Incorporating Uncertainty in Chess Engines: A Bayesian Reinforcement Learning Approach to Complex Decision-Making

Nikhil Murlidhar Shanbhogue, Sandilya Sai Garimella, Vidhya Kewale

September 2024

1 Overview

Existing chess engines like Stockfish and AlphaZero have set remarkable benchmarks in computational chess by utilizing different methodologies [4, 7]. Stockfish relies on classical algorithms like alpha-beta pruning combined with extensive heuristics to evaluate positions, and recently updated with Efficiently Updatable Neural Network (NNUE) [5]. In contrast, AlphaZero uses deep reinforcement learning with Monte Carlo Tree Search (MCTS) guided by neural networks to learn optimal policies through self-play [8]. However, both approaches predominantly use point estimates for value functions and policies, without explicitly modeling the uncertainty inherent in complex decision-making processes.
In the context of chess, uncertainty arises from several factors:

1. **Limited Exploration**: Certain positions may be underrepresented during training, leading to high uncertainty in their evaluations.
2. **Complexity of Positions**: Positions with vast branching factors and numerous plausible continuations introduce uncertainty due to the computational infeasibility of exhaustive analysis.
3. **Opponent Play**: The unpredictability of an opponent's strategy, especially when multiple strong moves are available, adds another layer of uncertainty.

A Bayesian Reinforcement Learning (BRL) approach addresses these challenges by incorporating uncertainty directly into the learning and decision-making process [2, 1]. Unlike the traditional frequentist methods used by AlphaZero, which optimize expected rewards based on fixed estimates, BRL maintains a probability distribution over the value functions and policies. This probabilistic framework allows the engine to quantify confidence levels in its evaluations and make decisions that consider both the expected reward and the associated uncertainty.

Technically, BRL updates prior beliefs about the environment or model parameters using observed data to compute posterior distributions. This Bayesian updating is crucial in chess, where new positions continually provide fresh information that can refine the engine's understanding of optimal play. By accounting for uncertainty, the engine can enhance exploration by focusing on underrepresented positions with high uncertainty, manage complexity by balancing expected rewards with evaluation reliability, and anticipate opponent moves by modeling the probability distribution of their potential strategies, leading to more robust and adaptable play.

2 Literature Survey

Artificial intelligence in chess has seen many advancements in recent years. Many approaches leverage deep learning, reinforcement learning, and Bayesian methods to create increasingly sophisticated chess engines. Key developments in AI chess algorithms include groundbreaking techniques like AlphaZero's deep reinforcement learning, comparative analyses of different AI approaches, and the potential of Bayesian methods for handling uncertainty in chess game-play. Exploring these perspectives establishes a comprehensive foundation for the proposed BRL approach to chess engines, which aims to combine the strengths of deep learning with probabilistic reasoning to create more robust and adaptable chess AI.

Silver et al. introduced AlphaZero, a groundbreaking reinforcement learning algorithm that achieved superhuman performance in chess, shogi, and Go without human knowledge [8]. Key aspects include using deep neural networks for policy and value estimation, Mote Carlo Tree Search for action selection, and tabula rasa learning through self-play. While revolutionary, AlphaZero does not explicitly model uncertainty, which our proposed BRL approach aims to address.

Several papers provide comparative analyses of different chess AI approaches. Maharaj et al. offer an overview of the evolution of chess engines from expert systems to machine learning approaches, discussing the strengths and limitations of different paradigms [4]. Sadler and Regan analyze AlphaZero's chess play, providing insights into its strategic and tactical innovations and unique decision-making process [6]. These works both highlight the potential for novel AI approaches in chess, the impact of deep learning and reinforcement learning on chess AI, and the potential for AI to generate new knowledge and enhance human creativity in chess.

