Towards Piece-by-Piece Explanations for Chess Positions with SHAP

Francesco Spinnato[1]

[1]University of Pisa, Largo B. Pontecorvo, 3, 56127, Pisa, Italy

Abstract

Contemporary chess engines offer precise yet opaque evaluations, typically expressed as centipawn scores. While effective for decision-making, these outputs obscure the underlying contributions of individual pieces or patterns. In this paper, we explore adapting SHAP (SHapley Additive exPlanations) to the domain of chess analysis, aiming to attribute a chess engine's evaluation to specific pieces on the board. By treating pieces as features and systematically ablating them, we compute additive, per-piece contributions that explain the engine's output in a locally faithful and human-interpretable manner. This method draws inspiration from classical chess pedagogy, where players assess positions by mentally removing pieces, and grounds it in modern explainable AI techniques. Our approach opens new possibilities for visualization, human training, and engine comparison. We release accompanying code and data to foster future research in interpretable chess AI.

Keywords

chess, explainable AI, shap

1. Introduction

Evaluating a chess position is a complex endeavor that combines long-term strategic foresight with immediate tactical precision. Contemporary chess engines summarize their assessments into a single scalar metric, typically expressed in centipawns, approximating the material advantage. This evaluation is indispensable for decision-making and training, yet it remains opaque: it does not reveal which specific positional elements underlie the overall judgment [1]. This lack of interpretability poses challenges for human players who seek strategic clarity, as well as for researchers striving to understand the internal logic of modern engines [2].

In contrast, the field of explainable AI (XAI) in machine learning has developed a rich array of methods for interpreting model outputs in classification and regression tasks [3]. Approaches such as feature attribution [4], saliency maps, and Shapley value decomposition [5] have proven effective in explaining model decisions across various data modalities, including tabular [3], image [6], and time series data [7, 8, 9]. Notably, many of these techniques are designed to be both model-agnostic and locally faithful, allowing them to be applied to any black-box model and to explain its behavior on a per-instance basis, providing actionable insights into complex decision-making and in critical domains [10, 11, 12]. Recent work has started to adapt these methods to games, particularly chess, by focusing on high-level features such as material balance and king safety [13, 14]. However, current approaches have yet to deliver fine-grained, per-piece explanations that are simultaneously additive, position-specific, and grounded in rigorous attribution theory.

In this paper, we introduce an interpretability framework based on SHAP (SHapley Additive exPlanations) [5], a principled XAI method grounded in cooperative game theory. We treat individual pieces as features and quantify their contributions by systematically ablating them, removing each piece in turn, and measuring the resulting shift in evaluation. These perturbations define a local neighborhood of similar board states, from which SHAP derives additive attributions that reflect not only the standalone value of each piece but also its interaction with others.

Proceedings of AI4HGI '25, the First Workshop on Artificial Intelligence for Human-Game Interaction at the 28th European Conference on Artificial Intelligence (ECAI '25), Bologna, October 25-30, 2025

francesco.spinnato@unipi.it (F. Spinnato)
https://fspinna.github.io/ (F. Spinnato)
0000-0002-3203-6716 (F. Spinnato)

Although removing pieces is not a legal operation within the rules of chess, the conceptual act of piece ablation has long been a tool in strategic analysis. Capablanca emphasized the value of simplification in order to clarify and exploit structural features, such as a pawn majority, that may become decisive in the endgame [15]. Authors such as Dvoretsky [16] and De la Villa [17] have explored this idea as both a pedagogical and analytical approach, encouraging players to mentally remove pieces to clarify the strategic essence of a position. This practice naturally invites evaluative questions such as: "What would happen if this piece were not on the board? Would my position improve?". Beyond evaluation, such simplification can also enhance a player's tactical recognition by revealing the most important pieces in the position. We aim to answer these questions with algorithmic precision, providing a systematic way to quantify the positional significance of each piece through principled feature attribution. By grounding interpretability in the well-established framework of SHAP, our method enables new forms of analysis and visualization. It can clarify the roles of individual pieces, help commentators explain sudden shifts in engine evaluations, and guide training tools in emphasizing the most critical components of a position. Furthermore, this methodology can be used to compare engines, revealing differences in how neural or hybrid systems assign value across structurally similar positions.

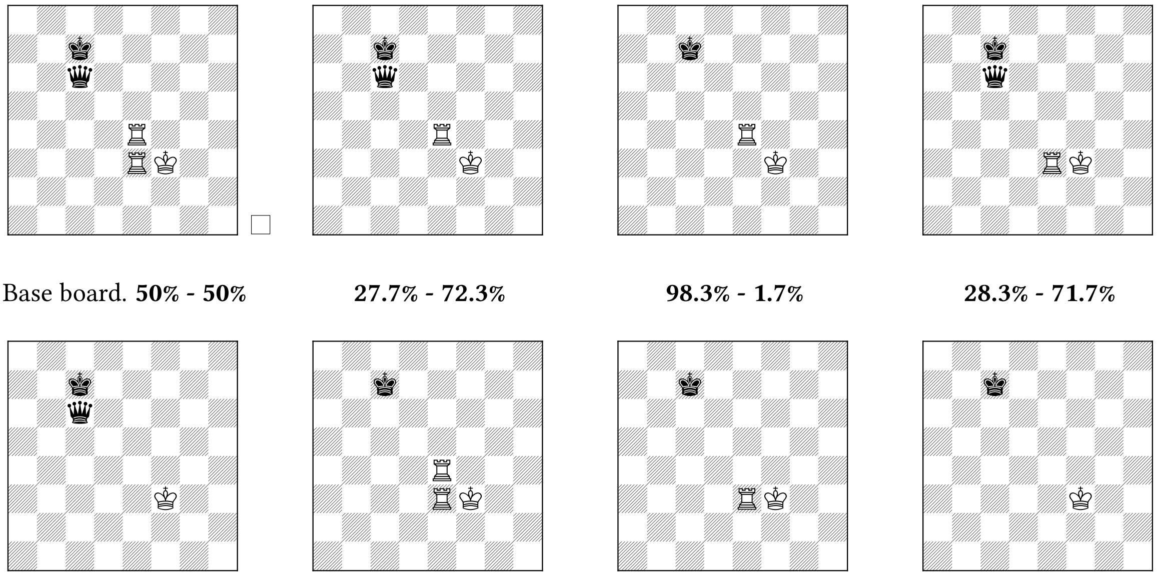
In summary, we propose a SHAP-based interpretability method for chess engine evaluations that leverages robust XAI methodology from machine learning and adapts it to the structured, combinatorial nature of chess. Specifically, our contributions are as follows: ***(1)*** We adapt SHAP to structured chess positions via a perturbation protocol based on piece ablation. ***(2)*** We compute fine-grained, per-piece local attributions that decompose the evaluation into interpretable, additive contributions. ***(3)*** We demonstrate with qualitative examples how these explanations can provide insights for several chess positional themes. ***(4)*** We release an open-source implementation to foster reproducibility and encourage further research in interpretable chess AI[1]. By aligning chess evaluation with mature techniques from explainable machine learning, our work opens a new direction for analyzing both the game and the algorithms that play it.

