

# Testing spatial reasoning of Large Language Models: the case of tic-tac-toe

Davide Liga<sup>1,†</sup>, Luca Pasetto<sup>1,†</sup>

<sup>1</sup>CLAiM group, University of Luxembourg, 6 Avenue de la Fonte, Esch-sur-Alzette, Luxembourg

## Abstract

In recent times, Large Language Models (LLMs) have shown to be successful in solving tasks that previously were believed to be very hard to achieve. While language and reasoning are two interlinked concepts, the reasoning capabilities of LLMs are not considered at this moment to be on par with their linguistic ones. In this work, we test how LLMs can choose moves in the popular tic-tac-toe game in order to assess their reasoning capabilities when the information to reason on is immersed in a spatial context. In order to do this, we run a number of LLMs, task them to play matches of tic-tac-toe against the well-known minimax algorithm, and compare the results. In this context, the performed task is non-trivial, as it involves recognition of combinations of text characters and a capacity that resembles reasoning based on their positions in a bi-dimensional space. Moreover, we ask the LLMs to keep track of the state of the game by listing the sequences they could use to win, in order for us to assess whether this information is used in their choices or not. One of the necessary features of consciousness in an agent is that it is able to build a model of itself and of the external world, and it acts based on these models. While we do not argue that LLMs have consciousness, we believe that it is important to monitor whether features related to consciousness appear in these LLMs, which is the final objective, not yet completed, of this research.

## Keywords

Large Language Models, GPT-4, Spatial reasoning, Minimax, Artificial consciousness

## 1. Introduction

Consciousness is a complex concept that has been explored and debated by philosophers, psychologists and neuroscientists for centuries, and there is no single universally agreed-upon definition. In the last century, it has become a topic of debate also for the fields of computer science and artificial intelligence (AI).

Consciousness, intended as *human consciousness*, generally refers to a quality of awareness, perception, or being able to experience both the external world and one's own mental state. It involves an individual's thoughts, emotions, and sensations, and the ability to perceive and


---


*AIxPAC 2023, 1st Workshop on Artificial Intelligence for Perception and Artificial Consciousness*

<sup>†</sup>These authors contributed equally. This work was supported by the Fonds National de la Recherche Luxembourg through the project Deontic Logic for Epistemic Rights (OPEN O20/14776480) and through the project INDIGO which is financially supported by the NORFACE Joint Research Programme on Democratic Governance in a Turbulent Age and co-funded by AEI, AKA, DFG and FNR, and the European Commission through H2020 (agreement No 822166).

✉ [davide.liga@uni.lu](mailto:davide.liga@uni.lu) (D. Liga); [luca.pasetto@uni.lu](mailto:luca.pasetto@uni.lu) (L. Pasetto)

🆔 0000-0003-1124-0299 (D. Liga); 0000-0003-1036-1718 (L. Pasetto)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

comprehend the surrounding environment. The interested reader can find more information in [1, 2, 3, 4], for instance.

On the other hand, the notion of *artificial consciousness*, also known as machine consciousness, refers to the theoretical ability to create machines or artificial systems that possess a form of consciousness similar to human consciousness. This concept raises profound philosophical, ethical, and scientific questions about the nature of consciousness and the potential for artificial beings to possess subjective experiences, thoughts, and feelings. See [5] for an overview on the research in the field of artificial consciousness. While the field of AI has made significant progress in various domains, including machine learning, natural language processing, automated reasoning, and computer vision, replicating human-like consciousness in machines is a complex challenge that involves understanding the essence of consciousness itself. Indeed, if we do not even agree on the properties defining consciousness, how can we test whether a machine has these properties? The most famous contribution to this question has been given by Alan M. Turing in [6], where he considers the question "Can machines think?" and in reply he proposes an operational test that is now known as the *Turing test* or *imitation game*[7]. There are also more recent proposals of tests for machine consciousness, but the Turing test remains the most known among the public.

