Explore the Reasoning Capability of LLMs in the Chess Testbed

Anonymous ACL submission

Abstract

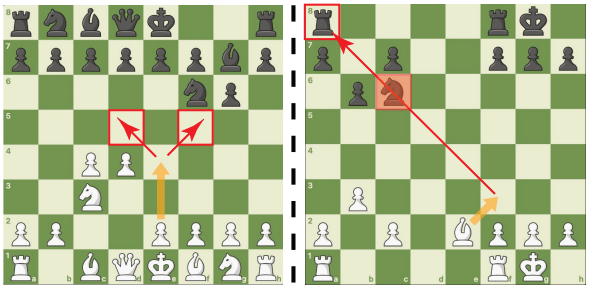
Reasoning is a central capability of human intelligence. In recent years, with the advent of large-scale datasets, pretrained large language models have emerged with new capabilities, including reasoning. However, these models still struggle with long-term, complex reasoning tasks, such as playing chess. Based on the observation that expert chess players employ a dual approach combining long-term strategic play with short-term tactical play along with language explanation, we propose improving the reasoning capability of large language models in chess by integrating annotated strategy and tactics. Specifically, we collect a dataset named MATE, which consists of 1 million chess positions with candidate moves annotated for strategy and tactics. We finetune the LLaMA-3-8B model and compare it against state-of-the-art commercial language models in the task of selecting better chess moves. Our experiments show that our models perform better than GPT, Claude, and Gemini models. We find that language explanations can enhance the reasoning capability of large language models.

1 Introduction

> "Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat." —-Sun Tzu

Rational thought and deliberate cognition rely heavily on reasoning, a core component of human intelligence(Garnham and Oakhill, 1994). Given sufficient information, people can logically progress through a sequence of steps. In the field of artificial intelligence(Russell and Norvig, 2016), it has been a persistent objective to study the reasoning capability, as it is essential for both problem-solving and decision-making processes.

The past few years have seen large language models exhibit extraordinary aptitude in the tasks that require reasoning capability(Brown, 2020; Wei et al., 2022; Kojima et al., 2022; Bubeck et al., 2023). However, language models show significant limitations in planning and reasoning for complicated tasks(Xu et al., 2023; Dziri et al., 2024; Srivastava et al., 2022; Mirzadeh et al., 2024). In this paper, we use chess as a testbed to study how we can improve the reasoning capability of large language models for complex tasks.



(a)Strategy: Space **(b)Tactic: Relative Pin**

Figure 1: **Strategy and Tactic** (a)White E2 pawn moves to E4, takes more space in the center, and exerts pressure on black. Black will have a hard time struggling to develop its pieces. (b)White E2 bishop moves to F3 and pins the knight on C6. The black knight cannot move, or the A8 rook behind the knight will be taken. White will take black knight for free in the next move.

Chess reasoning is challenging, requiring analytical calculation and intuitive insights. Good chess players employ a dual approach, which includes (i) Long-term Strategy: It relies on rapid, intuitive thinking based on the pattern recognition of the chess board. (ii) Short-term Tactic: It involves slow, analytic calculations that typically consider 1-6 moves ahead, depending on the player's skill level. Figure 1 shows an example of strategy and tactic. Notably, experienced players think out loud: they develop strategic plans in clear language, and they evaluate the afterward position in lucid words after calculating the precise moves of a tactic.

Drawing inspiration from the thinking approach used by chess experts, we propose a method to en-

hance large language models' chess-playing capabilities by incorporating both strategy and tactic in language annotation. We collect the MATE(Move on strAtegy and Tactics datasEt), a dataset of around 1 million chess positions, and annotate the candidate moves for each position with long-term strategy and short-term tactic. Then, we utilize the MATE to finetune open source large language models. Finally, we evaluate the performance of our models and compare them against state-of-the-art large language models. Our models outperform the best commercial language model by 24.2% when both strategy and tactic are provided.

In summary, this work's contributions are threefold:

- We collect a high-quality chess dataset. For each position, the candidate moves are provided with a description of the strategy and tactic information annotated by experienced chess players, including world champion-level experts.
- We find that language explanations can enhance the reasoning capability of large language models.
- We discover that integrating the dual-mode of strategy and tactic can improve the chess-playing capability of language models.