2. Related Work

Several recent studies have explored interpretability in chess engines from various perspectives. [13] employed Shapley value sampling to attribute Stockfish's evaluation output to a set of manually defined conceptual features, including "material," "passed pawns," and "king safety." Their analysis analyzed how different evaluation pipelines, classical versus efficiently updateable neural networks (NNUE), weigh these high-level concepts. While this demonstrates the applicability of Shapley values to chess evaluation, the granularity remains semantic and abstract; individual pieces and their positions are not explicitly taken into account. Our work extends this line of inquiry to the level of concrete board elements, applying Shapley values directly to individual pieces to yield localized attributions for specific positions. A complementary direction is explored in [18], where the authors trained a neural network to estimate the marginal win probability associated with each piece-square pair. Their model captures general trends, such as the high utility of knights on central outposts, by learning a global value function over a large dataset. However, these estimates reflect statistical averages across many games and positions. In contrast, our method produces per-position explanations, allowing us to assess the situational importance of each piece with additive precision based on its context.

Other work has applied attribution tools from explainable AI to neural chess models. In [14], the authors analyze an AlphaZero-style transformer using Integrated Gradients (IG) to assess which input features drive predictions of win probability. The attribution focuses on feature channels, such as piece maps or auxiliary indicators, aggregated across datasets. While this highlights the utility of gradient-based techniques for model interpretation, it remains coarse in resolution. Our method offers finer granularity by assigning attribution directly to the pieces present on the board in a given position, rather than to broader input modalities. A more perturbation-based approach is found in [19], which introduces SARFA (Specific and Relevant Feature Attribution) to explain move selection in chess and

[1] Code is available at: https://github.com/fspinna/chessplainer



Base board. **50% - 50%**

27.7% - 72.3%

98.3% - 1.7%

28.3% - 71.7%

1.7% - 98.3%

99.7% - 0.03%

98% - 2%

Empty board. **50% - 50%**

Figure 1: Perturbations generated by SHAP for evaluating the position on the top-left (white to move), evaluated using Lichess Win-Loss probability (% white to win - % black to win).

Go. SARFA identifies critical board squares by evaluating the impact of perturbations on move quality, balancing specificity with contextual relevance. Although SARFA and our method share a perturbation-based philosophy, their aims diverge: SARFA is designed to explain decisions (why a move was chosen), whereas we focus on evaluation decomposition (how each piece affects the position's value). Lastly, [20] examined the sensitivity of different engines to material and space advantage. Their results indicate that Stockfish places more weight on material factors than spatial ones, however, their analysis lacks the formal framework and guarantees provided by Shapley values.

Taken together, these works highlight a growing interest in interpretability, including statistical analysis, neural network modeling, and attribution techniques, reflecting a broader trend toward understanding how complex evaluation functions derive their outputs. However, despite this momentum, none of the aforementioned methods apply Shapley-based techniques to estimate the local, per-piece contribution within specific board states via targeted ablation. Our method fills this gap by adapting SHAP to the structured domain of chess, producing additive, locally faithful explanations that enable engine transparency, instructional tools, and strategic insight.

3. Explaining Chess Engines

A major challenge in explaining chess engine evaluations lies in the representation of the output. Most engines produce a scalar value in centipawns, where positive values indicate an advantage for White and negative values for Black. However, these scores are unbounded and require special handling for extreme cases such as forced mates, which are often represented by arbitrarily large constants. This lack of boundedness makes such outputs problematic for attribution methods like SHAP, which rely on finite and continuous model outputs to compute meaningful feature contributions.

To address this issue, we convert the centipawn evaluation into a probabilistic score, effectively framing the engine as a binary classifier that outputs the probability of a win for White. This transformation makes the output bounded in $[0, 1]$ and aligns well with SHAP's assumptions. Following the Lichess convention, we apply a logistic mapping from the centipawn score s to a probability p:

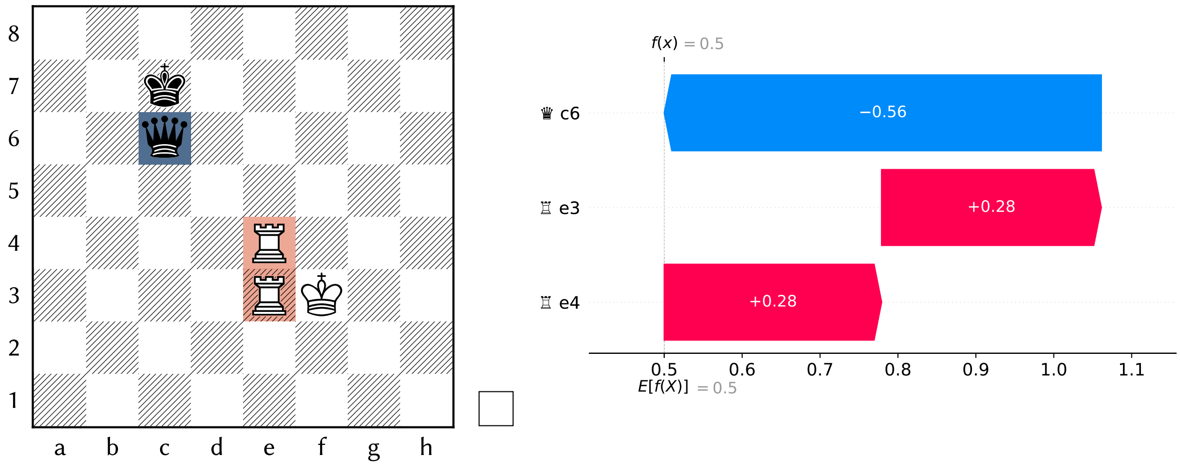


Figure 2: Explanation for the position in Figure 1. The two white rooks exactly offset the impact of the black queen, resulting in a perfectly drawn position.

$$p(s) = \frac{1}{1 + \exp(-\beta s)}, \tag{1}$$

where $\beta \approx 3.68 \times 10^{-3}$ is a calibration parameter. This maps $s = 0$ to $p = 0.5$, representing equal chances for both players, and ensures smooth asymptotic behavior for large centipawn values.

In this setup, we reframe the chess engine as a black-box classifier $f : X \to [0, 1]$ that maps a chess position $x \in X$ to the predicted probability that White will win the game. A position x is represented as a list of individual chess pieces currently on the board, where each piece is characterized by its type (e.g., rook, knight), color (white or black), and square location. In addition to the piece list, a complete position includes auxiliary metadata such as the side to move (White or Black), castling rights, and en passant availability, all of which are required for engine evaluation.

Our goal is to explain the prediction $f(x)$ by quantifying the marginal contribution of each individual piece to the overall evaluation. To this end, we adapt SHAP (SHapley Additive exPlanations) [5], a model-agnostic interpretability framework that decomposes $f(x)$ into additive attributions assigned to each feature, to our purpose. In our context, we view chess pieces as features, and exploit SHAP to express the predicted outcome as a sum of contributions from each piece present on the board.

Formally, let x' denote the set of non-king pieces present in the position x. Each element of x' corresponds to a specific piece instance, uniquely defined by its type, color, and square. We restrict our SHAP analysis to features in x' because removing either king would always result in an invalid position. Consequently, during SHAP perturbations, we hold both kings fixed and consider only the $n = |x'|$ non-king pieces as features. For any subset $S \subseteq x'$, we define x_S as the perturbed position containing the pieces in S together with both kings and the original metadata (e.g., side to move, castling rights).

The SHAP value ϕ_i assigned to a piece $i \in x'$ is defined as:

$$\phi_i(f, x) = \sum_{S \subseteq x' \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left[f(x_{S \cup \{i\}}) - f(x_S) \right], \tag{2}$$

where $f(x_S)$ denotes the engine's evaluation of the perturbed position, without piece i. An example of such perturbations can be viewed in Figure 1, starting from the original position (top-left), and removing pieces until there are none, (bottom-right). In general, SHAP explanations are defined over binary vectors $z' \in \{0, 1\}^n$ indicating feature presence or absence, and take the additive form $g(z') = \phi_0 + \sum_{i=1}^{n} \phi_i z'_i$. In our case, since all pieces in x' are present in the original position x, each

$z'_i = 1$, and the decomposition simplifies to the fully additive expression:

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i. \tag{3}$$

where ϕ_0 is the base value of the model when only the two kings are present. Since engines universally treat king-only positions as trivially drawn, $\phi_0 = 0.5$, corresponding to a neutral evaluation. This eliminates the need to estimate the baseline from data and grounds the explanation in a well-defined and interpretable configuration. An illustrative example of such an explanation, based on the base position from Figure 1, is presented in Figure 2. On the left, the position is visualized with each piece colored according to its SHAP contribution: red indicates a positive impact on White's winning probability, while blue indicates a contribution favoring Black. On the right, the same contributions are displayed numerically, showing how each individual piece shifts the evaluation from the base value $\phi_0 = \mathbb{E}[f(X)] = 0.5$, corresponding to a balanced position with only kings, to the full evaluation of the current position, $f(x) = 0.5$.

SHAP requires evaluating $f(x_S)$ for all subsets $S \subseteq x'$ of non-king pieces. However, not all such perturbed positions are legal or evaluable by a chess engine. In particular, configurations that result in illegal checks or impossible side-to-move conditions may be rejected. To ensure that most inputs to SHAP are valid, we implement a carefully designed perturbation strategy. First, as explained above, we never ablate the kings, thereby guaranteeing that each perturbed position includes exactly one white king and one black king. In cases where a perturbed position is deemed illegal by the engine, such as when a side is giving checkmate and it is their turn to move, we attempt to restore legality by flipping the board, i.e., switching the side to move. This often resolves inconsistencies introduced by the ablation process. If the position remains invalid even after this adjustment, we assign it a default evaluation of $f(x_S) = 0.5$, corresponding to a draw. This fallback ensures that SHAP's attribution remains well-defined while minimizing the introduction of bias or discontinuity in the output.

In summary, this approach yields an interpretable explanation of the engine's evaluation by attributing to each individual piece its marginal contribution to the predicted win probability for White. These attributions reveal how the presence of each piece increases or decreases the evaluation, providing intuitive insights into the strategic role each element plays in the position.

4. Thematic Examples

We show with thematic examples some positions in which SHAP can provide a good assessment of the pieces in the position, and some pitfalls showing limitations of such an approach. We use Stockfish 17.1 as the main engine, with a 5-second limit to evaluate the starting position, and a 0.1-second limit to evaluate the perturbations. We adopt the SHAP SamplingExplainer [5], evaluating a maximum of 10000 perturbations per board.