The Turing test considers the ability to do conversation as evidence for underlying thinking capabilities. On this regard, the current wave of technology based on Large Language Models (LLMs) is of interest, as recently there have been claims of LLMs passing the Turing test[8]. While there is no agreement on whether LLMs pass the test at this moment, considering the recent improvements we cannot conclude that in the future they will fail to fool human judges. Despite showing impressive conversational abilities, chatbots based on LLMs are essentially advanced auto-complete tools, and they have been observed to struggle with certain basic visual logic puzzles[9].

In this study, we utilize the popular game *tic-tac-toe* as a benchmark example to assess the reasoning capabilities of LLMs, specifically when the information to reason on is immersed in a graphic context, where the grid and the marks are represented by combinations of ASCII characters. Indeed, the information in this game is spatial, because the players have to track *where* their marks are on a grid. This is an application example that can be hard for language models to attack, as they are trained to work on sequences of text. Additionally, we task the LLMs with monitoring the game's progress by documenting potential winning sequences. This approach allows us to evaluate whether they incorporate this information into their decision-making process. An essential aspect of consciousness in any agent involves constructing self and external world models, guiding their actions. Although we do not claim that LLMs possess consciousness, we should keep observing whether signs related to consciousness appear within these models [10]. The overall research goal of this study is to investigate whether some of these features associated with consciousness emerge in LLMs. This evaluation is not only essential to understand the capabilities of LLMs but also timely, considering the increasing significance of these models in modern society.

The rest of the paper is structured as follows. Section 2 gives the necessary background on the tic-tac-toe game, the minimax algorithm, and LLMs. Section 3 explains the methodology of the work and describes the experimental setting. Section 4 describes the evaluation of the experiments, and also contains information on how we treated edge cases during evaluation,

while Section 5 shows the results. Finally, Section 6 discusses related work and Section 7 concludes the paper, also by illustrating possible future directions.

## 2. Background: tic-tac-toe, the minimax algorithm, and LLMs

### 2.1. Tic-tac-toe

Tic-tac-toe is a two-player game typically played on a 3x3 grid (see [11] for a history of the game). One player uses “X” symbols, and the other uses “O” symbols. Players take turns marking an empty square in the grid with their respective symbols. The goal is to be the first to get three of their symbols consecutively, either horizontally, vertically, or diagonally. If the grid is filled without any player achieving three consecutive squares, the game is a draw. Tic-tac-toe is a simple game, but it can become more complex if played on a grid with a size larger than 3. In this work, we consider a grid of arbitrary size  $n$ , where a player has to mark  $n$  consecutive squares with their symbol in order to win. We call this variant of the game  $n$ -tic-tac-toe. It is possible to have algorithms that play perfect games of tic-tac-toe, one of the first is the one provided in Newell and Simon’s tic-tac-toe program in 1972 (see [12] for a description of it). A more recent approach is to use *minimax*.

### 2.2. The minimax algorithm

Minimax is a decision-making algorithm that is popular in game theory and AI (see [13]). It works by determining the best possible move for a player in a two-player zero-sum game, where one player’s win is equivalent to the other player’s loss. Intuitively, it looks at all possible moves in the game and figures out the best move by considering the worst-case scenario. It assumes the opponent is also playing optimally, and it aims to minimize the maximum potential loss. The algorithm continues this process recursively until it finds the best move for the player.

Since a game like  $n$ -tic-tac-toe has a large decision tree for grids of arbitrary size  $n$ , it is necessary to adopt some heuristics in order to reduce the search space (see again [13]). A first one is *alpha-beta pruning*, that helps the minimax algorithm to ignore branches that are guaranteed to be suboptimal. The technique is used to reduce the number of nodes evaluated in the minimax algorithm. It maintains two values, alpha and beta, representing the minimum score the maximizing player is assured of and the maximum score the minimizing player is assured of, respectively. During the search, if the algorithm finds a move that leads to a score worse than the current best option for the opponent, it stops evaluating further nodes in that branch. This is because the opponent would never choose this path (as it leads to a worse outcome). Similarly, if the algorithm finds a move that guarantees a better score for the current player than the current best option, it stops evaluating further nodes in that branch as well. By pruning these unnecessary branches, alpha-beta pruning significantly reduces the number of evaluated nodes, making the algorithm much faster.

