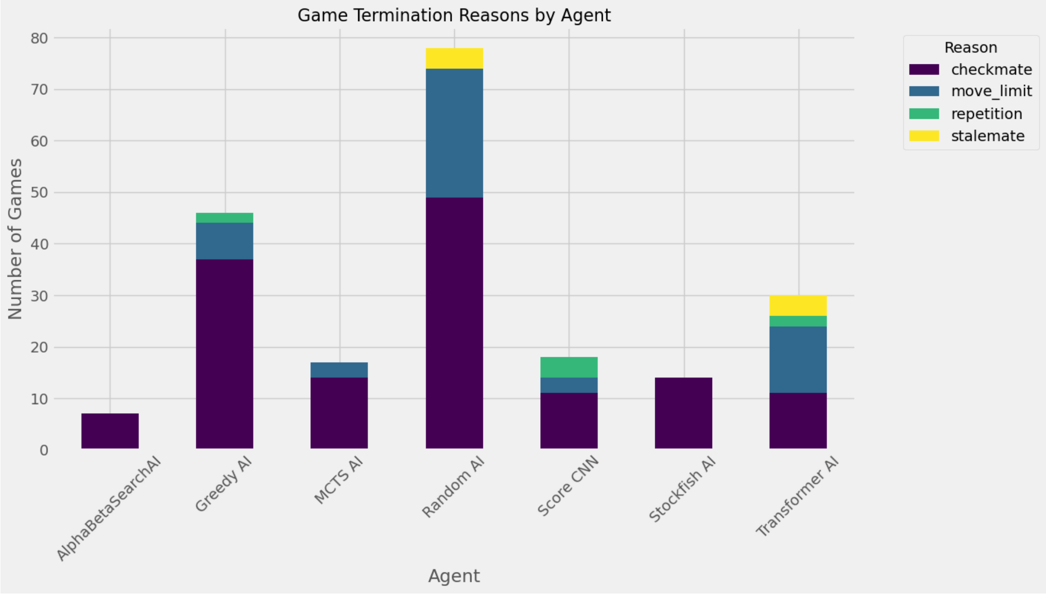


Move Quality Distribution for Greedy AI
Percentage of Moves
20.7%
5.5%
8.8%
21.0%
19.7%
24.2%
brilliant
excellent
good
inaccuracy
mistake
blunder

Evaluation

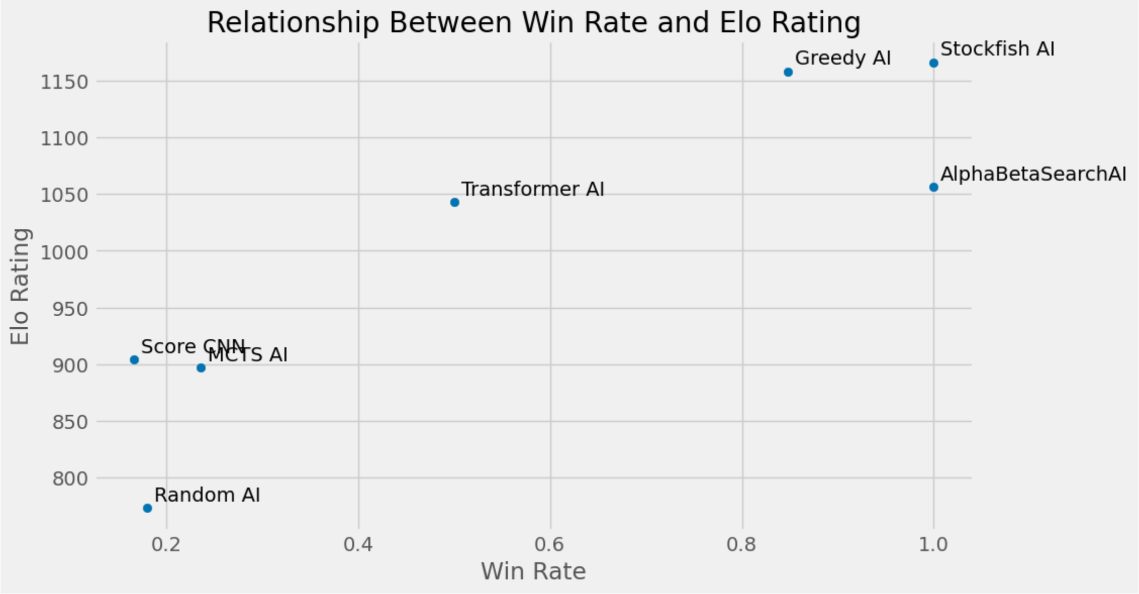


Game Results Distribution by Agent
Percentage of Games
0
20
40
60
80
100
AlphaBetaSearchAI
Greedy AI
MCTS AI
Random AI
Score CNN
Stockfish AI
Transformer AI
Agent
Result
Win %
Draw %
Loss %