The Bayesian perspective and methods in reinforcement and chess AI are explored in several key works. Ghavamzadeh et al. provide a comprehensive overview of Bayesian methods in reinforcement learning, covering model-based and model-free BRL approaches, Bayesian exploration strategies, and application in various domains [2]. Horacek explores the application of uncertainty reasoning in computer chess, focusing on representing and propagating uncertainty in position evaluation, handling incomplete information in chess game-play, and strategies for decision-making under uncertainty [3]. Baum presents a Bayesian perspective on game-playing, particularly in chess [1]. The paper explores modeling opponent behavior probabilistically, using Bayesian updating to refine strategy during game-play, and balancing exploration and exploitation in move selection. BRL approaches can handle uncertainty and exploration-exploitation trade-offs, provide valuable insights into incorporating uncertainty in chess engines, and offer foundational ideas for applying Bayesian reasoning to chess.

Significant progress has been made in applying reinforcement learning to chess, particularly through the success of AlphaZero, while its limitations in handling uncertainty have been pointed out. The potential of Bayesian methods in reinforcement learning is evident in their ability to incorporate uncertainty into decision-making– crucial for complex games like chess. The works discussed on uncertainty reasoning and Bayesian approaches in chess offer key insights for improving strategic evaluations and enhancing adaptability. By integrating deep reinforcement learning with Bayesian uncertainty modeling, our project aims to create a more robust and flexible chess engine capable of navigating intricate decision-making processes with greater confidence.

3 Datasets

For our experiments, we will utilize the Lichess.org Open Database, which contains over 6 billion standard-rated chess games. This dataset is publicly available and well-suited for open-source development, offering high-quality game data that reflects competitive and serious play.

Each game in the Lichess dataset is stored in Portable Game Notation (PGN) format, providing detailed metadata and game information, including:

- **Player Information**: Player usernames, Glicko-2 ratings (`WhiteElo`, `BlackElo`), and rating changes, which help model player skill progression.
- **Game Outcome**: Match results (`1-0`, `0-1`, `1/2-1/2`) and termination conditions, useful for constructing reward signals in the Bayesian Reinforcement Learning (BRL) model.
- **Time Control**: Time control settings for the game, relevant to different pacing in decision-making and strategy formulation.
- **Opening Information**: ECO (Encyclopedia of Chess Openings) codes and opening names, offering insights into early-game strategies.
- **Moves and Annotations**: Move sequences and Stockfish engine evaluations (e.g., [`%eval 0.17`]), essential for move selection modeling and estimating the value of game positions.

To train the BRL model, we will focus on standard rated games from the Lichess dataset to ensure high-quality data reflecting serious and competitive play.

The preprocessing will involve filtering out non-standard variants and incomplete games to maintain consistency, parsing PGN files using libraries like `python-chess` to extract moves, player ratings, and game outcomes, and converting board positions into numerical formats (e.g., bitboards or tensors) compatible with the BRL model.

We will identify all legal moves at each state to construct the action space and incorporate Stockfish evaluation scores where available to enhance value estimation. Using this processed dataset, we will establish prior probability distributions over model parameters (e.g., neural network weights) to reflect initial beliefs about game dynamics.

Posterior distributions will be updated using observed game data through techniques like variational inference or Monte Carlo methods, allowing us to model uncertainty by maintaining distributions rather than point estimates. Game results will serve as reward signals to update value function distributions, reinforcing effective strategies, and we will analyze the diversity of opponent play to model probability distributions of opponent moves.

Lastly, based on the feasibility and time requirement, we plan to incorporate FICS (Free Internet Chess Server) and KingBase Lite Databases in the training data. While Lichess provides a massive dataset with wide range of skill levels, FICS contributes diversity in time control and rating levels across rapid, blitz, and standard games, and KingBase Lite focuses on elite games, which can allow the model to learn the highest levels of strategy and tactical play. This ensures that the model is trained on a comprehensive and diverse range of chess data, from casual online games to professional tournaments, thereby helping it generalize across different playing styles and conditions.

