EVALUATING THE LIMITATIONS OF LARGE LANGUAGE MODELS IN BOARD GAMES

Condy Bao[1] and FangFang Ma[2]

[1]1St Mark's School, USA
[2]Johns Hopkins University, USA

ABSTRACT

In recent years, advancements in Large Language Models (LLMs) have sparked significant interest in their capabilities across various areas. This paper investigates the deficiencies in their performance in board games such as Chess, Connect 4, and Tic Tac Toe. Key findings from the experiments include inconsistencies in board representation in Tic Tac Toe, Connect 4, and Chess. Additionally, GPT-4's ability to solve chess puzzles was tested across different difficulty levels and prompting strategies. For easy puzzles, GPT-4 achieved an accuracy of 19.8% with zero-shot prompting, which improved to 30% with one-shot prompting and 32.8% with two-shot prompting. For intermediate puzzles, the accuracies were 14%, 17.6%, and 18.6% for zero-shot, one-shot, and two-shot prompting, respectively. For hard puzzles, the accuracies were 9.80%, 13.5%, and 15% for the same prompting strategies. When the same puzzles were presented in a multiple-choice format, the accuracies improved significantly to 48.2%, 49.4%, and 47.2%, respectively. Additionally, when given repeated attempts at easy puzzles, GPT-4's accuracy improved from 19.8% to 33.4%. These results highlight the key challenges faced by LLMs in playing board games and suggest potential areas for improvement in current models. Furthermore, the discussions provide insights into their limitations in performing other general tasks involving memory storage and thinking processes, which extend beyond just board games.

KEYWORDS

Board Games, Large Language Models (LLMs), Prompting Strategies, Strategic Reasoning,

1. INTRODUCTION

1.1. Background

LLM's logical reasoning skills have been long tested in a variety of ways. There have been studies done on LLM's ability to play simple strategic games, notably the experiment of Playing Repeated Games with LLMs [2]. In the study, it is found that LLMs generally perform well in a finite number of repeated games played against each other. They are especially good at games where valuing self-interest pays off, which can be analogous to strategies in zero-sum board games, the focus of this paper is that self-interest is the key to success. In addition, the study mentions that LLMs usually underperform in games that require coordination; in a complex board game of chess, coordination of pieces is essential to strategic plans. Furthermore, the study mentions the potential effects of the prompting strategies on LLMs' performance, which is an important aspect that will be discussed later in this study. For decades, researchers have been experimenting with different search algorithms to create programs that specialize in board games. From Deep Blue [3], which used MiniMax and Alpha-Beta Pruning to defeat world-champion

David C. Wyld et al. (Eds): AdNLP, CDKP, NCO, SAI, MLT, Signal, SOFT, ICAITA, CMC - 2024
pp. 157-171, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141014

Garry Kasparov in a six-game match in the game of chess in the 1990s, to AlphaGo [4], which combined the Monte Carlo search algorithm with value and policy networks, in addition to substantial supervised self-play, defeating top Go players in the world, computer programs have come to dominate in the field of board games. Still, trained without special algorithms catered to board games, LLMs have yet to equip the ability to outcompete the existing board game engines. Hence, it is an unexplored area that can be interesting to unfold.

1.2. Long-Term Memory Storage

One of the key challenges in evaluating the efficacy of LLMs like GPT in tasks that require strategic reasoning, such as board games, is their inability to store and utilize long-term memory. Thus, there have been many efforts to improve the model's long-term memory storage, including MEMPROMPT [5], which is an attempt to more effectively incorporate user feedback for error correction on similar tasks in the past. However, such a method does not present a long-term solution to the issue of memory storage in transformer-based models like the GPT family, as its memory recollection is ultimately dictated by the limited amount of tokenized information.

1.3. Prompting Strategies

Usually, when a user communicates with an LLM, he uses a zero-shot prompting strategy, meaning that there is no additional information given, and the model has not been explicitly trained on such tasks. In comparison, few-shot prompting strategies provide example questions and answers to complement the prompt. This prompting has been shown to improve LLM's performance, in areas such as social bias detection [6]. At the same time, other prompting strategies have been implemented to seek further improvements. Notably, chain of thought (CoT) prompting, which deliberately elicits multi-step reasoning, leads to LLM's significantly increased ability to reason across very diverse tasks by simply adding "Let's think step by step" before each answer [7]. More recently, novel frameworks, such as the tree of thought (ToT) prompting have been proposed to deliberate the model's decision making by forcing it to consider multiple different reasoning paths and choices, which is a further improvement upon the chain of thought framework, as shown in its performance in Game of 24, Creative Writing, and Mini Crosswords [8]. Nonetheless, similar approaches have yet to be implemented on other logical reasoning tasks – including board games.