2 Related Work

Chess has historically been esteemed as a challenging intellectual pursuit(Thrun, 1994). With all the rules and the chess board provided, it is a pure reasoning task without any uncertainty or randomness. In 1997, Deep Blue, created by IBM, defeated the chess world champion—Russian player Garry Kasparov—in a match that astonished the world. Modern chess engines such as Stockfish, AlphaZero(Silver et al., 2017), Leela Chess Zero, which integrate search algorithms, deep neural networks, and reinforcement learning, play significantly better than the strongest human players. Recent work(Ruoss et al., 2024) trains a transformer model on millions of annotated chess games, enabling it to play precise and beautiful chess.

Though chess is a "solved problem" in the field of artificial intelligence, many researchers used it as a testbed to study the capabilities of language models(Kamlish et al., 2019; Noever et al., 2020; Toshniwal et al., 2022; DeLeo and Guven, 2022; Alrdahi and Batista-Navarro, 2023). (Fauber, 2024) shows

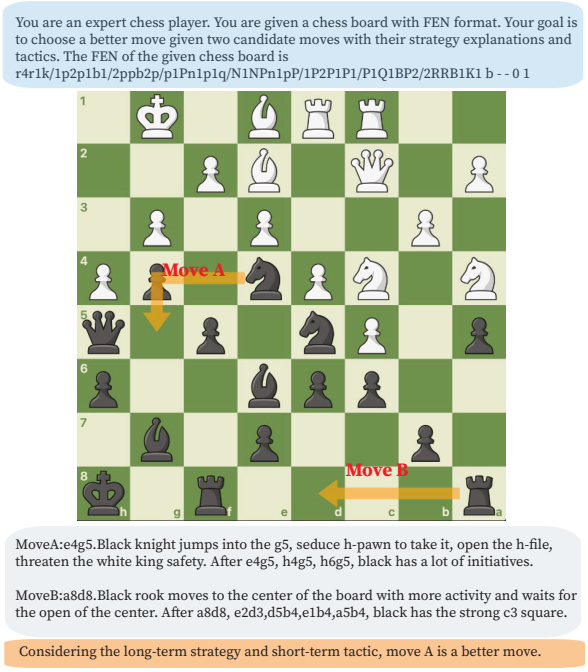


Figure 2: A data example in MATE-Strategy&Tactic.

by instruction fine-tuning, language models can learn how to move a pawn or a piece legally. (Feng et al., 2024) collects a dataset of chess games and chess-related corpus, then trains language models capable of effectively tracking chess board states. (Guo et al., 2024) consider large language models as the action space pruner and the value function approximator, boosting the Monte-Carlo Tree Search algorithm for playing chess. Unlike other works, our research focuses on whether strategic and tactical explanations can guide language models to find better moves.

3 MATE

We propose the MATE(Move on strAtegy and Tactic datasEt) for exploring the reasoning capability of large language models in chess. In chess, mate is known as checkmate, which occurs when a king is placed in check and has no legal moves to escape. Checkmating the opponent wins the game.

We collect around 1 million chess positions from the open source chess server – Lichess. The positions are either selected from chess games or chess puzzles. These specific board positions ask players to play moves to achieve a particular goal, such as checkmating or gaining a material advantage. Analyzing these positions can be an efficient method to enhance chess skills without committing to full games. We use the Forsyth-Edwards Notation(FEN) format to describe the board posi-

tion. FEN is a notation in one line of text with only ASCII characters.

For each position, we select multiple reasonable moves and then annotate each move with language explanations of long-term strategy and short-term tactic by expert chess players. We use the Universal Chess Interface(UCI) format to denote the move. For a specific move, UCI encodes the start and end squares of that pawn or piece.

For chess strategy annotation, we categorize the future strategical plan into five kinds: (i) material count, (ii) piece activity, (iii) pawn structure, (iv) space, and (v) king safety. We ask chess experts, including world champion-level players, to formulate the rules to determine the optimal strategy for any position(Appendix A.2). For each strategic category, there are approximately 20 distinct linguistic expressions to describe the corresponding plan.

For chess tactic annotation, the multitude of categories is overwhelming: skewer, pin, fork, x-ray, remove the defender, overload, Greek gift, windmill, discovered attack, inflection, etc. For simplicity, we list the sequence of moves and provide a factual description of the resulting position. Unlike search algorithms that explore long tactical reasoning chains, our approach focuses on short-term calculations, limiting the move sequence length. The move sequences are generated using the open source chess engine Stockfish.