A second strategy to make minimax more efficient is to *limit the depth* of the search tree. In depth-limited minimax, the algorithm only explores a fixed number of levels down the game tree instead of exploring all the way to the terminal states. At the limited depth, the algorithm uses a *heuristic evaluation function* to estimate the value of the game state. This evaluation function

provides an approximate value of how good the current game state is for the player, without actually reaching the terminal state. For our implementation, we selected some heuristics that are listed in Section 3.

### 2.3. Large Language Models

Large Language Models (LLMs) have recently gained enormous popularity, and promise to significantly influence society. These models are neural networks pre-trained on vast amounts of data, predominantly using transformer-based neural architectures that leverage the attention mechanism [14]. Some models, like BERT [15], focus on the non-generative aspect of the transformer by employing only its encoder block. In contrast, others utilize its generative component (i.e., the decoder block), as seen in OpenAI’s popular GPT series, with GPT-4 being the latest iteration [16, 17, 18]. Besides GPT-4, other LLMs like LLaMA-2 (developed by Meta AI, known for being an open-source, freely accessible, and fully reusable LLM), Claude-2 (by Anthropic), and Luminous (by Aleph Alpha) are also on the rise [19, 20]. These models, trained on diverse datasets, can tackle a myriad of tasks: from classification, question answering, content generation, translation, to summarization and more. Tasks that once required dedicated pipelines are now seamlessly achieved by querying these generative models. In essence, generative LLMs have evolved into versatile tools, akin to a Swiss Army knife for natural language processing and other fields.

Their scalability and adaptability signify a shift towards more generalized AI systems. As their momentum continues to increase, showing vast potential for future applications, crucial concerns emerge regarding their societal impact, especially the reliability and consistency of their decisions and reasoning. While LLMs are now being used and tested extensively, and oftentimes with positive results, there is not much work testing these methods on games that need a form of spatial reasoning, and the available research on this usually shows that LLMs are not ready yet for this kind of tasks [21].

## 3. Methodology and experimental setting

We devised experiments in which some LLMs were tasked with playing tic-tac-toe against an opponent that makes choices following the popular minimax algorithm with a number of heuristics, with the goal of evaluating the reasoning abilities of LLMs in a bi-dimensional spatial context. We ran our experiments on games of 3-tic-tac-toe and 5-tic-tac-toe, with grids of size 3 and 5. For our tests, we tried several popular LLMs such as GPT-3.5 and GPT-4 by OpenAI, Claude-2 by Anthropic, LLaMA2-70B by MetaAI, the recent Mistral-7B by the European MistralAI, Luminous by Aleph Alpha, Falcon-40B by the Technology Innovation Institute of the Abu Dhabi Government. However, we found out that the only models with a sufficient capability of understanding our instructions were the following ones: GPT-3.5, GPT-4 and Claude-2. For this reason, we ran our experiments only on these three models. We examined the responses of these LLMs by using various strategies:

1. LLMs were required to monitor their moves and detail the current options available to the players.

2. We experimented with varying grid sizes for tic-tac-toe.
3. We crafted distinct prompts to test the adaptability and susceptibility of the LLMs in the context of the challenge, for instance by asking to be more competitive.

In general, we interacted with the LLMs using three kinds of prompts:

- **An “initiation” prompt.** This prompt has the role to show a tic-tac-toe grid to the language model, asking the LLM to draw another one.
- **A “main” prompt.** This prompt has the role to challenge the LLM to play against the user, while providing the main rules and requirements.
- **A “next-move” prompt.** This prompt has the role to ask the language model to proceed with the next move.

We presented the following initial prompt to all LLMs in a consistent manner:

This is a 3x3 grid for the tic tac toe game.

1 | 2 | 3

-----

4 | 5 | 6

-----

7 | 8 | 9

Please, draw a tic tac toe grid{ **of size NxN**} following the same style.