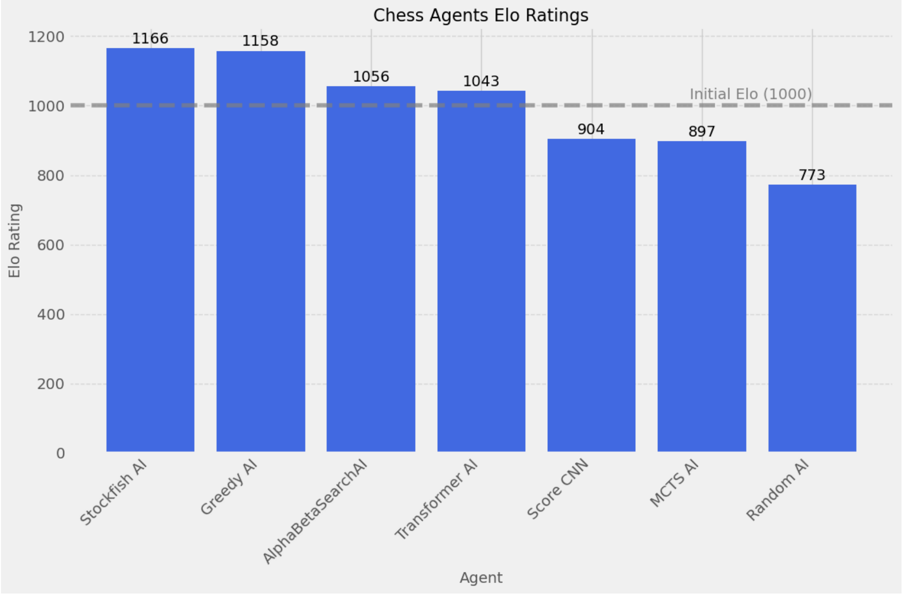


Game Termination Reasons by Agent
Number of Games
0
10
20
30
40
50
60
70
80
AlphaBetaSearchAI
Greedy AI
MCTS AI
Random AI
Score CNN
Stockfish AI
Transformer AI
Agent
Reason
checkmate
move_limit
repetition
stalemate

