Chess Rating Estimation from Moves and Clock Times

Abdelrahman Elazab, Michael Reda

The American University in Cairo, Department of Computer Science and Engineering

Introduction

Traditional chess rating systems like Elo and Glicko-2 rely on match outcomes rather than the quality or timing of individual moves. These systems often require many games to accurately reflect a player's true skill level, leading to potential mismatches—especially for new accounts, rapidly improving players, or those returning after inactivity.

Our Goal: Build a neural network that infers player skill directly from gameplay behavior by analyzing moves and clock times, providing accurate rating estimates on a move-by-move basis.

Applications:

- Early skill estimation for improved matchmaking
- Real-time training feedback for players
- Cheating detection through anomaly identification
- Dynamic rating updates within a single game

Dataset

Source: database.lichess.org
Size: 1.2 million games from April 2021 to July 2024
Coverage: Multiple time controls (UltraBullet, Bullet, Blitz, Rapid, Classical)

Data includes:
Player ratings (Glicko-2 system)
Complete move sequences
Clock times remaining after each move (%clk)
Game results and openings

Split: 80% training, 20% testing

Input Representation:
Board state: 12 binary planes (8×8) representing 6 piece types × 2 colors
Clock feature: Remaining seconds before each move (standardized)
Output: Continuous Elo rating predictions for both players after each move
Graph to include: Sample board representation showing the 12-plane encoding

Models

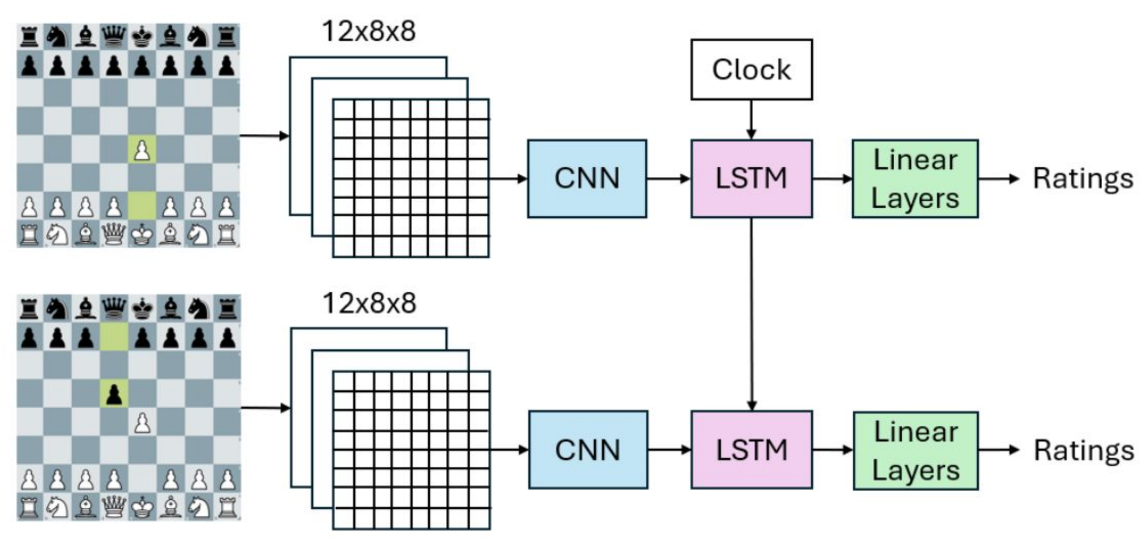
Baseline Architecture (RatingNet)
Original Model Components:

- 4-layer CNN for positional feature extraction
- Bidirectional LSTM for sequence modeling
- Fully connected layers for rating prediction

Key Features:

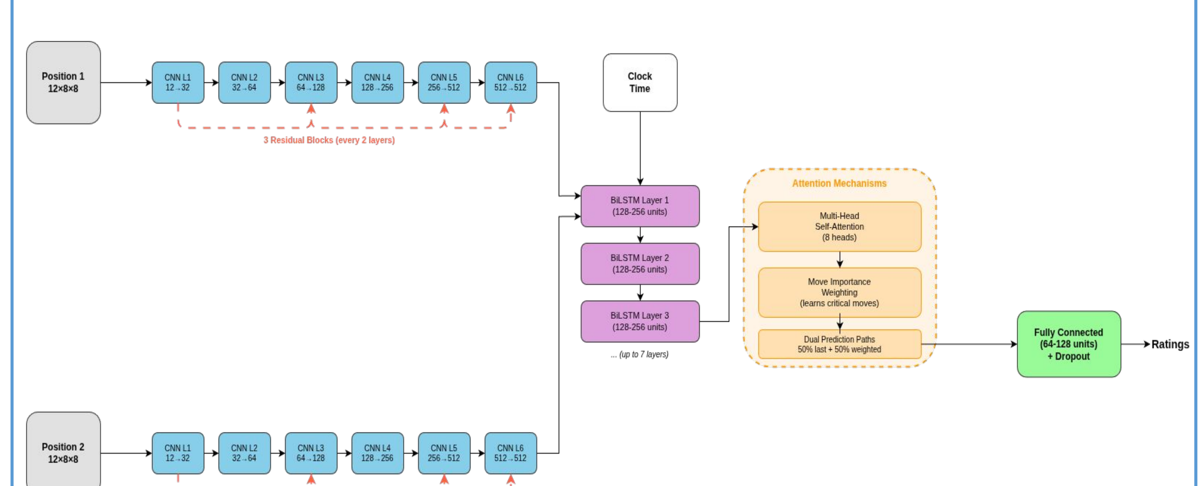
- Combines spatial (CNN) and temporal (LSTM) learning
- Incorporates clock time as an additional feature
- No hand-crafted features required