Self-blocking Pawn. One particularly compelling use case of the proposed explanations is their ability to highlight elements that are counterproductive to one's own position. In the position shown in Figure 3, Black has a completely winning advantage. Remarkably, the white pawn on **f4**, rather than supporting White's position, actually favors Black. In certain variations, the absence of this pawn would enable White to deliver checkmate, but its presence obstructs such tactical opportunities. This example illustrates how seemingly minor details can carry significant tactical weight. Even if such motifs are not immediately exploitable in the current game, recognizing them fosters deeper understanding and enhances pattern recognition.

Bishop vs Knight Endgame. In the endgame puzzle in Figure 4, White uses the bishop's long range to outmaneuver the knight, shifting play from one side to the other until the knight falls behind: **1 ♗b6 ♘b7 2 ♗d4 ♘d6 3 ♗g7 ♘f7 4 ♗c3**. White switches wings repeatedly, and the knight cannot keep

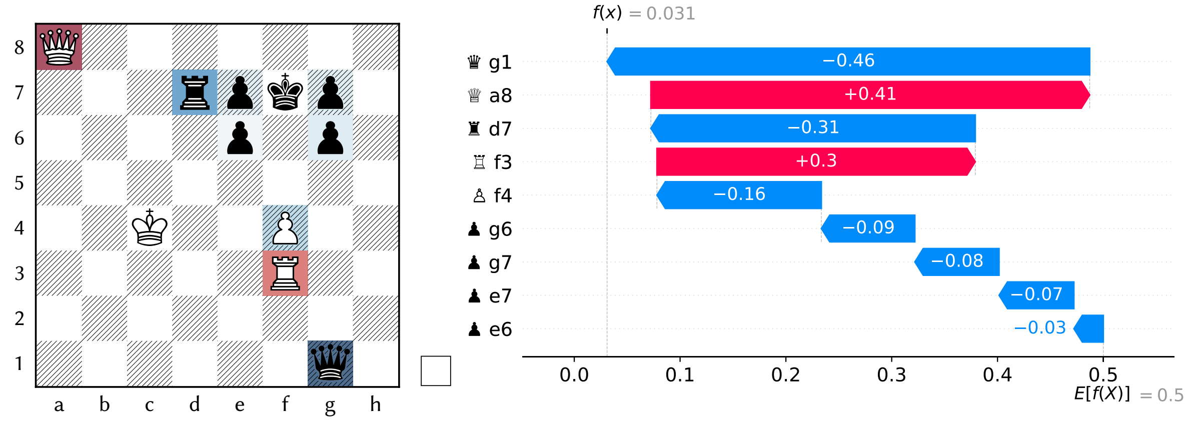


Figure 3: Self-blocking pawn. The position is completely winning for Black. Interestingly, the white pawn on **f4** aids Black's position: without it, White would mate in several perturbations of the board.

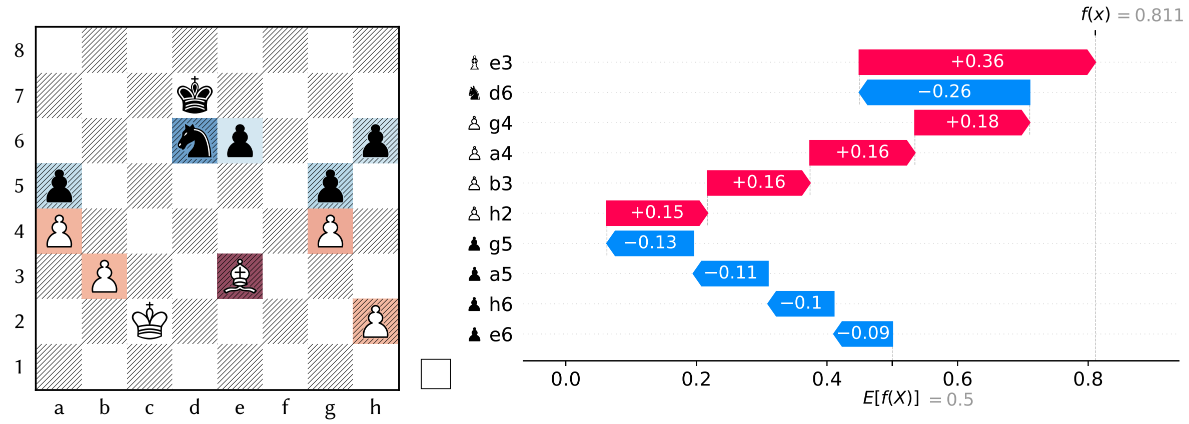


Figure 4: Bishop vs Knight Endgame. The bishop easily outmaneuvers the knight by switching wings, confirming its superiority in open endgames.

pace; in the end, Black's a-pawn is lost. SHAP correctly attributes greater importance to the bishop, which is generally stronger in endgames.

Good Knight, Bad Bishop. Contrary to the previous example, in the position shown in Figure 5, taken from the game Melkumyan–Gabuzyan, ARM-ch rapid, Yerevan 2016, the strategic inferiority of the bishop compared to the knight becomes evident. The critical pawn thrust **133...f4!** initiates a dynamic transformation of the position, exposing the bishop's limited mobility and poor coordination with its own pawns. The resulting structure confines the bishop to passive squares, severely reducing its influence. In contrast, the knight demonstrates superior maneuverability, exploiting both color complexes and coordinating effectively with the advancing pawns. This imbalance ultimately leads to a decisive advantage for Black, highlighting the practical superiority of the knight in positions where the bishop is obstructed by its own pawn formation. This superiority is correctly highlighted by SHAP and confirmed through the concrete evaluation of the resulting endgame scenario.