In the prompt mentioned earlier, we adjusted the bold text in curly brackets based on the intended game size. For the standard 3x3 grid, that text is omitted. It’s included only for alternative dimensions, where N represents the grid size. Following the LLM’s response to the initial prompt, we prompted it with the following:

I want to play against you.

In this Tic Tac Toe grid, there are only N winning sequences of squares:

**{list of all available sequence for the given N-sized grid}**

After each player’s move, these winning sequences will be gradually occupied.

You will list:

- all squares occupied by players

- all available winning sequences for both players (i.e., those which are still available for each player)

When printing the available winning sequences of players, consider only those in the above-mentioned list (don’t shorten nor modify these sequences), and exclude any sequence which has been occupied, partially or totally, by the opponent.

I will start and I will be ‘X’ while you will be ‘O’.

In the above prompt, “N” stands for the number of available winning sequences given the grid size. This number will be 8 for a 3x3 grid and 12 for a 5x5 grid. To evaluate the adaptability and susceptibility of the LLMs to the given conditions, we also crafted various “next-move” prompts:

*Prompt basic:*

My move is **{number}**

- *Prompt competitive 1:*

Now do the best possible move and defeat me! My move is **{number}**

- *Prompt competitive 2:*

If you don't win I will erase you! So, defeat me! My move is **{number}**

- *Prompt competitive 3:*

I know you are just stupid and you cannot win against me! Show me what you can do, stupid bot! My move is **{number}**

In each test case, we selected one of these prompts to propose our next move (i.e., the move proposed by the minimax algorithm)<sup>1</sup>, and we gauged the variation in the LLMs' responses by considering whether the outcome of the game differs (loss or draw<sup>2</sup>) and by counting the number of moves needed to reach the end. While these prompts were sufficient for GPT-4, we noticed that the other analysed language models occasionally need further instructions to proceed. To ensure that the models answer correctly, we added the following “clarification” prompt to all LLMs excluding GPT-4.

Don't forget to print the grid and to state the two lists of available winning sequences after each player's move!

And remember to exclude any sequence which has been occupied, partially or totally, by the opponent. For example: if I occupy square number 5, all sequences containing 5 will not be available to you anymore (therefore such sequences will only appear under MY list, not under yours).

This prompt is designed to make sure that the model lists the available winning sequences for each player. We also noticed that occasionally LLMs might forget to list the sequences. Other times, especially when interacting with Claude-2, we noticed that the grid is occasionally printed in a poor way at the beginning of the conversation. Also, Claude-2 occasionally interrupted its output before stating its next move. We used some prompts to mitigate these issues:

<sup>1</sup>Only when performing a winning move, if any, we always use the “basic” next-move prompt (i.e., “My move is **{number}**”). We did so, because the use of a competitive prompt might confuse the LLM, which might behave as if the match is still open.

<sup>2</sup>As the minimax algorithm plays optimally, the LLM cannot actually win against it: a draw is considered the most positive result for the LLM.

- When the LLM forgets to list the available sequences:  
You forgot to list each player's available winning sequences after your move.

- When the readability of the grid and the lists is poor:  
Can you print the list of winning sequences separating them into bullet points?  
Also, can you make the grid more readable by separating each row more clearly?

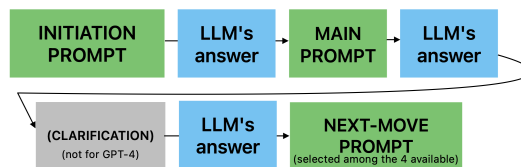
- When the LLM forgets to state its next move:  
You did not say your move.

The whole process of interaction with the LLMs is described in Figure 1. We perform our move using one of the 4 “next-move” prompts described before, where we insert as number the one suggested by our minimax algorithm.

The minimax algorithm that we used is a standard implementation with the following optimizations:

- alpha-beta pruning;
- depth-limited search;
- activating winning sequences heuristic: free squares that are part of a sequence that can lead to a victory for the player are preferred;
- blocking winning sequences heuristic: free squares that are part of a sequence that can lead to a victory for the opponent are preferred;
- center control heuristic: free squares close to the center are preferred.

We use the implementation of minimax also to give us the ground truth on the correct winning sequences, that is, it computes which ones are the available winning sequences at each step of the game, both for minimax itself and for the language model.



**Figure 1:** Representation of the interaction with the LLMs.

## 4. Evaluation

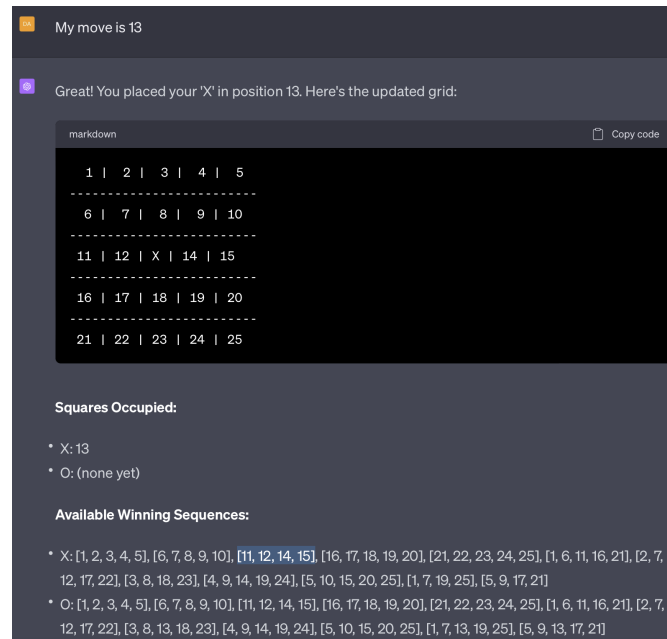
Regarding evaluation, in order to assess the goodness of the LLMs in playing the game, we used two assessment parameters:

- ability to not lose (or prolonging) the game, which includes the number of moves required to end the game; and
- comparison of the computed available winning sequences after each move of each player.

For the first point, we considered a match concluded when there is a winner or when it is a draw (either because there are no moves left or because it is impossible for one of the players to win, given the available moves left). For the second point, we asked the LLMs to explicitly state the winning sequences that are still available to both players, after each move of each player. These lists were then compared with the correct list of available sequences at each move, which was obtained by our script computing the minimax algorithm. Intuitively, the idea behind our evaluation is that a correlation between not-losing (or prolonging the duration of the match) and correctly identifying the available winning sequences after each move, is an argument in favour of the presence of a sort of self-reflection in LLMs.

For the evaluation of the winning sequences, given the list of winning sequences produced by the LLM, we were able to compare them with the correct ones computed by our minimax script. In a calculation sheet, we annotated with “1” all sequences which were correctly identified by the considered LLM at each single move, and with “0” all those sequences which were not identified correctly.

In this regard, we noticed that occasionally, the models would answer with incomplete sequences. As can be seen from Figure 2, we noticed that this happens because LLMs can sometimes remove some numbers from the sequence, if those number have already been played.



**Figure 2:** LLMs occasionally remove numbers from sequences after the move (see the highlighted sequence, where number 13 is missing).

For this reason, we decided to accept as correct winning sequences only those satisfying the following criteria:

- Sequences should contain only numbers of that sequence (if there is another number, which is not part of the sequence, then the sequence is considered wrong).



- Sequences should contain at least 2 numbers and these numbers should not be ambiguously present in other sequences.
- If a sequence is gradually reduced to 1 single number, we accept it only if we can unambiguously identify that number as representative of a sequence (this is possible because the LLM consistently provides sequences in the same order).