Evaluation

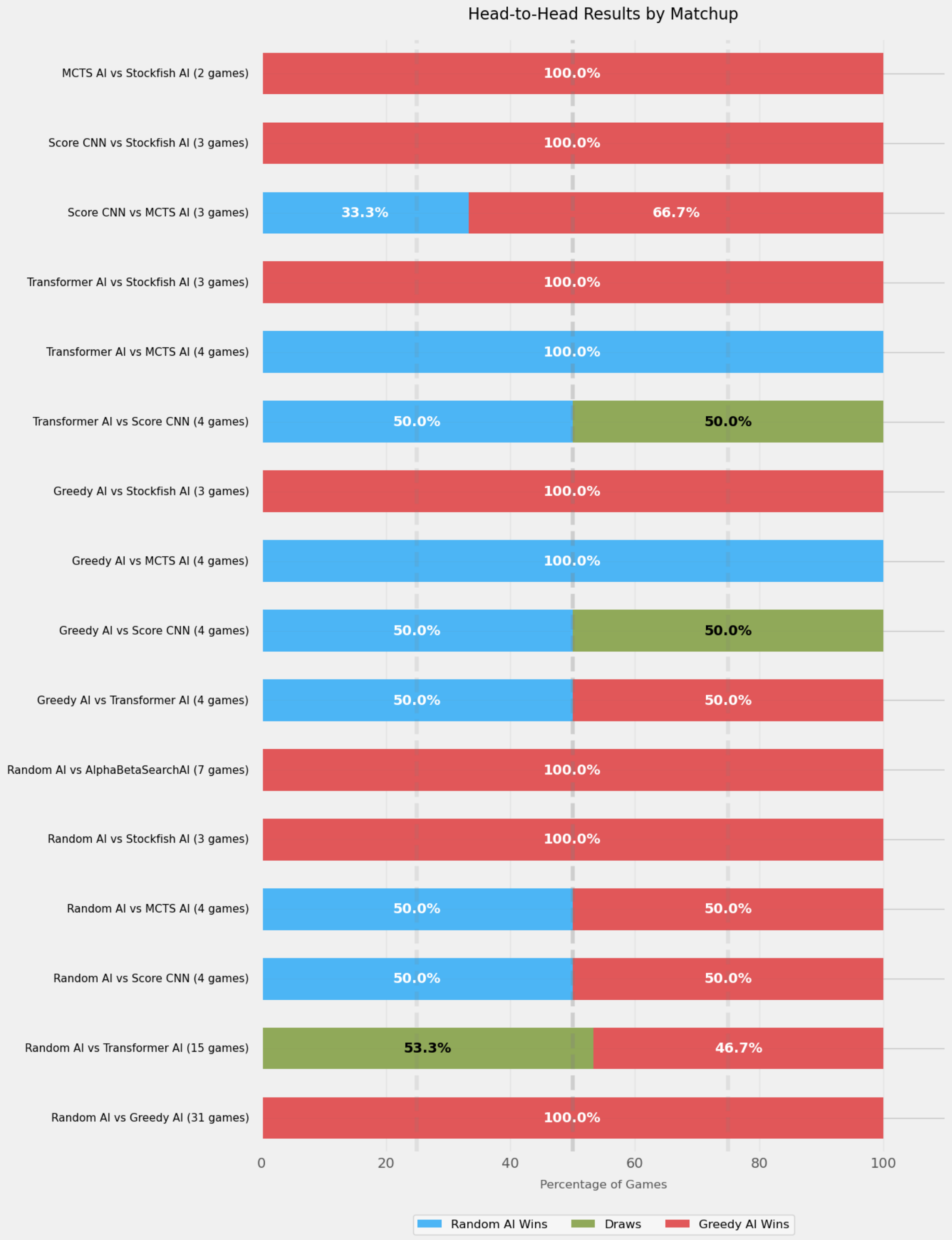


Relationship Between Win Rate and Elo Rating
Elo Rating
Win Rate
1150
1100
1050
1000
950
900
850
800
0.2
0.4
0.6
0.8
1.0
Greedy AI
Stockfish AI
Transformer AI
AlphaBetaSearchAI
Score CNN
MCTS AI
Random AI