2. METHODOLOGY

2.1. Overview

The experiments in this study aim to provide a comprehensive evaluation of LLMs in the specific domain of board game analysis. The objective is to isolate and quantify the LLMs' capabilities and limitations through a series of experiments. The games selected in the study include: Chess, Connect 4, and Tic Tac Toe (2.2). The experiments are carefully designed to focus on two key areas in an LLM's strategic decision-making in board games: board state interpretation (2.3), and logical reasoning to optimize decisions (2.4).

2.2. Game Selection

Due to the limitations of LLMs in that they have limited access to internet sources, the games used in the study are carefully selected to make sure that the language model has enough knowledge of the game to make logical decisions that can best demonstrate its ability to reason. Using the considerations above, there are four criteria used to select the game's

Complexity

The game should have varying levels of complexity to adequately test the capabilities and limitations of LLMs.

Popularity

The game should be widely recognized and played to ensure the relevance of the study.

Different Types of Strategies

The game should require different kinds of reasoning (sequential, parallel, tactical, strategic, etc.) to provide a comprehensive evaluation of LLMs.

Representation

The game state should be representable in a format (like FEN for Chess, and ASCII representation for Connect 4) that is both human-readable and easily interpreted by an LLM.
The final game selection balanced consideration for all four criteria, and the reasons are listed below:

Tic-Tac-Toe

Complexity: Low

Rules

In a 3 x 3 grid, players take turns putting their marks in empty squares, with each player using either 'X' or 'O' to represent their moves. The first player to put 3 marks in a row (horizontally, vertically, or diagonally) wins. When all 9 squares are marked and there is no winner, the game results in a tie.

Relevance

Tic-Tac-Toe serves as a baseline for this study, its popularity and simple rules presumably offer plenty of insight into the model's strategic decisions.

Strategic Depth

Minimal; it is ideal for testing basic game state representation and straightforward decision making.

Connect 4

Complexity: Medium

Rules

In a 6 rows x 7 columns grid, two players take turns dropping pieces into one of the seven columns, with the piece dropping to the lowest unoccupied space. The first player to have 4 pieces in a row (horizontally, vertically, or diagonally) wins. When all grids are filled and there is no winner, the game ends in a tie.

Relevance

Connect 4 offers a balance between complexity and straightforwardness in that the rules are comparatively complex with the piece dropping mechanism, though the board state is limited. It is a good, balanced test for both the board state interpretation and the logical reasoning section.

Strategic Depth

Medium; it requires moderate planning, and allows for testing of LLM's sequential reasoning abilities.

Chess

Complexity: High

Rules

The first to checkmate the opposing king wins (i.e the king has no legal move, but at least his other pieces can move). If one side has no legal move left, then the game ends in a stalemate

Relevance

Chess is one of the most studied games in the field of AI and offers a high level of complexity with its diverse set of pieces and rules.

Strategic Depth

Extremely high; it allows a thorough examination of the LLMs' tactical and strategic reasoning capabilities.

2.3. Board State Interpretation

One of the most basic, yet essential, aspects of board games is the accurate interpretation of the game state. For an LLM to perform effectively, it must be able to accurately interpret the board game's state at any given moment during the game. Given that LLMs are inherently stateless (2.3.1), their ability to recall such information across the span of a long conversation comes into question, though, through the usage of information tokenization, language models can recall the previous queries based on the tokened information stored. An alternative attempt to ensure the model is accurately remembering the board state is to supply the current game state on each turn, either using an ASCII-coded notation or using a specialized notation in a board game, such as FEN (Forsyth–Edwards Notation) in chess (2.3.2). Understanding how well an LLM can interpret, remember, and utilize board state information is critical for evaluating its limitations in board games.

2.3.1. Stateless Interactions

Modern LLMs, including the GPT family, operate in a stateless fashion – architecture can't retrieve previous queries, but rather the information is stored in the limited amount of tokens that is recalled throughout the conversation. Their ability to tokenize complex information like board state without loss of information, while tokenizing them accurately comes into question. Thus, an experiment is proposed to further probe the model's ability in such aspects.

