Leela Chess Zero and the Human Play

Yash Kumar Shukla

Abstract
This paper presents a study between Leela Chess Zero (Lc0), a neural network-based chess engine and elite-level human chess play. The objective of this research is to explore how artificial intelligence, reinforcement learning and neural network-based systems, are reshaping our understanding of chess. We have extracted and modified large datasets from public domain comprising over 2.7 million self-play games by Lc0 and 1.6 million games played by elite human players (rated above 2500 on Lichens platform) and pre-processed this data to enable robust statistical analyses. With the help of a systematic approach and Python-based data analysis, we have evaluated various dimensions of game play under different conditions. This study investigates not only the strengths of neural network-based engines in decision-making and pattern recognition but also highlights the blind spots and biases in elite human play of opening selection and game outcomes. Charts and visualizations have been shown to clearly represent key differences and correlations between the two types of play. Our findings suggest that AI engines like Lc0 do not just replicate human strategies but often uncover new approaches and deeper evaluations that challenge long human theories. The analysis shows how neural network-based engines expose flaws in classical human openings and they often suggest alternative continuations that were previously unexplored. Also, we discuss the role of tree search in enhancing prediction accuracy, especially in critical game phases such as endgames. This study aims to provide a bridge between human level game understanding and AI precision in chess, offering insights that can help chess professionals to enhance their play and rethink on traditional strategies. The results show how AI can be used as a powerful tool for chess analysis, training, and innovation. In the long term, this research can influence how chess is taught, analysed, and played, leading to a re-Evaluation of classical openings and the discovery of new theoretical lines.

Keywords- Leela Chess Zero, Artificial Intelligence, Neural Networks, Chess Engines, Python, AI and Comparison, Opening Theory, Endgame Analysis

I. Introduction

A. Background

Modern chess is growing rapidly, and it has seen huge improvement with the arrival of Artificial Intelligence [1]. Earlier we saw brute force [2], tree-based algorithms [3] to find out the best possible move in a particular chess position and the advantage it gives in the play. This method requires a lot of computation and resources as chess can get complicated with only the first four moves of the game having almost 318 billion ways to approach the game. In the previous development phases computers like Deep Blue [4] developed by IBM which defeated the world champion Garry Kasparov were a breakthrough to provide quality chess training and insights of the games to world class players. Chess engines such as Stockfish work by using a combination of algorithms and techniques to evaluate chess positions and find the best possible moves. It utilizes a search algorithm alpha-beta search to explore a tree of potential moves. It also employs a heuristic evaluation [5] function which has evolved from a hand-coded function to a neural network (NNUE) in more recent versions[6] to assess the value of different positions. And to this day we have stepped into the world of AI and deep learning where we are familiar with reinforcement learning based chess engines such as Alpha Zero [7] and Lc0 which utilize neural networks [8] to analyze and solve chess positions. The advantage of using reinforcement learning based chess engines is that they can get trained from the human available chess game database and exhibit human like play nature. Chess players follow strategies and opening theories [9] and are

DOI: https://doi.org/10.5281/zenodo.15301209

taught to capitalize in the middle game and the center of the board. Theories like grandmasters prefer bishops rather than knights as they cover long diagonals of the board and are efficient in endgame strategies while some players rely on the forking capabilities of the knight and its game complications to expand the chess horizons. This kind of behavior can be exhibited by reinforcement learning based chess engines to better help chess players with the depth of the game and research on classical or modern chess opening ideas.

B. Problem Statement

In the chess world, a debate exists between how elite human players approach the game and how modern chess engines compute optimal solutions. Many grandmasters and elite players have expressed concerns that chess engines, while extremely powerful, do not always align with the theoretical frameworks and principles that humans rely upon. Chess engines solve problems through deep brute-force calculations and pattern recognition, rather than by following traditional strategic lines. This different approach has sparked ongoing debates in the chess community regarding the interpretability and value of engine-recommended moves.