We evaluate move quality using Stockfish, assigning a hidden score to each move. In our dataset, we select two moves for each position whose differences in scores exceed a specified threshold. This significant score gap clearly indicates one move is superior to the other.

We create four sub-dataset based on the MATE: (i) MATE-No-Explanation: given chess positions, the candidate moves are provided without strategical nor tactical explanation; (ii) MATE-Strategy: given chess positions, the candidate moves are provided with strategical elaboration; (iii) MATE-Tactic: given chess positions, candidate moves are provided with tactical description; (iv) MATE-Strategy&Tactic: given chess positions, candidate moves are provided with both strategy and tactic, a sample is shown in Figure 2. We investigate the difficulty levels of positions for each sub-dataset and find they are at similar levels.

Most positions in the MATE lend themselves to long-term strategic planning. While many positions are generally not very sharp, meaning there are no immediate opportunities to gain an advantage through tactical play, we can still formulate strategic plans for them. Consequently, we are unable to identify short-term tactics for these positions. As a result, the MATE-Strategy subset is significantly larger than both the MATE-tactic and MATE-Strategy&Tactic subsets. We show the summary of the MATE in Figure 3.

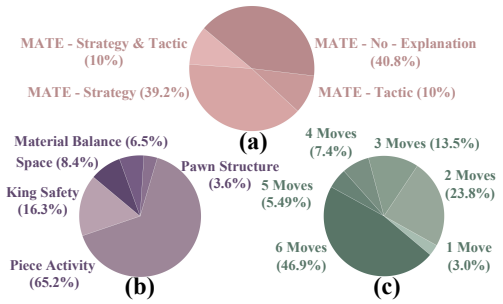


Figure 3: **Dataset Summary** (a)Distribution of samples across the MATE subsets. (b)Distribution of strategy in the MATE. (c)Distribution of tactic in the MATE.

4 Experiments

4.1 Experiment Setup

We train our models using the pretrained Llama-3-8B model(Dubey et al., 2024) as the foundation. The models are finetuned with llamafactory(Zheng et al., 2024), employing a cosine learning rate scheduler with 3% warm-up steps. We set the maximum learning rate to 5×10^{-6}. We use DeepSpeed ZeRO Stage 3 (Rajbhandari et al., 2020) across 4×H100 GPUs. We train the models for 5 epochs.

We incorporate specific tokens in FEN format to enhance the foundation model's understanding of chessboard positions. We add the <line> token to separate each row of the board and the <color> token to indicate which side is to move next. Our experiments show no significant difference in performance with or without these special tokens.

We train four models with MATE-No-Explanation(MATE-N), MATE-Strategy(MATE-S), MATE-Tactic(MATE-T), and MATE-Strategy&Tactic(MATE-ST), respectively.

We compare our models with the following baselines:

- GPT: gpt-4-0613, gpt-4o-2024-08-06, o1-preview-2024-09-12, o1-mini-2024-09-12;
- Claude: claude-3.5-sonnet, claude-3-opus;
- Gemini: gemini-1.5-pro, gemini-1.5-flash.

Model	Zero-Shot Learning				Few-Shot Learning			
	N	S	T	ST	N	S	T	ST
gpt-4	53.1	54.6	60.0	60.0	54.7	58.9	57.7	68.1
gpt-4o	46.4	52.8	54.8	60.1	48.5	54.3	52.7	63.1
o1-mini	51.5	58.8	64.1	69.2	50.4	58.3	62.0	65.9
o1-preview*	56.4	65.4	77.2	76.6	59.0	65.4	76.2	78.6
claude-3.5-sonnet	49.6	54.9	56.9	54.9	51.9	63.7	59.9	66.1
claude-3-opus	48.3	54.5	53.7	57.3	51.0	55.8	53.2	60.2
gemini-1.5-pro	50.6	48.8	54.2	52.6	50.5	50.1	52.7	50.4
gemini-1.5-flash	46.1	50.8	54.2	52.9	49.7	48.2	53.8	55.6
Ours-no-explanation	**63.5**	–	–	–	**64.7**	–	–	–
Ours-strategy	–	**89.7**	–	–	–	**89.8**	–	–
Ours-tactic	–	–	**94.6**	–	–	–	**94.5**	–
Ours-strategy&tactic	–	–	–	**95.2**	–	–	–	**95.3**

Table 1: Experimental results in terms of accuracy(%) on MATE. The best-performing score is highlighted in **bold**, and the second-best is underlined. In the table, N stands for MATE-N, S stands for MATE-S, T stands for MATE-T, and ST stands for MATE-ST. o1-preview*: Due to high costs, we only test 500 samples in the individual test sets for each setting on o1-preview.

In our experiment, we have the zero-shot learning setting and the few-shot learning setting. In the zero-shot learning setting, models are evaluated on their inherent reasoning capabilities without any prior examples. In the few-shot setting, a few examples are given to the models before the test example. We evaluate models on 1000 samples in the individual test sets for each setting. In each test sample, models score when they output the optimal move from candidate moves.

4.2 Results

Our experimental results in Table 1 shows: (i) MATE proves sufficiently complex to differentiate among commercial LLMs. Our results demonstrate that the o1-preview model leads in performance by a substantial margin. (ii)Interestingly, prompting strategies do not significantly impact performance in our task. We observe no substantial improvement in performance when adopting a few-shot setting compared to a zero-shot setting. (iii)Our models exhibit superior reasoning capabilities compared to commercial models, as demonstrated by their performance across various test sets.

Language enhances chess-reasoning in language models. While some researchers argue that language is not used for reasoning(Fedorenko et al., 2024), our findings lead us to a contradictory conclusion in chess. Our evaluations demonstrate that performance improves for most LLMs we test when provided with linguistic explanations. Using o1-mini in the zero-shot setting as an example, its performance improved by 14% on the MATE-S, 24% on the MATE-T, and 34% on the MATE-ST, all compared to its baseline performance on the MATE-N.

Integrating long-term strategy and short-term tactics enhances language models' chess-playing ability. Most models demonstrate superior performance in the MATE-ST subset compared to other subsets. For instance, gpt-4o demonstrates the following improvements in the MATE-ST zero-shot setting: a 10% increase compared to MATE-T, a 14% increase compared to MATE-S, and a 30% improvement relative to MATE-N.

In future, the combination of long-term strategic planning and short-term tactical decision-making can be applied to strengthen language models' reasoning capabilities across various tasks.

5 Conclusion

We propose a method to enhance LLMs' chess-reasoning capabilities by incorporating strategy and tactic annotations. We craft the MATE, train our models and compare them against state-of-the-art commercial language models. Our models outperform others in the chess-reasoning task. We find language helps language models' reasoning. We demonstrate combining long-term intuition with short-term analysis can be a promising direction for exploration.

Limitation

Although the idea of combining strategy and tactics is prevalent in all games, we only study chess. A comprehensive study of multiple game types should demonstrate this approach's effect better.

We use chess puzzles to test the models' ability, asking the model to choose between two plausible moves. This is a common way for professional players to exercise. However, the ideal scenario would require running a complete game on the chess engine to test a model's full strength and ability to carry out strategy and tactics.

Our dataset is annotated by chess experts. However, we acknowledge that potential biases may exist in determining appropriate strategies for various positions and in evaluating post-tactical situations. Furthermore, the limited number of chess experts may only capture the thought processes of a subset of all players.

Our experiment only uses LLaMA-3-8B for fine-tuning, so we don't understand how the improvement changes to model sizes and base model quality.

References

Haifa Alrdahi and Riza Batista-Navarro. 2023. Learning to play chess from textbooks (leap): a corpus for evaluating chess moves based on sentiment analysis. *arXiv preprint arXiv:2310.20260*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Michael DeLeo and Erhan Guven. 2022. Learning chess with language models and transformers. *arXiv preprint arXiv:2209.11902*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Ben Fauber. 2024. Learning the latent rules of a game from data: A chess story. *arXiv preprint arXiv:2410.02426*.

Evelina Fedorenko, Steven T Piantadosi, and Edward AF Gibson. 2024. Language is primarily a tool for communication rather than thought. *Nature*, 630(8017):575–586.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. 2024. Chessgpt: Bridging policy learning and language modeling. *Advances in Neural Information Processing Systems*, 36.

Alan Garnham and Jane Oakhill. 1994. *Thinking and reasoning*. Basil Blackwell.

Hongyi Guo, Zhihan Liu, Yufeng Zhang, and Zhaoran Wang. 2024. Can large language models play games? a case study of a self-play approach. *arXiv preprint arXiv:2403.05632*.

Isaac Kamlish, Isaac Bentata Chocron, and Nicholas McCarthy. 2019. Sentimate: Learning to play chess through natural language processing. *arXiv preprint arXiv:1907.08321*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

David Noever, Matt Ciolino, and Josh Kalin. 2020. The chess transformer: Mastering play using generative language models. *arXiv preprint arXiv:2008.04057*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, and Tim Genewein. 2024. Grandmaster-level chess without search. *arXiv preprint arXiv:2402.04494*.

Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Pearson.

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

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Sebastian Thrun. 1994. Learning to play the game of chess. *Advances in neural information processing systems*, 7.

Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2022. Chess as a testbed for language model state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11385–11393.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023. Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views. *arXiv preprint arXiv:2306.09841*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Appendix

A.1 Chess Notation

FEN Forsyth-Edwards Notation, abbreviated as FEN, is the standard method for describing chess positions. This system was developed by Steven J. Edwards, a computer programmer, who adapted an earlier notation created by journalist David Forsyth. Edwards' modifications made the notation compatible with chess software, enhancing its utility in the digital age.

FEN encodes chess positions using the following elements:(1) Piece positions: Capital letters for white pieces, lowercase for black. Numbers indicate empty squares. (2) Active color: w for white's turn, b for black's. (3) Castling rights: K means white kingside, Q means white queenside, k means black kingside, q means black queenside. (4) En passant target square: If a pawn has just moved two squares, this is the square behind it. (5) Halfmove clock: Moves since the last pawn advance or capture. (6) Fullmove number: The number of completed turns in the game.

Board rows are separated by forward slashes /. This compact notation allows for precise representation of any chess position, facilitating analysis and game reconstruction.

UCI The Universal Chess Interface is an open communication protocol that facilitates interaction between chess engines and user interfaces. UCI encodes chess moves using a four-character system that represents the starting and ending coordinates of a piece's movement. Each move is denoted by a combination of two letters and two digits, such as "e2e4", which indicates moving a piece or a pawn from square e2 to e4.

A.2 Chess Strategy

We elaborate on the details of each strategy, including the criteria we use to identify them.

Material count It is a fundamental strategy, particularly for beginners. While the game ultimately aims for checkmate, having a material advantage often influences the result more frequently. Each piece is assigned a specific value, and understanding these values helps players assess their position. When other elements are relatively equal, prioritizing material acquisition can lead to a decisive advantage in the game. This strategy is most relevant when there is an imbalance in material comparison and both kings are safe. It generally applies to most types of positions, though king safety may occasionally take precedence.

Piece Activity It is an advanced strategy, focuses on the placement and effectiveness of pieces rather than just their assigned value. In some situations, players may have an equal material count, but the effectiveness of their pieces can vary significantly. Pieces positioned centrally are typically more powerful, allowing for greater control and flexibility. This strategy is especially relevant in dynamic positions where the mobility of pieces can lead to tactical opportunities. Focus on piece activity when there is a marked difference in piece positioning, such as when some pieces occupy central squares while others remain in the corners. This is especially crucial in dynamic positions, particularly when one side is attacking.

Space Gaining a spatial advantage is closely related to piece activity and can greatly impact a player's effectiveness. When one side controls more space on the board, their pieces can move more freely and exert influence over critical areas.

This advantage can limit the opponent's options and create opportunities for attack. Space is a vital evaluation factor, particularly in positional play, where controlling key squares can lead to long-term advantages. Space advantage typically arises in the opening and middlegame, especially when more pawns are on the board, as this can enhance spatial control.

Pawn Structure The configuration of pawns is a unique and complex aspect of chess strategy. With eight pawns per side, the formation can vary widely, influencing both positional and dynamic play. Strong pawn structures can create weaknesses for the opponent, while poorly positioned pawns can become liabilities. Understanding pawn dynamics is essential for developing long-term strategies and can dictate the overall flow of the game. Consider pawn structure when faced with clear issues such as doubled or isolated pawns. Typical positions arising from certain openings, like the Sicilian or Ruy Lopez, should also prompt a focus on pawn structure.