Trapped Rook. Figure 6 illustrates an example of how positional constraints can drastically affect the evaluation of seemingly equivalent pieces. In this position, the black rook on **b5** contributes only 0.32 towards Black's advantage, while the white rook on **b2** is valued at 0.56. Despite both being rooks, the discrepancy arises from their differing mobility impact. The black rook is severely restricted in its movement due to surrounding pawns and limited open files, reducing its tactical and positional

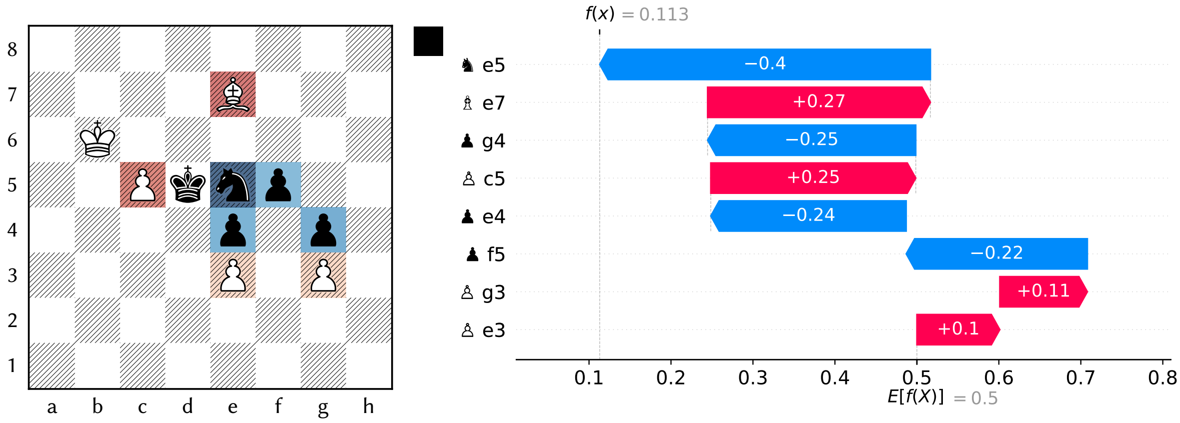


Figure 5: Good Knight, Bad Bishop. The knight dominates the board due to its flexibility and harmony with the pawn structure, while the bishop is severely restricted by its own pawns.

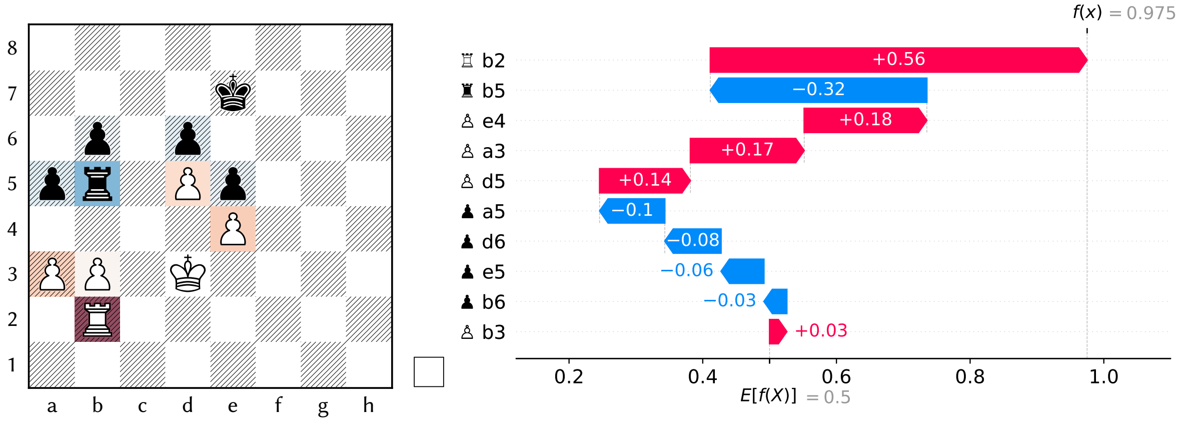


Figure 6: Trapped rook. The SHAP explanation highlights how the white rook is much more valuable than the black one.

influence. In contrast, the white rook enjoys greater freedom and exerts pressure along key files.

Pins. Explanations can also assist in identifying pinned pieces and the pieces responsible for the pin. In Figure 7, the black queen is evaluated as the worst piece, while the white bishop is ranked as the second best. This information can guide a novice player to recognize that the black queen is pinned by the white bishop, an extremely important piece in this configuration. Moreover, the similar evaluations of the white queen and bishop suggest that the white queen is not as valuable as it should be, as it is also pinned, indicating that either the bishop or the king is probably the piece to move. In this case, the best continuation is given by **1 ♔h8 ♕×f3 2 ♕c8#**.

Non-contributing Piece. Explanations can be used in chess puzzles to quickly identify pieces that contribute less to the position. For example, in Figure 8, the white knight plays a minimal role in White's winning strategy, allowing the player to focus on the more critical pieces. In this case, the quickest mate is **1 ♕b4 ♖×a7 2 ♖c8#**.

Comparing Engines. Explanations can also be employed to compare different engine evaluations. In Figure 9, we examine the assessments made by Stockfish and Leela Zero (v0.31.2) on a critical position from the third game between Stockfish and AlphaZero (2018). While both engines agree that White stands better, they diverge significantly in their evaluation of the relative importance of individual pieces.

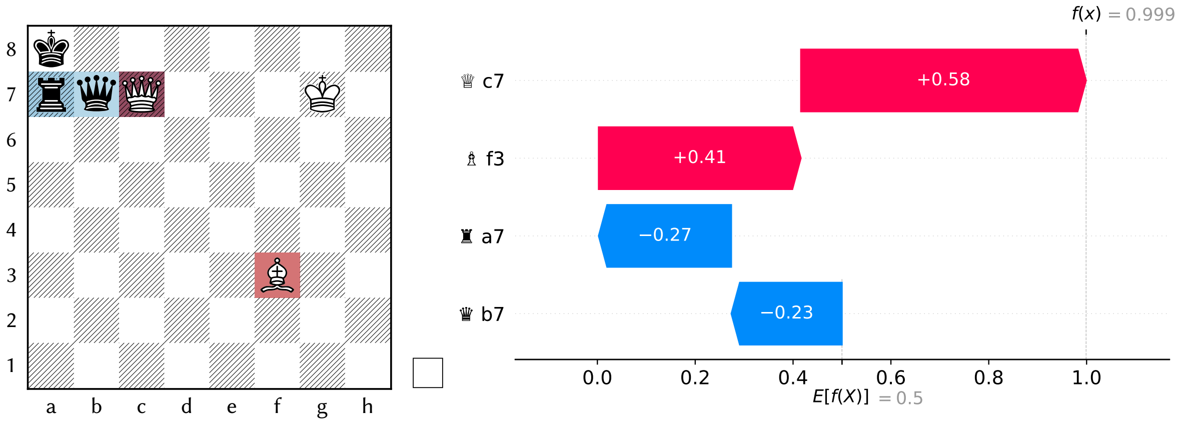


Figure 7: Pins. The high evaluation of the white bishop can help in identifying the two pins in this position.

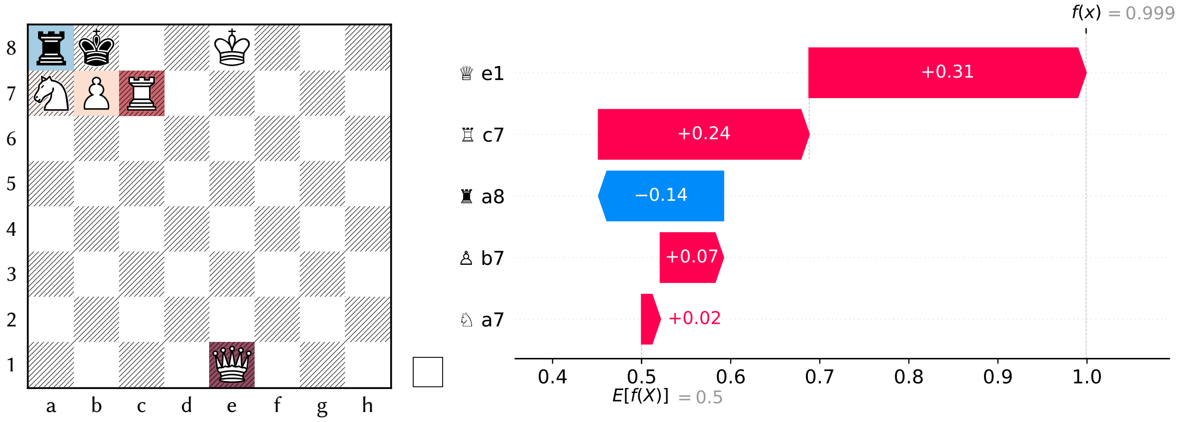
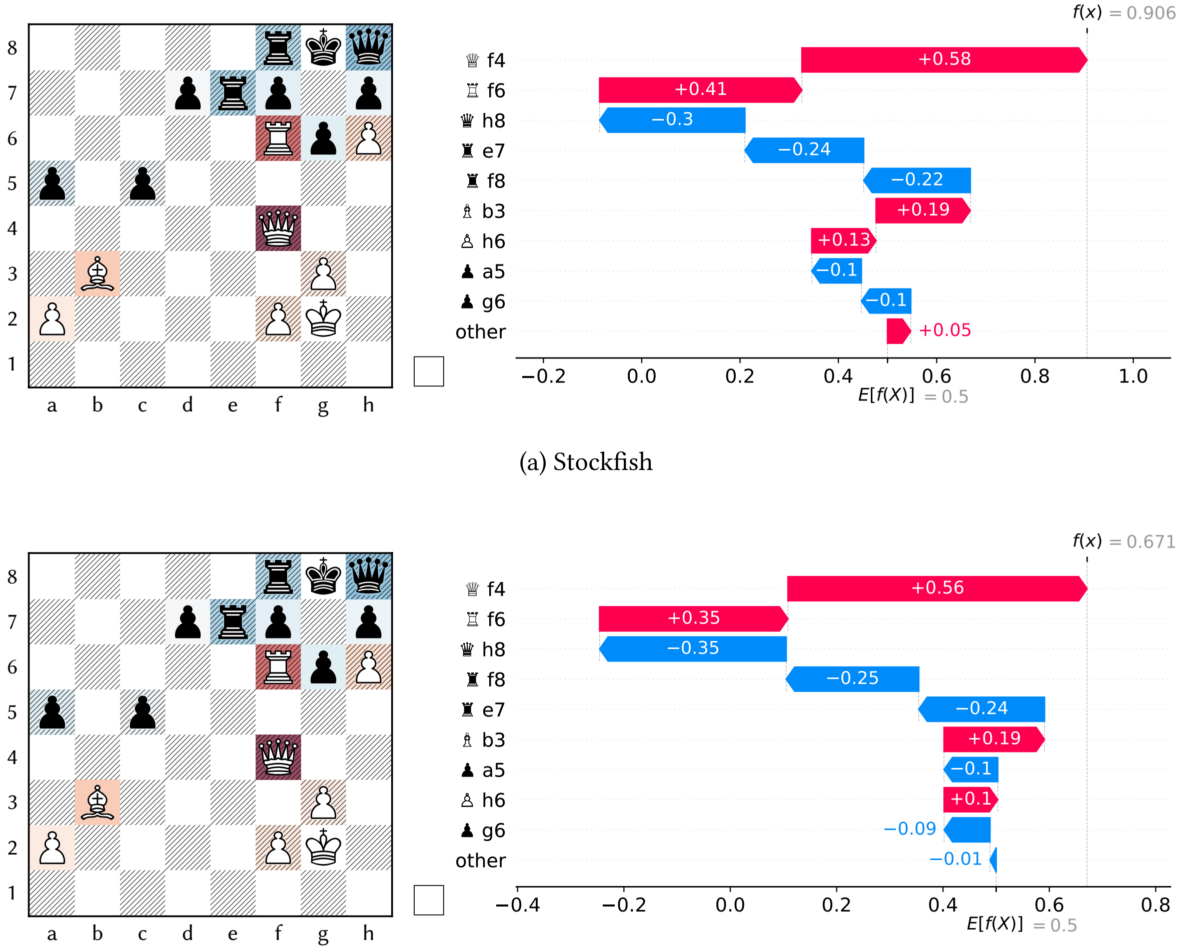


Figure 8: Non-contributing Piece. When analyzing a chess puzzle, it could be helpful for finding a solution to ignore pieces that do not contribute significantly to the position, in this case, the white knight.