Top-level human players structure their play around established theories, emphasizing the importance of control over the centre, timely piece development, king safety through castling, and positional imbalances such as bishop pair advantage, knight superiority in closed positions. These principles are supported by centuries of collective learning and have been passed down through chess literature, coaching, and historical analysis. For example, bishops are preferred in endgames because of their long diagonal covering capabilities, whereas knights are highly valued in complex middle-game positions for their forking abilities and unpredictability in deep games. This narrative forms the foundation of human chess learning, guiding players through the opening, middle game, and endgame phases.

Chess engines often propose moves that defy conventional wisdom, opting for lines that may appear counterintuitive to human understanding. These moves may indeed look like objectively optimal, but they often lack theoretical justification, making it difficult for players to understand and evaluate their logic. Also, engines can overlook alternative moves that may be slightly less optimal in evaluation but are easier for humans to follow and study, thus offering greater long-term value.

This underscores the potential of neural network-basedchess engines like Lc0, which combine computational strength with reinforcement learning from self-play to serve as a bridge between theory and calculation. Unlike in older chess engines that rely on deterministic search, neural network-based chess engines learn through experience and try to rediscover strategically sound patterns without being explicitly made to do so. As a result, they can provide position insights and decision trees that are not only strong but also theoretically meaningful, helping humans to expand their understanding of both classical and modern positions.

The analytical depth of model trained from reinforcement learning can be used to train and evaluate human games, potentially enhancing the theoretical version of chess. By studying AI-generated games alongside classical games, players and researchers can explore new ideas of existing openings, discover unexplored variations, and develop rich endgame heuristics. The blend of AI enabled game precision and human-curated chess theory may hold the key to unlock undiscovered aspects of the game.

Chess remains an unsolved problem, and its complexity is further amplified in alternative formats such as freestyle chess (also known as Chess960 or Fischer Random Chess), where the back-rank pieces are shuffled randomly. In these formats, traditional opening preparation becomes useless and even the strongest grandmasters have struggled to adapt to it. This shows that human players rely not only on real-time calculation but also on memorized theoretical knowledge and historical game references. The freestyle format strips players of their opening familiarity and challenges them to rely solely on principled understanding and adaptability—areas where neural network engines may excel due to their ability to evaluate unfamiliar positions without prior opening knowledge.

DOI: https://doi.org/10.5281/zenodo.15301209

This study shows that working on the strengths of neural network-based engines, such as Lc0, can give a new model of learning where computational analysis is combined with theoretical exploration. This synergy can ultimately contribute to a more complete understanding of chess, pushing both human and machine toward discovering new dimensions of play and perhaps inching closer to a more solved or explainable model of the game.

II. Related Work

A. Lc0 Against Endgame Tablebases

Previous studies have demonstrated that neural networks tend to approach perfect play as the training progresses, particularly when the model is exposed to a diverse and sufficiently large set of training positions [10]. The refinement of neural network parameters over time enables the model to better approximate optimal policy and value functions. However, the intrinsic limitations of neural networks in handling deep combinatorial decision spaces such as those encountered in chess necessitate the use of tree search algorithms to further enhance performance.

Monte Carlo Tree Search (MCTS), as in engines like Alpha Zero and Lc0, plays a crucial role in enhancing prediction accuracy, especially in complex game states. Empirical evidence suggests that neural network error rates tend to decrease at decision depths where the sample density is high, indicating that frequent exposure to similar positions allows the model to generalize more accurately. Conversely, in rare or highly specific positions such as in deep endgames, the model alone may exhibit inaccuracies due to sparse training data.

In such scenarios, tree search significantly improves the quality of decision-making, even at relatively shallow depths. Specifically, in endgames, tree search facilitates a higher rate of perfect play, compensating for the limitations of the policy network by thoroughly evaluating variations through playouts. This corrective mechanism is particularly effective when the value head of the neural network maintains high accuracy, allowing tree search to confidently refine move selection based on reliable evaluations.

However, it is important to note that tree search can introduce negative impacts on overall prediction accuracy in cases where the value head error is substantial. In such situations, the tree may over commit to suboptimal lines based on flawed evaluations, leading terrors during move selection. Therefore, the effectiveness of tree search is tightly coupled with the reliability of the underlying neural network evaluations, highlighting the importance of joint optimization in both policy and value heads.

B. Stockfish or Lc0

Recent comparative studies have revealed nuanced differences in endgame prediction accuracy between Stockfish [11], a deterministic, rule-based engine relying on handcrafted evaluation functions and alpha-beta pruning, and Lc0, a neural-network-based engine trained via reinforcement learning and self-play. In simplified board states such as 3-piece endgames, it has been observed that Stockfish's policy function consistently performs at a level equal to or superior to that of Lc0 in terms of predicting the perfect move [12]. This aligns with Stockfish's design philosophy, which excels in calculating precise tactical sequences and table base lookups in minimal-piece scenarios.

However, in 4-piece endgames, Lc0 demonstrates a clear edge. It tends to make fewer mistakes compared to Stockfish, particularly in positions that require strategic nuance rather than brute-force calculation. This suggests that Lc0'sdeep learning framework is more effective at generalizing positional principles, even in rare or sparsely encountered endgame configurations. One notable strength of Lc0's is its superior ability to evaluate and recognize weaknesses in positions, a trait attributed to its learned representations of long-term positional imbalances rather than short-term tactical threats. When tree search is applied, both engines significantly enhance their performance, reducing their respective error rates and bringing their predictive accuracy closer together. The gap in performance narrows, highlighting the critical role of search in refining engine decisions, especially when

DOI: https://doi.org/10.5281/zenodo.15301209

navigating low-piece count positions with subtle winning or drawing chances.

Interestingly, there is a divergence in prediction specialization between the two engines. Stockfish demonstrates higher reliability in predicting winning positions, likely due to its tactical depth and efficient pruning mechanisms. Conversely, Lc0 is more adept at correctly identifying drawing positions, an ability that likely stems from its probabilistic evaluation approach and its capacity to model game-theoretic equality over long sequences.

Another area where Lc0 shows clear superiority is in scenarios where the opponent's last remaining pawn is under threat. Hence, Lc0 commits less errors as compared to Stockfish engine, indicating a stronger edge over pawn endgames. This further supports the view that Leela's strength lies not only in its learned policy but also in its intuitive, human-like assessment of position quality, even without explicit table base access.

III. Methodology

We have followed certain steps to neatly extract and clean the dataset. All chess games were in PGN (Portable Game Notation), a format to save chess data. We converted this data into csv file format for better interpretation and further analysis. Then a simple yet effective data plotting was done to visualize the outcomes and understand the results.

A. Dataset

For this study we have used two datasets which were compiled to facilitate a comprehensive comparison between artificial intelligence-driven chess play and human expert-level performance. The first dataset comprises 2,756,982 self-play games generated by Lc0, sourced from an openly available Kaggle repository [13]. These games were played under various time controls and employed a policy temperature of 2.25, a parameter used to increase move diversity during training. All games were trained using the CUDA-fp16 backend, optimizing computational efficiency on GPU hardware. The dataset reflects a broad range of positions and strategic themes encountered during the self-play training cycle of Lc0, making it a rich resource for understanding AI learning behavior in chess.

The second dataset was downloaded from the Lichess.org platform [14], a widely used, open-source online chess server. This collection includes 1,645,041 games played by elite human players, specifically those rated 2500 and above, against opponents rated 2300 and above. To maintain high analytical quality, bullet and speed games were excluded, focusing the dataset on rapid, classical, and blitz formats where deeper strategic planning is more observable. The Lichess data offers a robust representation of expert-level human play and serves as a real-world benchmark for comparison with AI-generated games.

Both datasets are provided in Portable Game Notation (PGN) format and include a standardized set of metadata fields, such as Event, Site, Date, Round, White Player, Black Player, Result, ECO Code, Opening Name, Variation, Game Duration, Game Start and End Times, Play Count, and Time Control. These structured headers facilitate detailed filtering and parsing, enabling comparative analyses on multiple axes including opening theory, time management, game outcome trends, and positional depth.

B. Data Conversion

We have utilized python-chess library [15] to extract relevant headers for our data and pandas library to convert that into data frames. The data was then saved in CSV file. For Lc0 games a total of 15 PGN files were used which contains self-played games in different time controls. They were converted into CSV and then compiled together to finally create a large file of games. The same was repeated for Lichess Elite Database games which contained monthly games of players.

C. Exploratory Data Analysis

The game distribution of Lc0 shows that we have a highly balanced data containing almost equal amounts of won, lost and drawn games.

DOI: https://doi.org/10.5281/zenodo.15301209

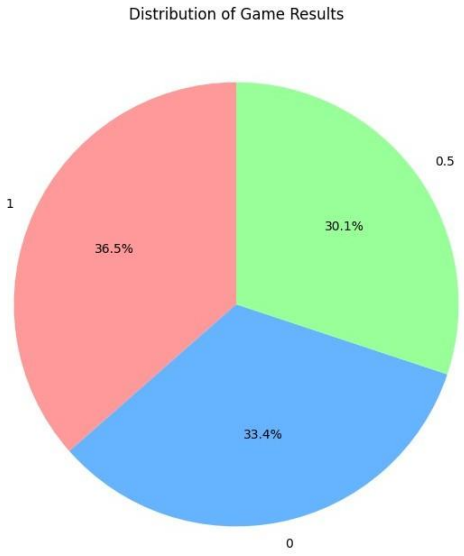

Fig. 1. Lc0 game distribution shows 36.5% white wins, 33.4% black wins and 30.1% draws shocasing highly balanced dataset.

The below table shows the most played openings by Lc0 and provides us an idea that how much the chess engine loves to play the Petrov opening.

TABLE I. CHESS OPENING DISTRIBUTION (MOST PLAYED)

Opening Name	No. of Games Played
Petrov	700195
Queen's Gambit Declined	657063
Queen's Pawn Game	534611
Queen's Gambit Accepted	512790
Queen's Gambit Accepted 4.e3	63235

The net performance score of the dataset also tells the reason behind why these openings are popular. It is seen that Leela has a significant edge in converting the games towards a decisive ending. Net performance score is calculated by subtracting total losses from total wins showing the net win result.

The below graph shows how well Lc0 performs with the top – 5 most played openings.

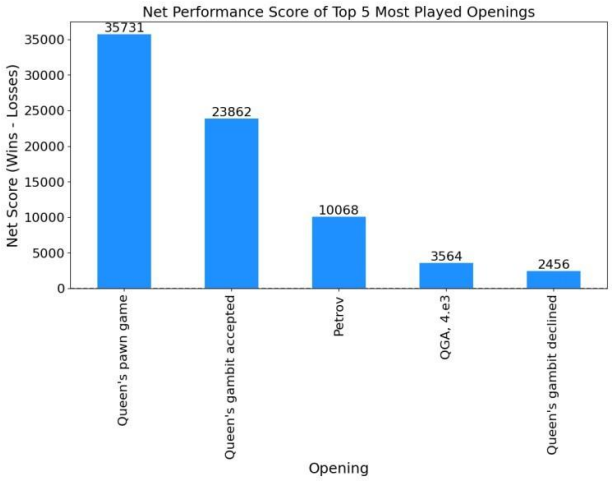

Fig. 2. Net Performance Score

Queen's Pawn Game performed the best with a massive net score of 35,731, meaning it led to many more wins than losses suggesting it's extremely effective. Queen's Gambit Accepted is second, with a strong net score of 23,862showing that accepting the gambit was also quite successful. Petrov Defense comes third with a net score of 10,068 which is still solid but significantly lower than the top two. QGA, 4.e3 (a variation of Queen's Gambit Accepted) and Queen's Gambit Declined are much lower, with net scores of 3,564 and 2,456, respectively. They still show a positive trend (more wins than losses), but not nearly as dominant.

The below table shows the least played openings by Lc0 which tells us Leela is not much interested in exploring these openings.

TABLE II. CHESS OPENING DISTRIBUTION (LEAST PLAYED)

Opening Name	No. of Games Played
Queen's Bishop Game	190
Bogo Indian Defense	86
Sicilian	77
Catalan	15
Giuoco Pianno	4

DOI: https://doi.org/10.5281/zenodo.15301209

The win rate does not tell much about the games as we have a highly balanced dataset therefore the chess engine does not bends towards a particular opening in terms of win rate. But we have calculated the net performance score which provides some estimated idea of which opening and variation works well with Leela.

TABLE III. NET PERFORMANCE SCORE OF OPENING AND VARIATION (TOP – 5)

Opening Variation Name	Net Score (Wins – Losses)
Queen's Pawn Game	34905
Queen's Gambit Accepted, Alekhine Defence	11832
Petrov, modern (Steinitz) attack	10070
Queen's Gambit Accepted	10033
Slav Defence	1985

The above table shows how well Leela performs with these openings. Elite players can study these openings in order to solidify their middle game strategies.

IV. Result

Leela vs Humans

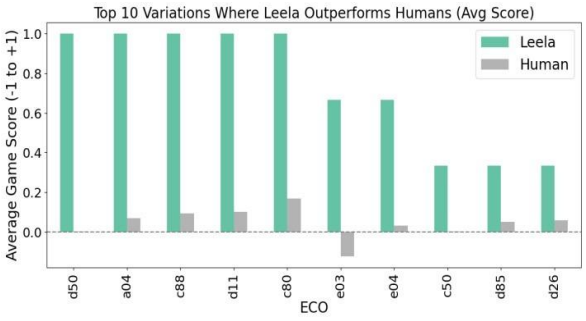

Fig. 3. Top variations where leela outperforms humans

Leela consistently scores higher average game results and human scores are notably lower. Humans can study how Leela plays these variations, especially d50 (Queen's Gambit Declined) and c88 (Ruy Lopez, Closed) and compare move-by-move ideas and plans.

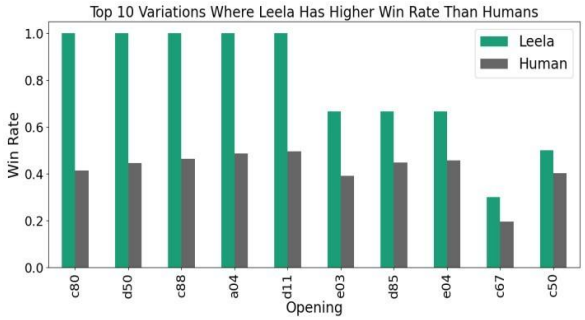

Fig. 4. Win rate comparision of Lc0 and Elite Human Chess Dataset

Again in openings like C80 (Ruy Lopez), D50, A04 Leela's win rate is much higher. Players can work on endgame conversion, attack coordination, and tactical awareness in these ECO lines where humans are missing chances.

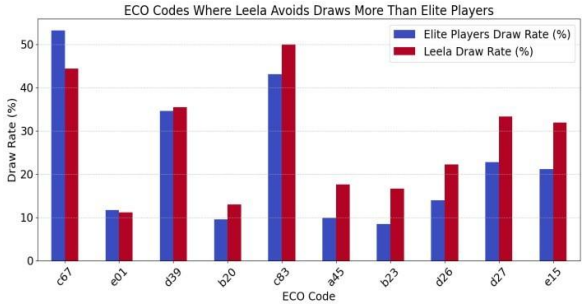

Fig. 5. ECO codes where Lc0 avoids draws more than elite players.

Leela has a lower draw rate than elite humans in many positions (e.g., c67, e01), showing that it pushes for decisive results even in commonly drawish openings. In e01 (Catalan Closed Game), Leela avoids draws better than elite players. This shows that humans can push for imbalanced positions and delay simplifications further going in depth of the game to find a solution.

V. Discussion

A. Limitations

While this study provides a valuable analytical comparison between the game play of Lc0 and elite human players on Lichess, it is important to recognize the inherent limitations in both the

DOI: https://doi.org/10.5281/zenodo.15301209

dataset and the scope of analysis, which frame the boundaries of the insights presented.

The Lichess dataset, although restricted to high-level games between players rated 2500+ and 2300+, does not include individual player ratings, titles, or detailed metadata. As a result, the analysis cannot account for player-specific factors such as form, style, or experience, nor can it differentiate between titled players like Grandmasters (GMs) and International Masters (IMs). The absence of these variables limits the ability to perform player-level stratification or assess variability in human decision-making based on rank or rating gap.

This study primarily focuses on openings and their variations, with statistical analysis centered on game initiation trends rather than move-by-move accuracy or game phase-specific evaluations. While this provides a strong foundation for understanding opening preferences and deviations, the research does not engage deeply with positional quality assessment or middle-game complexity. Tactical motifs, positional imbalances, and material sacrifices which often define human stylistic play are not quantitatively assessed in this work.

The study does not utilize evaluation engines like Stockfish or neural evaluators to classify move quality (e.g., best move, inaccuracy, blunder), nor does it assess how accurately players convert advantages or defend inferior positions. This restricts the analysis to aggregate patterns and results, rather than providing a granular picture of decision-making quality.

In terms of game phase coverage, endgames are only addressed at a high level, without leveraging endgame table bases or conducting exhaustive case-based reviews of theoretically drawn or won positions. Middle game statistics are not considered, and the rich tactical and strategic interplay that often occurs in this phase remains outside the scope of this study.

Time control data was not included in the analytical modeling. Although the dataset avoids bullet and hyper-speed games, a detailed analysis of time usage behavior (e.g., time spent per move or per phase) was not conducted, which may affect comparisons in practical decision-making scenarios.

The study takes a broad, statistically oriented approach to comparing Lc0 and elite human play, particularly in the domain of openings and game outcomes. However, due to the limitations in available data and analytical depth, the study refrains from drawing conclusions about in-game tactical depth, player-specific behaviors, or time-based decision efficiency. These limitations underscore the need for further work that can explore the game at a more detailed and nuanced level.

B. Future Directions

While this study establishes a foundational comparison between Lc0 and elite human players on Lichess, it also opens the door to numerous promising avenues for extended research and more granular analysis. These directions aim to deepen our understanding of neural network-driven chess engines in relation to human cognition, decision-making, and theoretical frameworks.

Current analysis primarily focuses on game outcomes and aggregate behavior, but contextual player-level information can provide a much richer perspective. Future studies should integrate player ratings, titles (e.g., GM, IM, FM), and even performance trends or historical form from the Lichess dataset. This would enable performance stratification across different skill levels and help in understanding how neural engines compare to human play at varying tiers of expertise. On the engine side, analyzing Lc0's behavior across multiple training checkpoints can reveal how its strategic understanding matures over time, potentially offering insights into how reinforcement learning systems modify their opening strategies, middle game and endgame.

While outcome-based metrics and opening trends offer a macro-level comparison, a micro-level evaluation of individual move quality would allow a more detailed contrast between Leela and human decision-making. Utilizing Stockfish's "best move" annotations, each move can be categorized into classes such as brilliant, best, excellent, inaccurate, mistake, or blunder. This analysis would reveal error profiles, positional tendencies, and the relative risk tolerance of each player type. Such fine-grained assessment is

DOI: https://doi.org/10.5281/zenodo.15301209

critical for mapping the cognitive footprint of each move, exposing how humans and AI diverge in calculation, intuition, and evaluation.

Though this study touches on endgame performance, a more exhaustive analysis using 7-piece table bases could determine the optimality of endgame conversion and defense. Evaluating how often human players and Lc0 deviate from table base-perfect play can help quantify practical versus theoretical proficiency. Moreover, such analysis could be extended to imbalanced material scenarios, such as rook vs. bishop and pawn, or queen vs. two rooks, where technique, understanding of drawing mechanisms, and patience are paramount.

Future work can use positional evaluation models, such as those derived from neural networks or expert annotations, to assess common motifs (e.g., outposts, pawn breaks, king safety) and tactical density. By comparing how Leela and elite humans approach these nuanced positions, researchers can uncover stylistic signatures such as whether Leela prefers simplification or complexity, or how often it sacrifices material for long-term positional gain.

When timestamps are available, future studies could explore how much time players and Leela spend per move, especially in critical positions. This can offer insight into efficiency, confidence, and complexity recognition, and might be especially useful in developing training methodologies for humans, such as prioritizing time usage in complex positions and simplifying when under time pressure.

A compelling reverse-engineering approach would involve training a neural network solely on high-level human games and then comparing its move choices and style to Lc0. This could answer pivotal questions such as: Which human heuristics persist in AI-trained on human data? How does it differ from reinforcement-learned engines? This approach could uncover latent patterns of human strategy, highlight strengths and blind spots, and allow the development of hybrid models that combine the interpretability of human reasoning with the computational power of AI.

While this paper focuses on Lc0, the chess AI ecosystem includes other influential engines like Stockfish, Komodo, and AlphaZero. A broader comparative study that includes these engines would provide a multi-dimensional view of AI behavior and strategic preferences. Each engine employs distinct architecture and search methodologies (e.g., NNUE in Stockfish, reinforcement learning in Alpha Zero), and comparing them can yield valuable insights into which approaches are most effective in various game phases and how they can be leveraged in training and pedagogy.

By pursuing these research directions, scholars and chess professionals alike can deepen their understanding of AI-human dynamics in chess, refine educational tools, and perhaps push the boundaries of both artificial and human strategy. This approach not only promises to enrich theoretical chess knowledge, but also holds potential for creating more interpretable, cooperative AI systems that align more closely with human thought processes.

VI. References

[1] M. Sadler and N. Regan, "Game Changer: AlphaZero's Groundbreaking Chess Strategies and the Promise of AI," New in Chess, 2019.

[2] J. Schaeffer et al., "A re-examination of brute-force search," in Proc. AAAI Fall Symp. on Games: Planning and Learning, Menlo Park, CA, USA, 1993.

[3] W. Fredlund and H. Wigforss, "Comparative Analysis of Monte Carlo Tree Search and Alpha-Beta Pruning in Chess AI Development," 2025.

[4] M. Campbell, A. J. Hoane Jr, and F. Hsu, "Deep blue," Artificial Intelligence, vol. 134, no. 1-2, pp. 57-83, 2002.

[5] J. Schaeffer, "The history heuristic and alpha-beta search enhancements in practice," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 11, pp. 1203-1212, 1989.

[6] A. M. Chitale et al., "Implementing the Chess Engine using NNUE with Nega-Max Algorithm," in 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2024.

DOI: https://doi.org/10.5281/zenodo.15301209

[7] T. McGrath et al., "Acquisition of chess knowledge in alphazero," Proceedings of the National Academy of Sciences, vol. 119, no. 47, pp. e2206625119, 2022.

[8] D. Klein, "Neural networks for chess," arXiv preprint arXiv:2209.01506, 2022.

[9] F. Pratesi, "Chess Theory–Its Structure And Evolution."

[10] R. Haque, T. H. Wei, and M. Müller, "On the road to perfection? evaluating Lc0 against endgame tablebases," in Advances in Computer Games, 2021, pp. 142-152.

[11] Stockfish Chess Engine, "Stockfish GitHub Repository," [Online]. Available: https://github.com/official-stockfish/Stockfish. [Accessed: Apr. 12, 2025].

[12] Q. A. Sadmine, A. Husna, and M. Müller, "Stockfish or Lc0? a comparison against endgame tablebases," in Advances in Computer Games, 2023, pp. 26-35.

[13] Lc0 Self-Play Chess Games – Bundle,[Online]. Available:https://www.kaggle.com/datasets/anthonytherrien/leela-chess-zero-self-play-chess-games-bundle, [accessed April 2025].

[14] LichessElite Database,[Online].Available:https://database.nikonoel.fr/, [Accessed April 2025].

[15] Python-Chess Library Documentation,[Online]. Available:https://python-chess.readthedocs.io, [Accessed April 2025].

DOI: https://doi.org/10.5281/zenodo.15301209