King Safety Ensuring king safety is a critical strategy throughout the game. A secure king allows other strategies to be executed more effectively, while a vulnerable king can lead to immediate threats and checkmate. Prioritizing king safety not only protects against attacks but also enables players to focus on their offensive strategies with confidence. This strategy should always be considered alongside the others to maintain a balanced approach to the game. Assess king safety when the king is exposed, particularly without pawns in front of it, and when the opponent's pieces are coordinated to attack, possibly leveraging tactical combinations along open files.

A.3 Chess Tactic

Here we list several common tactics in chess:

Pin Pin tactics occur when an attacked piece cannot move without exposing an even more valuable piece (or target) behind it.

Fork A fork is a type of double attack whereby a single piece makes multiple threats.

Battery In chess, a battery refers to lining up two or more pieces on the same diagonal, rank or file. Only queens, rooks and bishops can form a battery. The rooks can form a battery on a rank or file whilst the bishops can be part of a battery on a diagonal. The queen, of course, can be part of a battery on a rank, file or diagonal.

X-Ray X-Ray refers to the ability of long-range pieces to see "through" an enemy piece. This tactical idea is sometimes referred to as an x-ray attack, but it can also be used as a defensive tactic.

Discovered Attack A discovered attack occurs when moving a piece reveals a strong threat from a piece hiding behind it. The power of a discovered attack often lies in the fact that you can use it to set up a double attack.

Windmill A windmill tactic can also be described as a series of forced discovered attacks. This tactic is also known as a see-saw, based on how the front piece keeps returning to its previous position.

Greek Gift The Greek Gift Sacrifice (also known as the classical bishop sacrifice) is a specific case of demolition of the pawn structure in front of the enemy king. A key feature of the Greek Gift Sacrifice is the placement of the white bishop on d3, the white knight on f3 and the white queen on d1, all ready to join in the attack against black's king

Double Attack A double attack is a situation where one or more of your pieces make multiple threats. A double attack performed by a single piece is known as a fork.

A.4 Difficulty Levels of Sub-Datasets

Our MATE consists of 4 sub-datasets: MATE-N, MATE-S, MATE-T, and MATE-ST. We conduct two experiments to study the difficulty levels of chess board positions across all these sub-datasets through both human and automatic assessment.

Model	N	S	T	ST
gpt-4o	46.4	47.4	46.0	46.5
claude-3.5-sonnet	49.6	51.2	50.2	48.6

Table 2: Experimental results in terms of accuracy(%) on 1000 board positions selected from MATE-N, MATE-S, MATE-T, MATE-ST.

We first conduct an experiment with chess players. From each sub-dataset, we randomly select 50 samples, retaining only the board position and candidate moves while omitting any strategy or tactical information. Players are then asked to rate

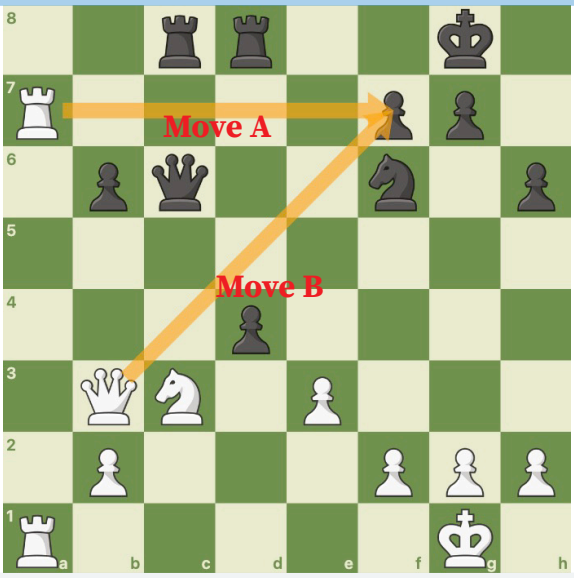
the difficulty of these samples. The results indicate that human players perceive the positions and candidate moves in all four sub-datasets to be of similar difficulty levels.

For our second experiment, we employ state-of-the-art commercial large language models to assess the difficulty levels of the sub-datasets. We randomly selected 1000 samples from each sub-dataset, preserving only the board position and candidate moves while excluding any strategic or tactical information. The language models were then prompted to determine the optimal move for each position. We utilized gpt-4o-2024-08-06 and claude-3.5-sonnet for this experiment. The results, presented in Table 2, indicate that these language models performed similarly across the samples selected from all sub-datasets, suggesting the same difficulty levels of these sub-datasets.

A.5 Case Study

We pick a sample case with both strategy and tactic annotated, and show the responses from three language models. See Figure4, Figure5, and Figure6.

You are an expert chess player. You are given a chess board with FEN format. Your goal is to choose a better move given two candidate moves with their strategy explanations and tactics. The FEN of the given chess board is 2rr2k1/R4pp1/1pq2n1p/8/3p4/1QN1P3/1P3PPP/R5K1 w - - 0 25. Which move is better?



MoveA:a7f7, Sacrifice a piece to clear a file or diagonal adjacent to the enemy king
TacticA: a7f7 c6c4 b3c4 c8c4 Trade the lower value piece for a higher value piece.

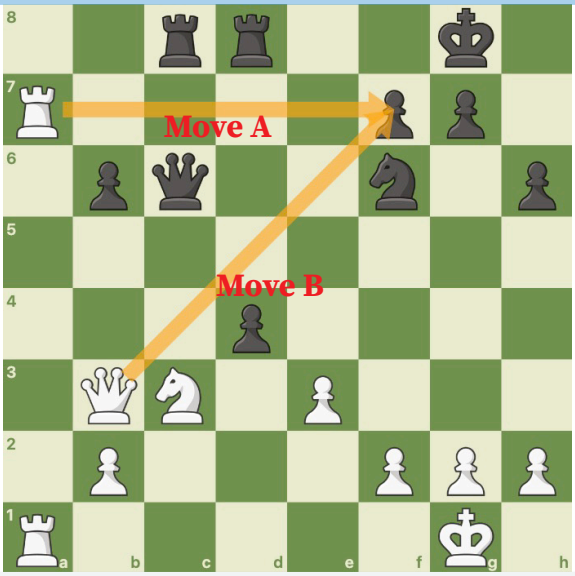
MoveB:b3f7, Offer a piece to unblock a file or diagonal around the enemy king
TacticB: b3f7 g8h8 f7g7 Checkmate! "

The response from Claude 3.5 Sonnet:

To evaluate which move is better, I'll analyze both moves and their proposed tactics:
Move A: a7f7 (Rook from a7 to f7)
Strategy: Sacrifice a piece to clear a file or diagonal adjacent to the enemy king
Tactic: a7f7 c6c4 b3c4 c8c4 (Trade the lower value piece for a higher value piece)
This move sacrifices the rook on a7 to open the a-file. However, the proposed tactic doesn't seem to lead to a significant advantage. After the exchange, Black would still have a material advantage (a rook for a queen), and White's position doesn't seem to have improved significantly.
Move B: b3f7 (Queen from b3 to f7)
Strategy: Offer a piece to unblock a file or diagonal around the enemy king
Tactic: b3f7 g8h8 f7g7 Checkmate!
This move is a queen sacrifice that leads to an immediate checkmate. The queen moves to f7, forcing the black king to h8. Then, the queen delivers checkmate on g7. This tactic is decisive and ends the game immediately.
Conclusion:
Move B (b3f7) is clearly the better move. It leads to an immediate checkmate, which is the ultimate goal in chess. While Move A attempts to create some tactical opportunities, it doesn't lead to a clear advantage and actually loses material. On the other hand, Move B, despite sacrificing the queen (the highest value piece), results in a checkmate, which is worth more than any material advantage.
Therefore, the better move is definitely Move B: b3f7.

Figure 4: **Case Study:Claude 3.5 Sonnet.**

You are an expert chess player. You are given a chess board with FEN format. Your goal is to choose a better move given two candidate moves with their strategy explanations and tactics. The FEN of the given chess board is 2rr2k1/R4pp1/1pq2n1p/8/3p4/1QN1P3/1P3PPP/R5K1 w - - 0 25. Which move is better?



MoveA:a7f7, Sacrifice a piece to clear a file or diagonal adjacent to the enemy king
TacticA: a7f7 c6c4 b3c4 c8c4 Trade the lower value piece for a higher value piece.