Baseline Performance:
Overall MAE: 303.9 Elo points
Best time control: Classical (209.8 Elo)
Worst time control: UltraBullet (408.6 Elo)



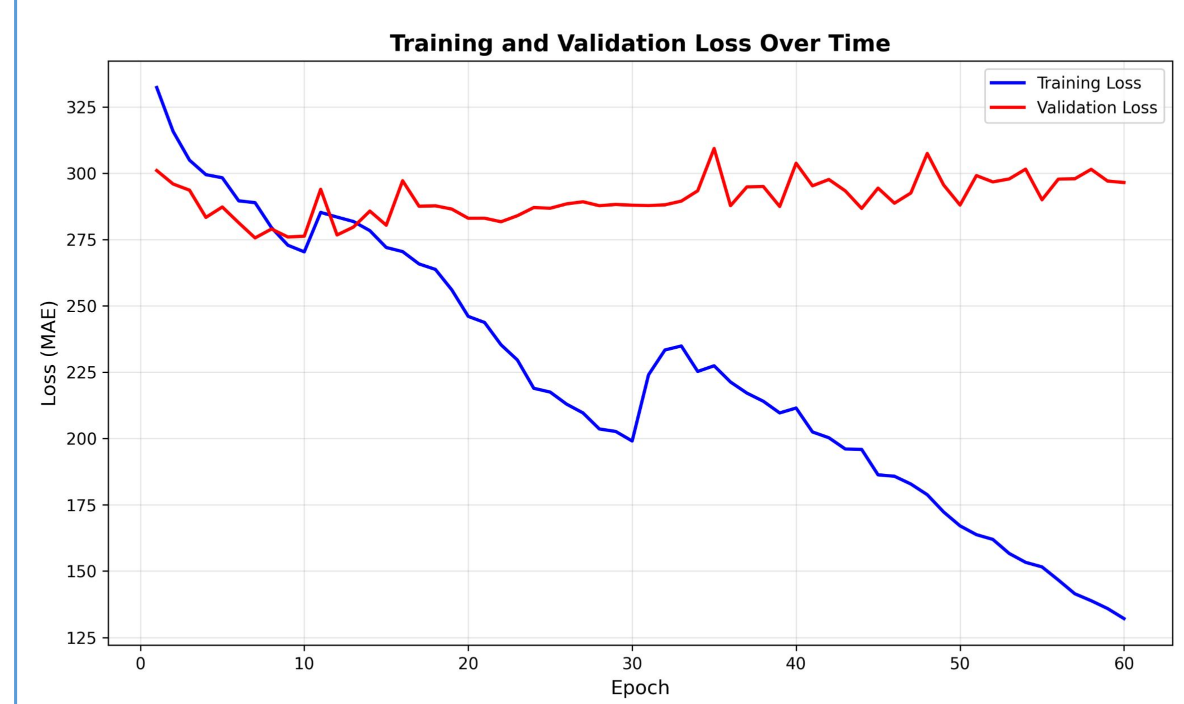
Our Proposed Improvements
Architecture Enhancements:
Deeper CNN: Increased from 4 to 6 convolutional layers (12→32→64→128→256→512→512 filters)
Residual Connections: Added 3 residual blocks with skip connections to prevent gradient degradation
Enhanced LSTM: Increased hidden size from 64 to 128-256 units
Larger FC Layers: Expanded from 16-32 to 64-128 hidden units
Attention Mechanisms:
Multi-Head Self-Attention (8 heads): Learns move-to-move relationships and long-range dependencies
Move Importance Weighting: Identifies which moves are most critical for rating prediction
Dual Prediction Paths: Combines last timestep (50%) with importance-weighted features (50%)
Training Enhancements:

- Data augmentation (horizontal flip, color swap with vertical flip)
- Cosine annealing learning rate scheduler
- Mixed precision training for efficiency
- Gradient clipping to prevent explosion
- Early stopping when no improvement
- Graph to include: New architecture diagram showing attention mechanisms and residual blocks



Results

The learning curves demonstrate effective model convergence with the attention-based architecture. Training loss (blue) decreases steadily from ~340 to ~170 MAE over 10 epochs, indicating the model successfully learns rating patterns from game data. Validation loss (red) follows a similar downward trend, starting at ~330 and stabilizing around ~280 MAE after epoch 8.