The greatest disagreement concerns the white rook on **f6** and the black queen on **h8**: Stockfish assigns significantly higher value to the rook, whereas Leela regards the two pieces as similarly important. Such evaluations can be useful for understanding the strategic priorities and heuristics of different engines, and for uncovering subtleties that might otherwise be overlooked.

4.1. Pitfalls

Despite the interpretability benefits of SHAP-based attributions in the chess domain, several caveats must be considered to avoid misinterpreting the resulting scores. A central limitation is that SHAP values represent *average marginal contributions* over all possible subsets (coalitions) of pieces. This means that a piece's attribution reflects its contribution in the context of many different board configurations, not just the one currently under analysis. Consequently, the attribution assigned to a piece does not directly correspond to the change in evaluation that would result from its removal. In other words, SHAP explanations capture statistical relevance across hypothetical perturbations rather than causal or deterministic influence in the original position. Another key limitation is that the explanations are not guaranteed to be actionable. Many of the perturbed positions used in the SHAP computation may not be legally reachable from the original game state, due to the combinatorial nature of piece ablations and the disregard for move history. Therefore, while the model assigns attribution scores based on its evaluation function, these scores do not imply that a given piece can or should be moved or removed to obtain a particular outcome. Instead, the attributions are best interpreted as pedagogical tools: they



(a) Stockfish

(b) Leela

Figure 9: Comparing Engines. The same position can be evaluated differently by various engines. Explanations can help identify the pieces whose evaluations differ the most.

offer insights into the strategic value assigned to each piece by the engine, helping players develop intuition and enhance their positional understanding.

King's Importance. One inherent limitation of the proposed approach is that the king's strength cannot be directly evaluated, as engines are unable to assess positions in which the king is absent. In the critical position taken from Denis Khismatullin vs. Pavel Eljanov, European Individual Championship Jerusalem ISR (2005), shown in Figure 10, the best move is **44 ♔g1!**. SHAP is unable to attribute importance to it, as its removal invalidates the position from the engine's perspective. Nevertheless, SHAP highlights other strategically significant features, such as the passed black pawns on **d3** which, if absent, leads to a white mate in several perturbations of the board.

High Number of Pieces. When the board contains too many pieces, evaluating all possible combinations becomes computationally infeasible. Consequently, the estimated marginal contributions may overlook relevant configurations. Since the number of potential ablations grows exponentially with the number of pieces (2^n), brute-force methods quickly become impractical. In practice, a manageable upper limit is around 14 pieces (excluding kings), which requires approximately 5 minutes of computation[2]. For highly populated positions, targeted optimizations would be required to intelligently reduce the search space, ensuring that the resulting explanations remain accurate and reliable.

[2] Apple M1 Pro, 32GB of RAM, Sonoma OS.

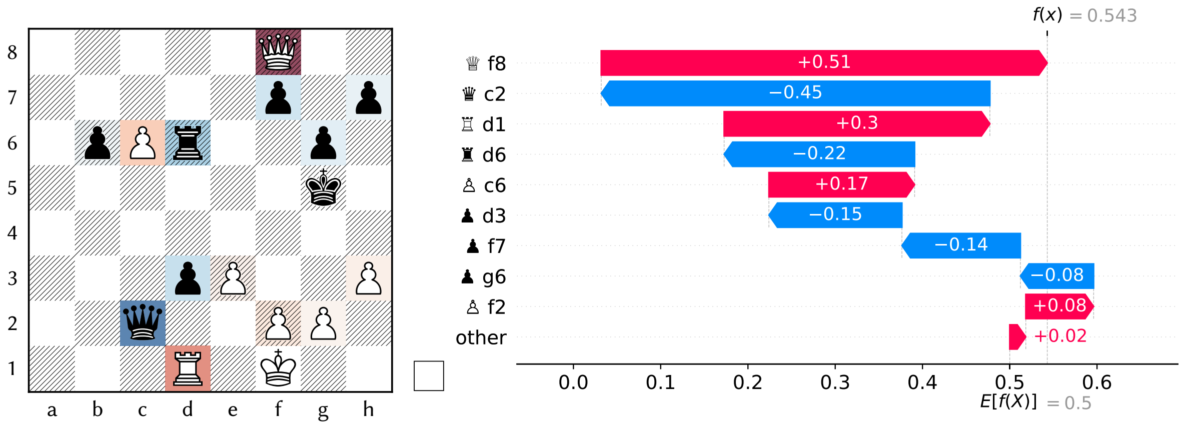


Figure 10: Pitfall: King's Importance. The best move in this position is **44 ♔g1**, however, an inherent limitation of the approach is that it is not able to assign importance to the king.

In summary, while SHAP provides a powerful framework for interpreting model predictions, its application in chess evaluation comes with inherent limitations. These explanations should be viewed as heuristic insights rather than prescriptive guides for decision-making.

5. Discussion and Conclusion

We have presented a method for attributing a chess engine's evaluation to individual pieces on the board by adapting SHAP, a principled model-agnostic interpretability framework, to the structured domain of chess. Our approach frames the engine as a probabilistic evaluator and computes piecewise contributions through systematic ablations, yielding additive and locally faithful explanations. The resulting attributions not only align with established pedagogical insights but also provide a rigorous foundation for analyzing the strategic and tactical value of each piece in a given position.

A central motivation of this work lies in its didactic potential. Our explanations may provide a bridge between human teaching practices and modern chess engines. While this paper does not yet provide a controlled study with human learners, future evaluations in instructional settings could assess whether such explanations improve chess understanding and skill acquisition. Beyond game analysis, this framework also opens promising avenues. One concrete direction suggested by our results is the evaluation of chess puzzles. Since puzzle quality is often judged by elegance, difficulty, and the contribution of specific pieces to the solution, piecewise attributions could provide quantitative support for puzzle generation and ranking. However, despite its interpretability benefits, future work is needed to scale the approach to more complex positions. One promising direction is the incorporation of hierarchical or structured coalitions of pieces, which could reduce the exponential search space without sacrificing fidelity. Similarly, sampling strategies guided by strategic priors, rather than uniform random ablations, may offer further efficiency gains.

Beyond chess, this methodology could generalize to other domains in which models evaluate structured states based on multiple interacting components, such as turn-based strategy settings, as well as non-game domains like multi-agent simulations or complex decision environments. These domains could benefit from similar forms of structured, per-component explanation. For example, in a turn-based strategy game, one could ablate individual units or resources to quantify their marginal impact on the probability of victory, highlighting which assets or tactical elements are most decisive. Likewise, in a multi-agent simulation, selectively removing or altering a single agent's behavior could reveal how cooperation, competition, or coordination among agents contributes to emergent outcomes. By bridging model-agnostic interpretability with combinatorial structure, our work contributes a reusable blueprint for localized attribution in settings where understanding why a model prefers a particular configuration is just as important as the evaluation itself.

Acknowledgments

This work has been partially supported by the Italian Project Fondo Italiano per la Scienza FIS00001966 "MIMOSA", by the PRIN 2022 framework project "PIANO" (Personalized Interventions Against Online Toxicity) under CUP B53D23013290006, by the European Community Horizon 2020 programme under the funding schemes ERC-2018-ADG G.A. 834756 "XAI", by the European Commission under the NextGeneration EU programme – National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) Project: "SoBigData.it – Strengthening the Italian RI for Social Mining and Big Data Analytics" – Prot. IR0000013 – Av. n. 3264 del 28/12/2021, M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR" - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", and "FINDHR" that has received funding from the European Union's Horizon Europe research and innovation program under G.A. 101070212.

Declaration on Generative AI

Grammar and spelling check.

References

[1] S. Mücke, L. Pfahler, Check mate: A sanity check for trustworthy ai., in: LWDA, 2022, pp. 91–103.

[2] P. Hammersborg, I. Strümke, Information based explanation methods for deep learning agents—with applications on large open-source chess models, Scientific Reports 14 (2024) 20174.

[3] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, S. Rinzivillo, Benchmarking and survey of explanation methods for black box models, Data Mining and Knowledge Discovery 37 (2023) 1719–1778.

[4] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[5] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[6] B. H. Van der Velden, H. J. Kuijf, K. G. Gilhuijs, M. A. Viergever, Explainable artificial intelligence (xai) in deep learning-based medical image analysis, Medical Image Analysis 79 (2022) 102470.

[7] M. Poggioli, F. Spinnato, R. Guidotti, Text to time series representations: Towards interpretable predictive models, in: International Conference on Discovery Science, Springer, 2023, pp. 230–245.

[8] F. Spinnato, R. Guidotti, A. Monreale, M. Nanni, Fast, interpretable, and deterministic time series classification with a bag-of-receptive-fields, IEEE Access 12 (2024) 137893–137912. doi:10.1109/ACCESS.2024.3464743.

[9] D. Płudowski, F. Spinnato, P. Wilczyński, K. Kotowski, E. V. Ntagiou, R. Guidotti, P. Biecek, Mascots: Model-agnostic symbolic counterfactual explanations for time series, in: Machine Learning and Knowledge Discovery in Databases. Research Track, Springer Nature Switzerland, Cham, 2026, pp. 94–112.

[10] T. Hulsen, Explainable artificial intelligence (xai): concepts and challenges in healthcare, Ai 4 (2023) 652–666.

[11] F. Spinnato, R. Guidotti, M. Nanni, D. Maccagnola, G. Paciello, A. B. Farina, Explaining crash predictions on multivariate time series data, in: P. Poncelet, D. Ienco (Eds.), Discovery Science - 25th International Conference, DS 2022, Montpellier, France, October 10-12, 2022, Proceedings, volume 13601 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 556–566. doi:10.1007/978-3-031-18840-4_39.

[12] M. Bianchi, F. Spinnato, R. Guidotti, D. Maccagnola, A. Bencini Farina, Multivariate asynchronous shapelets for imbalanced car crash predictions, in: Discovery Science, Springer Nature Switzerland, Cham, 2025, pp. 150–166. doi:10.1007/978-3-031-78977-9_10.

[13] A. Pálsson, Y. Björnsson, Unveiling Concepts Learned by a World-Class Chess-Playing Agent, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, ????, pp. 4864–4872. URL: https://www.ijcai.org/proceedings/2023/541. doi:10.24963/ijcai.2023/541.

[14] J. Czech, J. Blüml, K. Kersting, H. Steingrimsson, Representation matters for mastering chess: Improved feature representation in alphazero outperforms switching to transformers, in: ECAI 2024, IOS Press, 2024, pp. 2378–2385.

[15] J. Capablanca, Chess Fundamentals, Library of Alexandria, Library of Alexandria, 1921. URL: https://books.google.it/books?id=jfdq5oixkgUC.

[16] M. Dvoretsky, K. Mueller, Dvoretsky's Analytical Manual, Russell Enterprises, Incorporated, 2023. URL: https://books.google.it/books?id=d_XDEAAAQBAJ.

[17] J. de la Villa, 50 Mistakes You Should Know: Valuable Lessons for Every Chess Player, 1 ed., New in Chess, Alkmaar, Netherlands, 2024.

[18] A. Gupta, S. Maharaj, N. Polson, V. Sokolov, The Value of Chess Squares 25 (????) 1374. URL: http://arxiv.org/abs/2307.05330. doi:10.3390/e25101374. arXiv:2307.05330.

[19] N. Puri, S. Verma, P. Gupta, D. Kayastha, S. Deshmukh, B. Krishnamurthy, S. Singh, Explain your move: Understanding agent actions using specific and relevant feature attribution, arXiv preprint arXiv:1912.12191 (2019).

[20] R. Kaushikan, W. Park, Effects of material advantage and space advantage on the komodo and stockfish chess engines, Journal of Emerging Investigators (2024). doi:10.59720/23-131.