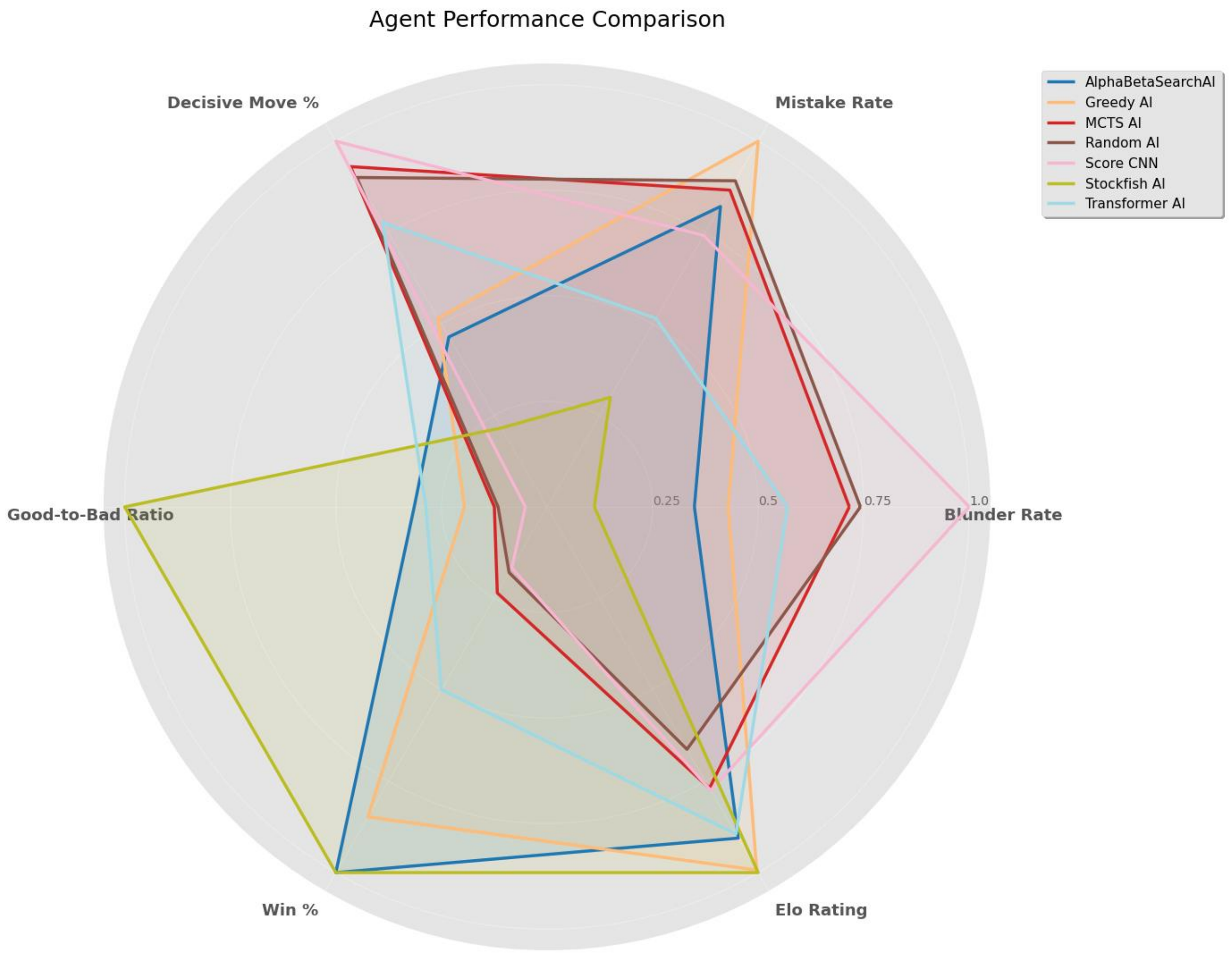
Chess Agents Elo Ratings
Elo Rating
Agent
1200
1000
800
600
400
200
0
1166
1158
1056
1043
904
897
773
Initial Elo (1000)
Stockfish AI
Greedy AI
AlphaBetaSearchAI
Transformer AI
Score CNN
MCTS AI
Random AI

Evaluation



Head-to-Head Results by Matchup
MCTS AI vs Stockfish AI (2 games)
100.0%
Score CNN vs Stockfish AI (3 games)
100.0%
Score CNN vs MCTS AI (3 games)
33.3%
66.7%
Transformer AI vs Stockfish AI (3 games)
100.0%
Transformer AI vs MCTS AI (4 games)
100.0%
Transformer AI vs Score CNN (4 games)
50.0%
50.0%
Greedy AI vs Stockfish AI (3 games)
100.0%
Greedy AI vs MCTS AI (4 games)
100.0%
Greedy AI vs Score CNN (4 games)
50.0%
50.0%
Greedy AI vs Transformer AI (4 games)
50.0%
50.0%
Random AI vs AlphaBetaSearchAI (7 games)
100.0%
Random AI vs Stockfish AI (3 games)
100.0%
Random AI vs MCTS AI (4 games)
50.0%
50.0%
Random AI vs Score CNN (4 games)
50.0%
50.0%
Random AI vs Transformer AI (15 games)
53.3%
46.7%
Random AI vs Greedy AI (31 games)
100.0%
0
20
40
60
80
100
Percentage of Games
Random AI Wins
Draws
Greedy AI Wins

Evaluation



Agent Performance Comparison
Decisive Move %
Mistake Rate
Blunder Rate
Elo Rating
Win %
Good-to-Bad Ratio
0.25
0.5
0.75
1.0
AlphaBetaSearchAI
Greedy AI
MCTS AI
Random AI
Score CNN
Stockfish AI
Transformer AI

Conclusion

Modular Framework
Enabled rapid experimentation and direct comparison of diverse chess-playing methods.

Heuristic Efficiency
Stockfish and Greedy AI excelled due to efficient search strategies and handcrafted evaluations.

Transformer Strengths
Transformer models effectively leveraged attention mechanisms for global positional understanding.

CNN Limitations
CNN approaches faced generalization challenges related to spatial invariance and sequential dependencies.