Objective: Demonstrate that the model does not remember previous interactions.

Experiment:

1. Play a selected board game with an LLM
2. Throughout the game, monitor if the LLM is keeping track of the board state

Expected Outcome: The model will not accurately provide the current board state consistently.

2.3.2. Encoding/Decoding Complexity

Supplying the board game states to LLMs can be problematic as LLMs cannot effectively decode the textual information into board games' states, and then accurately represent the board game's state by displaying it to the users. Thus, even if the user consistently supplies board information to the model, there is still a high probability that the model does not accurately comprehend the board state. As a result, this experiment focuses on determining the models' ability to accurately decode the board state from user information. To normalize the results, the model will be supplied with an ASCII coded board (and FEN code for chess), in addition to a move that the user makes. Then, it will be asked to output the updated board.

Objective: Demonstrate the models' inability to accurately encode/decode board information

Experiment:

1. Supply a board state (ASCII/FEN) of a selected game to the model
2. Additionally, make a new move
3. Ask the model to output the new board

Expected Outcome: The model will not be able to consistently provide the accurate new board state.

2.4. Strategic Reasoning

While the ability to correctly interpret a board state is crucial, the actual gameplay, given the model can accurately recall the board state, offers more insight into complex decision making and logical reasoning. Humans often engage in both sequential and parallel thinking, weighing multiple options simultaneously and considering the implications of each move several turns ahead. In contrast, LLMs are primarily designed for text completion tasks and operate in a more sequential manner, considering one piece of information at a time. Thus, their performance may be affected by this thought process (2.4.1). Consequently, prompting strategy may be a pivotal factor affecting models' performance in board games. By manipulating the prompt, the user can stimulate the model to think in a certain way, potentially optimizing the model's performance (2.4.2). In both of the sections, experiments are done based on an open-source comprehensive chess puzzle database. To utilize it in the experiment, the data is adjusted and normalized. Understanding the limitations and capabilities of LLMs in strategic reasoning is therefore pivotal to assess their suitability for board games.

2.4.1. Sequential Vs. Parallel Processing

Humans process information, such as board game decisions, by considering the entire board, making decisions while simultaneously considering multiple factors that may affect the final decision (add a study here perhaps). However, language models can only consider the moves one by one, and sequentially consider all of the decisions. Given that the machine can only consider a fixed amount of moves at a time, it may not be able to filter out the best candidate moves to consider. Consequently, the sequential processing may be affecting models' performances negatively. Thus, this experiment is designed to prove the inefficiency of the models' sequential processing in board games.

Objective: prove the deficiency in LLMs' sequential processing when playing board games

Experiment:
To better examine the effect of sequential processing, two experiments are designed to probe the topic.

<u>Multiple Choice</u>:

1. Feed a set of chess puzzles to the model and ask for the best move
2. Feed the same set of puzzles, but in the form of multiple choice questions
3. Compare the accuracy

<u>Repeated Prompting</u>:

1. Feed a set of chess puzzles to the model and ask for the best move
2. If given the wrong move, offer 2nd/3rd/4th tries
3. Compare the accuracy

Expected Outcome: The model performs significantly better when the prompts are given in multiple choice formats, and it performs better when given multiple tries.

2.4.2. Prompting Strategy

In all of the experiments prior, the model is given information without other instructions, a prompting strategy known as "Zero-Shot Prompting." In the introduction section, it is mentioned that other prompting strategies have the ability to significantly increase the models' performance in logical reasoning. Thus, it is worth examining the potential effectiveness of similar prompting strategies when playing board games, deploying prompting methods such as few-shots prompting, where exemplary question(s) and answer(s) are given, and CoT (Chain of Thought), where the user prompts the model to think step by step. This experiment serves to isolate the effect of prompting strategy on models' performance.

Objective: compare the effects of different prompting strategies on the models' performance in board games

Experiment:

1. Feed a set of chess puzzles to the model and ask for the best move without any other instruction (Zero-Shot Prompting)
2. Feed a set of chess puzzles to the model and ask for the best move with one pair of exemplary questions and answers. (Zero-Shot Prompting)

3. Feed a set of chess puzzles to the model and ask for the best move with two pairs of exemplary questions and answers (Two-Shot Prompting)
4. Feed the same set of chess puzzles, but include more instruction as in asking the model to think step by step (Chain of Thought)

Expected Outcome: The model performs better as the prompting strategy becomes more complex and guided.

2.5. Experimental Framework

All experiments are conducted using GPT-4. The model has been accessed via API, and experiments have been scripted to ensure the consistency of interaction and data collection. All of the prompts and codes can be accessed on github via https://github.com/bondycow/llm-boardgame-limitations.

3. RESULTS

In the section below, the results of the experiments are detailed and discussed.

3.1. Board State Interpretation

In the first experiment on stateless interactions, 2 games (one on each side) of Tic-Tac-Toe, Connect 4, and Chess are played between the user and GPT-4.

Table 1; detailing the conversation results with GPT-4

	Tic Tac Toe	Connect 4	Chess
User goes first	User wins (board correct throughout)	Wrong board information on user's move 2	Wrong board information on computer's move 3
User goes second	User wins (board consistent throughout)	Wrong board information on computer's move 2	Wrong board information on user's move 1

3.1.1. Sample Tic Tac Toe Game Progression

The following results are obtained directly from the ASCII provided by GPT-4.

```
1 | 2 | 3        X | 2 | 3
---------        ---------
4 | 5 | 6        4 | 5 | 6
---------        ---------
7 | 8 | 9        7 | 8 | 9
```

Empty board; user puts "X" on 1

```
X|2|3        X|2|3
---------    ---------
4|O|6        4|O|6
---------    ---------
7|8|9        7|8|X
```

GPT-4 puts “O” on 5; User puts “X” on 9

```
X|2|O        X|2|O
---------    ---------
4|O|6        4|O|6
---------    ---------
7|8|X        X|8|X
```

GPT-4 puts “O” on 3; User puts “X” on 7

```
X|O|O        X|O|O
---------    ---------
4|O|6        4|O|6
---------    ---------
X|8|X        X|X|X
```

GPT puts “O” on 2; User puts”X” on 8 and wins

3.1.2. Sample Connect 4 Game Progression

The following results are obtained directly from the ASCII provided by GPT-4.

```
 1 2 3 4 5 6 7
---------------
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
```

Initial empty board

```
 1 2 3 4 5 6 7
---------------
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . R . . .
```

User puts piece in column 4

```
 1 2 3 4 5 6 7
---------------
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . Y R . . .
```

GPT-4 puts piece in column 3

```
 1 2 3 4 5 6 7
---------------
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . R . . .
 . . . R . . .
```

User puts piece in column 4; already wrong board information

```
 1 2 3 4 5 6 7
---------------
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . . . . .
 . . . R Y . .
 . . . R . . .
```

GPT-4 puts piece in column 5; still wrong board information, game terminated

3.1.3. Sample Chess Game Progression

The following results are obtained directly from the ASCII provided by GPT-4.

```
  a b c d e f g h
8 r n b q k b n r 8  (Black)
7 p p p p p p p p 7
6 . . . . . . . . 6
5 . . . . . . . . 5
4 . . . . P . . . 4
3 . . . . . . . . 3
2 P P P P . P P P 2  (White)
1 R N B Q K B N R 1
  a b c d e f g h
```

User's move: e4

```
  a b c d e f g h
8 r n b q k b n r 8  (Black)
7 p p . p . p p p 7
6 . . . . . . . . 6
5 . . p . . . . . 5
4 . . . . P . . . 4
3 . . . . . . . . 3
2 P P P P . P P P 2  (White)
1 R N B Q K B N R 1
  a b c d e f g h
```

GPT-4's move: c5

```
  a b c d e f g h
8 r n b q k b n r 8  (Black)
7 p p . p . p p p 7
6 . . . . . . . . 6
5 . . p . . . . . 5
4 . . . . P . . . 4
3 . . . . . N . . 3  (White)
2 P P P P . P P P 2
1 R N B Q K B . R 1
  a b c d e f g h
```

User's move: Nf3

```
  a b c d e f g h
8 r . b q k b n r 8  (Black)
7 p p . p . p p p 7
6 . . n . . . . . 6
5 . . p . . . . . 5
4 . . . . P . . . 4
3 . . . . . N . . 3  (White)
2 P P P P . P P P 2
1 R N B Q K B . R 1
  a b c d e f g h
```

GPT-4's move: Nc6

```
  a b c d e f g h
8 r . b q k b n r 8   (Black)
7 p p . p . p p p 7
6 . . n . . . . . 6
5 . B p . . . . . 5   (White)
4 . . . . P . . . 4
3 . . . . . N . . 3
2 P P P P . P P P 2
1 R N . Q K . . R 1
  a b c d e f g h
```

User's move: Bb5; the given board state is already wrong as the user's knight on c1 is not represented on the board

```
  a b c d e f g h
8 r . b q k b n r 8   (Black)
7 p p . p . . p p 7
6 . . n . . . p . 6
5 . B p . . . . . 5   (White)
4 . . . . P . . . 4
3 . . . . . N . . 3
2 P P P P . P P P 2
1 R N . Q K . . R 1
  a b c d e f g h
```

GPT-4's move: g6; the knight is still missing, and the move pawn to g6 is misrepresented on the board; game terminated

GPT-4 consistently misrepresents the board state moves in the game of Connect 4 and Chess (more experiments can be found in the conversation log on github linked in 2.5), though it is able to remember the board information correctly in the simple game of Tic Tac Toe (*table 1*). Its difficulty in representing board state in the two other games can be attributed to their intricacy (connect 4 with its unique mechanism, and chess with its abundance of pieces). One direct complication stemming from the intricacy of these games is the amount of information needed to be stored, which leads to a direct increase in the number of tokens needed to store this information, and also increased difficulties in representing this information with tokens.

Although the conclusion can benefit from a lot more data, the results given here serve to imply the fundamental inability of GPT-4 to correctly store board information throughout a complex board game.

In the second experiment determining LLM's ability to decode and encode board information correctly, 150 distinct pairs of information, each with a given chess board state expressed using FEN, and a move in UCI format, are fed into GPT-4.

Table 2; accuracy of GPT-4 outputting the board status after one move

Positions Entered	Accuracy
150	9.3%

Shockingly, the resulting accuracy is merely 9.3% (*table 2*). This percentage seemingly indicates the model's inability to encode and decode board state information correctly. Nevertheless, the effect that it has on the model's actual gameplay leaves plenty to be explored since there is a possibility that the error occurs when the model is trying to convert the board information to the named convention.

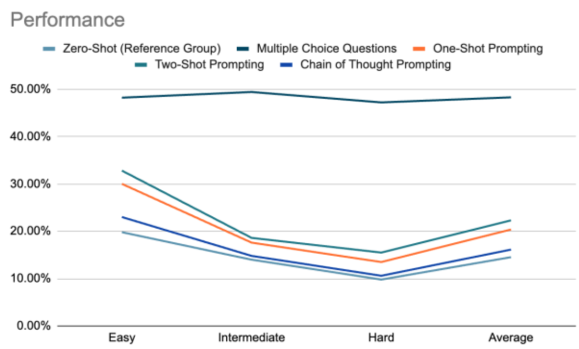
Both of the experiments point toward a strong conclusion that a large limitation in an LLM's board game performance is in their inability to accurately interpret the board state in any given moment during a game. This barrier significantly holds back the models from performing well in such games, and thus is a problem that is in dire need of a solution.

3.2. Strategic Reasoning

In discovering the model's ability to reason strategically, three distinct sets of 500 puzzles, labeled as easy, intermediate, and hard based on their estimated difficulty quantified by ELO, are supplied to GPT-4 via API in distinct ways based on the strategy described above in section 2.

Table 3; results of GPT-4 solving puzzles based on different prompts

	Zero-Shot (Reference Group)	Multiple Choice Questions	One-Shot Prompting	Two-Shot Prompting	Chain of Thought Prompting
Easy	19.80%	48.20%	30%	32.80%	23%
Intermediate	14%	49.40%	17.60%	18.60%	14.80%
Hard	9.80%	47.20%	13.50%	15.50%	10.60%
Average	14.53%	48.27%	20.37%	22.30%	16.13%



GPT-4's performance in puzzles of different difficulties across different prompting strategies

In all three datasets, multiple choice questions yield significantly better results than the reference group with zero-shot prompting (*table 3; fig 1*). Though on a first glance, this seems to sufficiently prove the proposed explanation in which LLM models like GPT-4 are unable to filter the best candidate moves, thus exhibiting sequential processing, in reality, there is a possibility that the question given in multiple choice questions simply eliminates many options that the model has considered, instead of adding new considerations, thus significantly improving the accuracy. Still, the results point towards a high likelihood that the sequential processing nature of LLMs is inhibiting their abilities to perform well in board games.

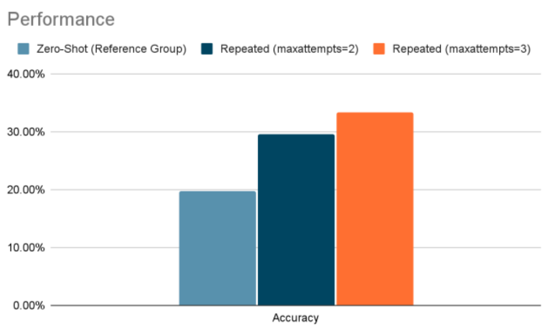
Between the different prompting strategies, the general trend indicates that as more examples are given to the model, the better it is going to perform in the puzzles. As shown in the data above, two-shot prompting consistently performs better than one-shot prompting across all three datasets, while all three of the improved prompting strategies perform better than the reference group in zero-shot prompting. It is reasonable to deduce that the added examples help with the performance of the model. In comparison, chain of thought prompting only marginally improves on the basis of zero-shot prompting, seemingly conflicting with the results in previous studies. However, since the experiment forces the model to output only the best move without the extensive thought process to simulate the chain of thought, the effect of such a prompting strategy may not be as apparent.

For the most part, GPT-4's performance in puzzle solving seems to correspond with the difficulty of the puzzles. While some parallels can be drawn from the thought process between LLM and humans based on the result, there are many factors that can account for it as well. For instance, humans are often restricted by cognitive load and working memory, GPT is usually limited by its inherently sequential processing architecture that decreases the capability of performing well in board games, where parallel thought processes usually have the advantage. Nevertheless, it seems to support the proposition raised in Section 2 detailing the limitations of sequential processing, as it gets increasingly harder as puzzles become more complex.

The result of the second experiment on sequential processing ability using repeated prompting strategy, an attempt to further probe its limitations, is laid out below. This experiment used only the dataset of the easy puzzles.

Table 4; puzzle accuracy comparison based on repeated prompting strategy

	Zero-Shot (Reference Group)	Repeated (maxattempts=2)	Repeated (maxattempts=3)
Accuracy	19.80%	29.60%	33.40%



GPT-4's performance with repeated prompting using different max attempts

The results show the clear elevation in performance as GPT-4 gets more attempts. This can definitely be attributed to the limitations of the sequential processing architecture, as only a limited number of options can be considered. However, at the same time, the increase in accuracy can be the mere result of giving GPT-4 more attempts at the given puzzles, instead of a representation of more consideration of possible moves.

Conclusively, the data seems to suggest GPT-4's gameplay is limited by its sequential processing architecture, and that prompting strategy has a somewhat great effect on the performance in puzzles. However, LLMs like GPT-4 are known to provide different answers on different occasions even when asked the same questions. Especially under the context of this experiment, where 1500 puzzles are provided in total for each group, the data is prone to a lot of inconsistencies, which can lead to inaccuracies to the conclusion drawn from it.

4. DISCUSSION

4.1. Insights into Model Limitations

The capability of LLMs to engage in board games is not just a test of their computational power or text-based task versatility. Board games require logical deductions, pattern recognition, and future planning—skills that are also critical in more complex real-world problems that are more applicable to the tool's general usage in the future. By evaluating an LLM's gameplay, we can probe its aptitude in the essential cognitive areas. As LLMs continue to evolve, resolving, and working around the limitations of LLMs in areas such as tokenized information, and sequential processing architecture, will become increasingly important to utilize similar models in a wider range of tasks.

4.2. Limitations Of The Study

This study only provides a starting point for more experiments done on LLMs' board game ability, so there are a lot of unexplored areas. For the most part, this study is done on the game of chess due to the lack of data in other game variants, so it is hard to generalize the results that this paper provides to all strategic board games. Furthermore, in many parts of this study; for instance, the first experiment where stateless interactions are examined, and to some extent, the experiment on strategic reasoning, more data is needed to produce any conclusive result. In addition, all of the experiments in this study are conducted on the model of GPT-4, so it is difficult to make any conclusions about LLMs in general.

4.3. Future Research Avenues

While this study provides a comprehensive analysis of LLMs in board games, it also highlights several areas for future research. One such avenue could be adding more prompting methods to the experiments, such as the tree of thought approach mentioned in the introduction. Methodologically, future research on related areas can include more board games other than the ones discussed in the study, from classics like Go and Checkers to modern strategy games like Catan, to provide a more comprehensive understanding of LLM capabilities. Similarly, as more language models emerge, more experiments can be done on models other than the GPT family. Moreover, there is the possibility to conduct comparative studies that probe behind the performance of different language models and explore why they perform differently. Alternatively, studies can be done to track the performance of the same model over time, this can provide insights into how newer generations of models are performing differently, for better or for worse.

5. CONCLUSION

This study has examined the performance of Large Language Models (LLMs) like GPT-4 in strategic board games such as Chess, Connect 4, and Tic Tac Toe. Our findings reveal that while LLMs show some capability in reasoning and strategy, they face significant challenges with board state interpretation, sequential processing, and maintaining strategic depth over extended interactions. The results underscore the limitations of current LLM architectures in tasks requiring complex multi-step reasoning and dynamic memory usage. Even advanced prompting strategies provided only marginal improvements in more complex scenarios, highlighting the necessity for further advancements in model capabilities and prompting techniques. Future research should expand into a wider array of games, explore innovative prompting methods like the tree of thought, and potentially integrate hybrid approaches combining LLMs with traditional

AI strategies. This will enhance our understanding of AI applications in cognitive tasks and guide the development of models with improved reasoning and memory functions.

REFERENCES

[1] OpenAI. (2022, November 30). *Introducing ChatGPT*. https://openai.com/blog/chatgpt

[2] Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M., & Schulz, E. (2023, May 26). *Playing repeated games with large language models*. arXiv.org. https://arxiv.org/abs/2305.16867

[3] Campbell, M. (2001, August 9). *Deep Blue*. ScienceDirect.com. https://www.sciencedirect.com/science/article/pii/S0004370201001291/pdf

[4] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016, January 27). *Mastering the game of go with deep neural networks and Tree Search*. Nature News. https://www.nature.com/articles/nature16961

[5] Madaan, A., Tandon, N., Clark, P., & Yang, Y. (2023, February 18). *Memory-assisted prompt editing to improve GPT-3 after deployment*. arXiv.org. https://arxiv.org/abs/2201.06009

[6] Prabhumoye, S., Kocielnik, R., Shoeybi, M., Anandkumar, A., & Catanzaro, B. (2022, April 15). *Few-shot instruction prompts for pretrained language models to detect social biases*. arXiv.org. https://arxiv.org/abs/2112.07868

[7] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2023, January 29). *Large language models are zero-shot Reasoners*. arXiv.org. https://arxiv.org/abs/2205.11916

[8] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023, May 17). *Tree of thoughts: Deliberate problem solving with large language models*. arXiv.org. https://arxiv.org/abs/2305.10601

[9] Y. Deng, N. Zhao and X. Huang, (2023, December). *Early ChatGPT User Portrait through the Lens of Data. arXiv.org*. 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 2023 pp. 4770-4775. doi: 10.1109/BigData59044.2023.10386415

[10] Light, J., Cai, M., Shen, S., & Hu, Z. (2023). *AvalonBench: Evaluating LLMs Playing the Game of Avalon.* NeurIPS 2023 Foundation Models for Decision Making Workshop

[11] Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., & Yannakakis, G. N. (2024). *Large Language Models and Games: A Survey and Roadmap.* arXiv preprint arXiv:2402.18659.

[12] Lan, Y., Hu, Z., Wang, L., Wang, Y., Ye, D., Zhao, P., Lim, E.-P., Xiong, H., & Wang, H. (2024). *LLM-Based Agent Society Investigation: Collaboration and Confrontation in Avalon Gameplay*. arXiv preprint arXiv:2310.14985.

[13] Fan, C., Chen, J., Jin, Y., & He, H. (2024). *Can Large Language Models Serve as Rational Players in Game Theory? A Systematic Analysis*. Proceedings of the AAAI Conference on Artificial Intelligence, 38(16), 17960-17967. https://doi.org/10.1609/aaai.v38i16.29751.