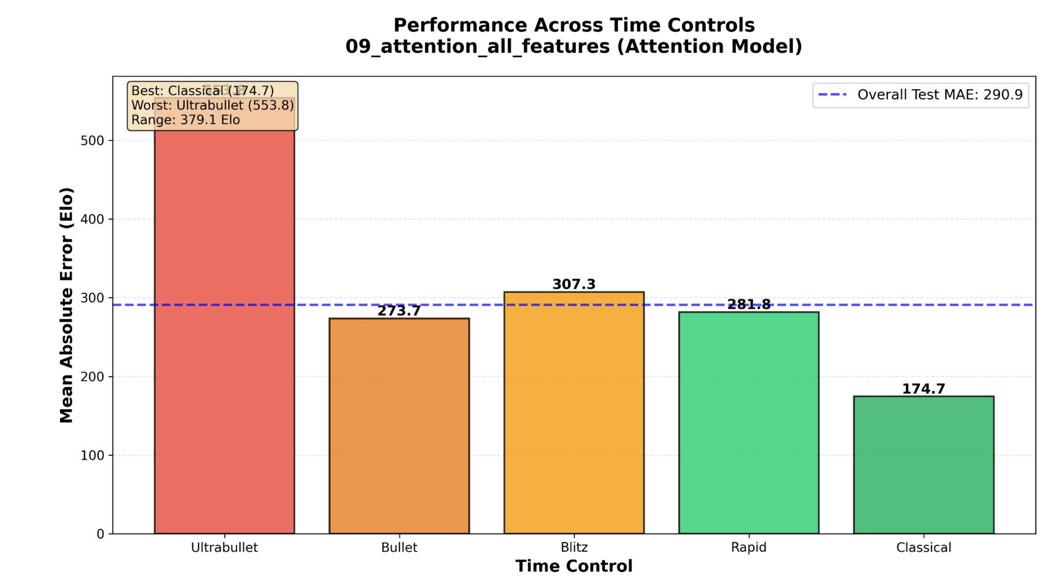
The table reveals a critical insight: architectural depth alone degrades performance. Simply adding convolutional layers (315.1 MAE), residual blocks (320.7 MAE), data augmentation (305.0 MAE), or cosine scheduling (327.0 MAE) all increased error compared to baseline (303.9 MAE). Even combining all ResNet-style improvements yielded 321.5 MAE—worse than baseline.

The Breakthrough: Introducing attention mechanisms reversed this trend:

- Basic multi-head attention: 302.2 MAE (-1.7 improvement)
- Adding move importance weighting: 290.7 MAE (-13.2 improvement)
- Full architecture with all enhancements: 290.9 MAE (-13.0 improvement)

Model	Test MAE	Improvement (vs. 308.6)
Baseline (Exp 01)	308.6	-
Improved No Residual (02)	303.8	-4.7 ✅
With Residual (03)	324.1	+15.5 ❌
+ Augmentation (04)	303.0	-5.5 ✅
+ Cosine Scheduler (05)	335.0	+26.4 ❌
All Features (Improved) (06)	303.3	-5.2 ✅
Attention Basic (07)	304.4	-4.2 ✅
+ Move Importance (08)	304.8	-3.8 ✅
+ All Features (Attn) (09)	286.5	-22.1 ✅

The 211 Elo gap between UltraBullet and Classical demonstrates that longer time controls produce moves more representative of genuine player strength. This suggests future work should consider time-control-specific models or weighted loss functions that account for move quality variance across speeds.



Future Work

Future Work:

- Implement engine-derived features (centipawn loss, blunder probability) from Stockfish
- Explore time-control-specific models to handle the variability between bullet and classical games
- Further hyperparameter tuning to reduce overfitting and improve model stability
- Explore transformer-based architectures as an alternative to LSTM
- Investigate why cosine scheduler alone decreased performance, potentially exploring different learning rate schedules

Conclusion

We successfully improved upon the baseline RatingNet model by implementing attention mechanisms combined with architectural enhancements, achieving a 22.1 Elo reduction in MAE (from 308.6 to 286.5).

References

- Omori, M., & Tadepalli, P. (2024). Chess rating estimation from moves and clock times using a CNN-LSTM. arXiv preprint arXiv:2409.11506.
- Chowdhury, S., Iacopini, I., & Battiston, F. (2023). Quantifying human performance in chess. Scientific Reports, 13(1), 2113.
- Glickman, M.E. (2012). Example of the Glicko-2 system. Boston University.
- IEEE. (2024). Big Data Cup 2024: Predicting chess puzzle difficulty. KnowledgePit.
- Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems.