Project Proposal: LLM-Powered Chess Bot for Move Prediction and Strategy Evaluation

Kyle Tranfaglia, Dustin O'Brien

Motivation:

Chess AI has evolved from rule-based engines like Stockfish to deep-learning models like AlphaZero. Large Language Models (LLMs) have recently shown promise in predicting optimal moves and evaluating positions by leveraging vast chess game datasets. Like natural language, chess involves nuanced, fuzzy logic, benefiting from extensive online databases. LLM-based chess AI can efficiently analyze large-scale game data, and training such models on consumer-grade hardware is increasingly feasible. However, modern chess engines typically fail to replicate human-like gameplay; thus, this project will use attention models to consider human playstyles over theoretically optimal play.

Literature Review:

A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play[1]

This groundbreaking paper introduced AlphaZero, which achieved superhuman performance in chess, shogi, and Go using reinforcement learning without human knowledge. The algorithm combines a neural network, and Monte Carlo tree search to develop strategic understanding through self-play, a concept our project utilizes by incorporating probabilistic reasoning into move selection.

The Chess Transformer: Mastering Play Using Generative Language Models[2]

This research explored using GPT models for chess move prediction by treating the game as a language with moves as tokens. Their approach demonstrated the viability of using language models for chess, which directly informs our LLM-based methodology while we extend their work by incorporating probabilistic evaluation of position quality.

Chess as a Testbed for Language Model State Tracking[3]

This study showed how transformer architectures can effectively encode chess positions and track game states through attention mechanisms. Their findings on position representation are particularly relevant to our project's encoding of board states for the LLM, and we build upon their work by focusing specifically on move prediction probability.

Bayesian optimization with large-scale Monte Carlo tree search: A case study in chess[4]

This paper explored Bayesian approaches to chess move selection, combining probabilistic reasoning with traditional search techniques. Their methodology for representing uncertainty in position evaluation provides valuable insights for our probability-based heuristic. However, our approach differs by leveraging large datasets of human games rather than strict Monte Carlo simulations.

Aligning Superhuman AI with Human Behavior: Chess as a Model System[5]

This research developed models that predict human moves rather than simply optimal moves, focusing on modeling typical player behavior across skill levels. This work is particularly relevant to our project as we similarly aim to create a system that plays in a human-like manner using probabilistic decision-making. However, we extend it by incorporating win probability prediction into move selection to construct a strong chess bot that plays human-like moves.

---

[1] Silver et al.
[2] Noever et al.
[3] Toshniwal et al.
[4] Guo et al.
[5] McIlroy-Young et al.

Hypothesis and Expected Outcome:

We hypothesize that an LLM-powered chess bot utilizing probabilistic decision-making will demonstrate move patterns more closely resembling human experts than traditional chess engines while maintaining competitive performance. Specifically, we expect that the model will develop a distinctive "playing style" that consistently favors certain types of positions, similar to human players; when benchmarked against Stockfish, our model will select the engine's top three recommended moves approximately 80% of the time, while occasionally making "human-like" suboptimal but reasonable choices; The system will achieve an estimated Elo rating of 2200 when tested with Stockfish evaluation, demonstrating strong but not superhuman performance. These outcomes would validate our approach of using probability distributions for move selection as a viable alternative to traditional minimax algorithms with alpha-beta pruning, particularly for applications where human-like play is preferred over perfect play. We expect a human-player dataset to produce human-oriented gameplay instead of theoretically optimal gameplay, a factor not often accounted for in Chess bots.

Experiments and Methodology:

Our methodology integrates data preprocessing, model development, and experimental validation through structured experiments. First, we will extract and preprocess a Lichess dataset, converting Algebraic Notation to tensor-based board representations with encoded outcomes. We will implement a Transformer-based architecture with attention mechanisms, training it to output win/loss/draw probabilities for each potential move. For evaluation, we will conduct four key experiments: Move Prediction Accuracy – comparing our model's suggested moves against actual moves played by experts; Stockfish Comparison – measuring the evaluation score differences between model-selected moves and Stockfish recommendations at various depths; Cross-validation Testing – evaluating the model's performance against reserved test data using various metrics; Live Performance – deploying the bot against online chess bots and human players with Stockfish analysis to measure its practical Elo rating. Each experiment will systematically isolate specific aspects of performance, from pure prediction accuracy to real-world gameplay effectiveness. Throughout testing, we will record both quantitative metrics and qualitative observations about the model's "playing style" and strategic tendencies to evaluate its human-like qualities alongside its competitive strength comprehensively.

Success Metrics:

Stockfish top move alignment, Elo rating achievement (2200), observation of human-like play patterns, and probability model effectiveness.

Dataset:

This study will utilize a dataset of 6.25 million chess games sourced from an open chess database by Lichess.org, covering games played online on the Lichess website during July 2016 (Revel, 2016). The dataset includes fifteen features: Event (type), White, Black (ID), Result (outcome), UTCDate (date of match), UTCTime (time of match), WhiteElo, BlackElo (Elo rating), WhiteRatingDiff, BlackRatingDiff (post-match Elo difference), ECO (opening code), Opening (name), TimeControl (plus increment), Termination (type), AN (moves as text). Given the immense dataset size, completeness, and abundant features, it is ideal for analysis.

Halfway Milestone:

At the halfway milestone, we aim to have a functional prototype trained on a subset of the Lichess dataset, capable of predicting move probabilities and basic win/loss/draw evaluations. Initial tests will assess move prediction alignment with expert play and may use Stockfish evaluations integrated to refine move selection. This will establish a foundation for improving strategic consistency and gameplay.

References

Guo, Xintong, Satinder Singh, Honglak Lee, Richard L. Lewis, and Xiaoxiao Wang. 2022. *"Bayesian Optimization with Large-Scale Monte Carlo Tree Search: A Case Study in Chess."* arXiv preprint arXiv:2201.11678. https://arxiv.org/abs/2201.11678.

McIlroy-Young, Reid, Srijan Sen, Jon Kleinberg, and Ashton Anderson. 2020. *"Aligning Superhuman AI with Human Behavior: Chess as a Model System." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1677–87. https://doi.org/10.1145/3394486.3403080.

Noever, David, Matt Ciolino, and Joshua Kalin. 2020. *"The Chess Transformer: Mastering Play Using Generative Language Models."* arXiv preprint arXiv:2008.04057. https://arxiv.org/abs/2008.04057.

Revel, A. 2019. *"Chess Games (Version 1.0)."* Kaggle. https://www.kaggle.com/datasets/arevel/chess-games.

Silver, David, 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. 2018. *"A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play." Science* 362 (6419): 1140–44. https://doi.org/10.1126/science.aar6404.

Toshniwal, Shubham, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2022. *"Chess as a Testbed for Language Model State Tracking."* arXiv preprint arXiv:2202.05556. https://arxiv.org/abs/2202.05556.