4 Plan of Activities

Phase 1: Theoretical Foundation and Setup

Our project will begin with a comprehensive development of the Bayesian Reinforcement Learning (BRL) formulation for chess. We will meticulously compare this BRL approach with AlphaZero's Reinforcement Learning methodology, identifying the parameters to be estimated as distributions and defining the Bayesian update mechanism using Bayes' rule. This phase will also involve specifying the optimization problem and determining the role of Monte Carlo Tree Search (MCTS) in our BRL approach. Concurrently, we will set up our chess environment using the python-chess library, ensuring seamless integration for chess simulation. We'll establish a robust data pipeline for loading and processing PGN datasets from Lichess, and verify compatibility between python-chess and the BayesRL library to ensure smooth operations in subsequent phases.

Phase2: Implementation and Initial Training

Following the theoretical groundwork, we will move on to implementing the core BRL algorithm using the BayesRL Python library. This implementation will be tightly integrated with the python-chess environment. We'll design and implement a comprehensive training pipeline, including data preprocessing for Lichess datasets. Initial training runs and parameter tuning will be conducted to refine our approach. Simultaneously, we'll set up Leela Chess Zero as an AlphaZero surrogate, developing an interface for position analysis. As a contingency, we'll implement a fallback option using Sunfish for board evaluation. This phase will culminate in creating a unified interface for comparing our BRL approach with AlphaZero-like methodologies.

Midway Submission

At the project's midpoint, we will prepare a comprehensive progress report detailing our theoretical formulation and initial implementation results. This report will also address challenges encountered during the first half of the project and propose solutions. We'll provide an updated timeline and set revised goals for the second half of the project, ensuring we remain on track for successful completion.

Phase 3: Advanced Implementation and Comparative Analysis

Building on our initial results, we will refine the BRL algorithm, implementing ”uncertainty metrics” for move evaluation and optimizing our method for efficient next-move prediction. This refined algorithm will undergo extensive training. We'll then develop a comprehensive evaluation framework to conduct a thorough comparative analysis between our BRL method and Sunfish. This analysis will focus on move quality and uncertainty quantification, investigating the strengths and weaknesses of our BRL approach across various chess scenarios.

Final Experimentation and Reporting

In the final phase, we will design and conduct additional experiments to further validate our BRL method. These experiments will explore edge cases and challenging chess positions, analyzing our method's performance across different game stages (opening, middlegame, endgame). The project will conclude with a detailed report, providing a comprehensive explanation of the theoretical methods for both AlphaZero and our BRL approach. We'll describe implementation details, including challenges faced and solutions devised. The report will present our experimental setup and methodology, followed by a thorough analysis and discussion of results, emphasizing the comparison with Sunfish. We'll draw conclusions on the effectiveness of the BRL approach in chess and suggest future research directions and potential improvements.

References

[1] Eric B Baum. How a bayesian approaches games like chess. In *Proceedings of the AAAI'93 Fall Symposium*, pages 48–50, 1993.

[2] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[3] Helmut Horacek. Reasoning with uncertainty in computer chess. *Artificial Intelligence*, 43(1):37–56, 1990.

[4] Shiva Maharaj, Nick Polson, and Alex Turk. Chess ai: competing paradigms for machine intelligence. *Entropy*, 24(4):550, 2022.

[5] Andrea Manzo and Paolo Ciancarini. Enhancing stockfish: A chess engine tailored for training human players. In *International Conference on Entertainment Computing*, pages 275–289. Springer, 2023.

[6] Matthew Sadler and Natasha Regan. Game changer. *AlphaZero's Groundbreaking Chess Strategies and the Promise of AI. Alkmaar. The Netherlands. New in Chess*, 2019.

[7] Quazi Asif Sadmine, Asmaul Husna, and Martin Müller. Stockfish or leela chess zero? a comparison against endgame tablebases. In *Advances in Computer Games*, pages 26–35. Springer, 2023.

[8] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.