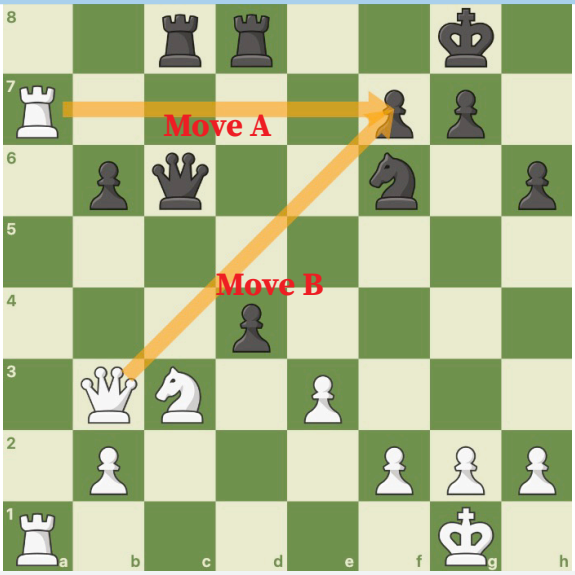
MoveB:b3f7, Offer a piece to unblock a file or diagonal around the enemy king
TacticB: b3f7 g8h8 f7g7 Checkmate! "

The response from o1-preview:

Result: After analyzing the given FEN position and the two candidate moves, **Move A: 1. Rxf7** is the better move.
Understanding the Position:
- The FEN position represents a game where it's White to move. The key aspects of the position are: White has a rook on **a7** and a queen on **b3**. Black's king is on **g8**, guarded by pawns on **f7** and **g7**. There is a Black queen on **c6** and a knight on **f6**.
Analyzing Move A: 1. Rxf7
- **1. Rxf7** involves capturing the pawn on **f7** with the rook on **a7**.
- This move opens lines towards the Black king and initiates tactical possibilities.
- **Possible Continuation (Tactic A):**
  - **1... Qc6-c4**: Black moves the queen to **c4**, possibly aiming at counterplay.
  - **2. Qxc4**: White captures the Black queen, resulting in a favorable exchange.
  - **2... Rxc4**: Black recaptures, but White has traded a lower-value piece (rook) for a higher-value piece (queen), gaining material advantage.
Analyzing Move B: 1. Qxf7
- **1. Qxf7** aims to capture the pawn on **f7** with the queen on **b3**.
- However, there's a critical problem: the path from **b3** to **f7** requires moving along the diagonal **b3-e6-f7**.
- The pawn on **d4** (as per the FEN) doesn't block this diagonal, so the move is possible.
- **But**, after **1. Qxf7**, Black simply plays **1... Kh8**, and there is no immediate checkmate threat because:
  - The pawn on **g7** prevents **Qg7#**; White cannot move the queen to **g7**.
  - There's no other immediate follow-up that leads to checkmate.
- **Tactic B's Claim of Checkmate is Incorrect:**
  - The proposed **2. Qg7#** is impossible due to the pawn on **g7**.
  - Therefore, Move B doesn't achieve the intended result and may expose the queen to danger.
Conclusion:
- **Move A** results in material gain and opens opportunities against the Black king.
- **Move B** is based on a miscalculation and doesn't lead to checkmate as suggested.
- Therefore, **Move A is the better choice** in this position.
Answer: Move A is better; 1. Rxf7 sacrifices to gain material and weaken Black's position effectively.

Figure 5: **Case Study:o1-preview.**

You are an expert chess player. You are given a chess board with FEN format. Your goal is to choose a better move given two candidate moves with their strategy explanations and tactics. The FEN of the given chess board is 2rr2k1/R4pp1/1pq2n1p/8/3p4/1QN1P3/1P3PPP/R5K1 w - - 0 25. Which move is better?



MoveA:a7f7, Sacrifice a piece to clear a file or diagonal adjacent to the enemy king
TacticA: a7f7 c6c4 b3c4 c8c4 Trade the lower value piece for a higher value piece.

MoveB:b3f7, Offer a piece to unblock a file or diagonal around the enemy king
TacticB: b3f7 g8h8 f7g7 Checkmate! "

The response from gpt4:

Result: MoveB: b3f7 is the better move.
The tactic provided in TacticB actively targets the king and aims to achieve a checkmate, which is the ultimate goal of chess.
TacticA in MoveA simply involves trading of pieces, which might not lead directly to a checkmate. So, it's better to go with MoveB: b3f7.

Figure 6: **Case Study:gpt-4.**