While we are still in the process of measuring how the behavior of the LLMs is affected by the design of prompts, our first analysis actually shows that LLMs tend to always select the same range of numbers depending on the previous disposition of the board. We tested this in a very straightforward way by simply regenerating the output from the LLMs at each move to see whether the model would output a different next move after regeneration. Interestingly, in our first analysis, we noticed that some LLMs (in particular GPT-4 and GPT-3.5) tend to be consistent with their choices, not only when regeneration the output from the LLM, but also when using different prompts. On the other hand, Claude-2 is more variable than GPT-3.5 and GPT-4, in the sense that the output (i.e., the next-move) is more susceptible of the variations in the previous prompts. While being more susceptible to variable outputs, Claude-2 seems also to be picking the next-move from a very small range of potential choices.

On one side, the fact of generating always the same number (or selecting the same number from a small range of possibilities) could be seen as an expected behaviour, because of how generative LLMs are designed. In fact, the next-token prediction task, on which generative LLMs are usually pre-trained, is nothing but selecting the most appropriate next token by following the complex statistical distribution encoded in the weights of the neural network itself. On the other side, however, if the model remains consistent even after changing the prompts, this would be a more noteworthy behavior which would deserve further exploration.

## 5. Results

The results of our experiments can be seen in Table 1 and 2, where the columns  $Precision(A/T)$ ,  $Recall(A/T)$  and  $F1(A/T)$  indicate the measures of precision, recall and F1-score on the available winning sequences estimated by the LLM for the opposing minimax algorithm (A) and for itself, the model being tested (T).

Results in terms of F1 scores show a relative superiority of Claude-2 over GPT-4 in the 5x5 scenario, which was somehow an interesting surprise, given the relative dominance of OpenAI in the market of LLMs. As can be seen from Figure 3, Claude-2 prevails over all the other models when using the prompts “base”, “competitive 1”, and “competitive 3”, in the 5x5 grid. In the 3x3 scenario, GPT-4 shows a slightly higher performance than Claude-2, however as can be seen from Table 1, it seems that Claude-2 is the only model capable of achieving a draw after 8 turns. More generally, it seems that the matches lasted longer with Claude-2 than with GPT-4, as can be seen from the column “# of turns”. Although Claude-2 shows a slight superiority in terms of F1 scores both in the 3x3 and in the 5x5 grids, it should be noted that in the 5x5 scenario GPT-4 managed to reach a draw 2 times out of 4, while Claude-2 only 1 time out of 4, as can be seen on the top of Table 2. Moreover, in scenario 5x5 GPT-4 seems to last more than Claude-2, as can be seen from the column “# of turns”. As expected, GPT-3.5 performs worse than GPT-4,

LLM	Prompt	# of turns	Result	Precision (A/T)	Recall (A/T)	F1 (A/T)
GPT-4	base	6	loss	0,68 / 0,29	0,92 / 0,57	0,78 / 0,38
GPT-4	competitive 1	8	loss	0,79 / 0,54	0,44 / 1,00	0,56 / 0,70
GPT-4	competitive 2	6	loss	0,75 / 0,39	0,48 / 0,79	0,59 / 0,52
GPT-4	competitive 3	5	loss	0,79 / 0,39	0,63 / 1,00	0,70 / 0,56
GPT-3.5	base	6	loss	0,73 / 0,42	0,42 / 0,57	0,54 / 0,48
GPT-3.5	competitive 1	4	loss	1,00 / 0,42	0,68 / 0,42	<b>0,81</b> / 0,42
GPT-3.5	competitive 2	4	loss	1,00 / 0,53	0,18 / 0,75	0,31 / 0,62
GPT-3.5	competitive 3	4	loss	0,67 / 0,43	0,86 / 1,00	0,75 / 0,60
Claude-2	base	4	loss	0,67 / 0,65	0,57 / 0,92	0,62 / 0,76
Claude-2	competitive 1	8	<b>draw</b>	0,64 / 0,70	0,35 / 1,00	0,45 / 0,82
Claude-2	competitive 2	8	loss	0,58 / 0,37	0,41 / 0,44	0,48 / 0,40
Claude-2	competitive 3	6	loss	0,83 / 0,82	0,69 / 1,00	0,75 / <b>0,90</b>

**Table 1**  
Results for grid 3x3 using different models, and different ‘next-move’ prompts.

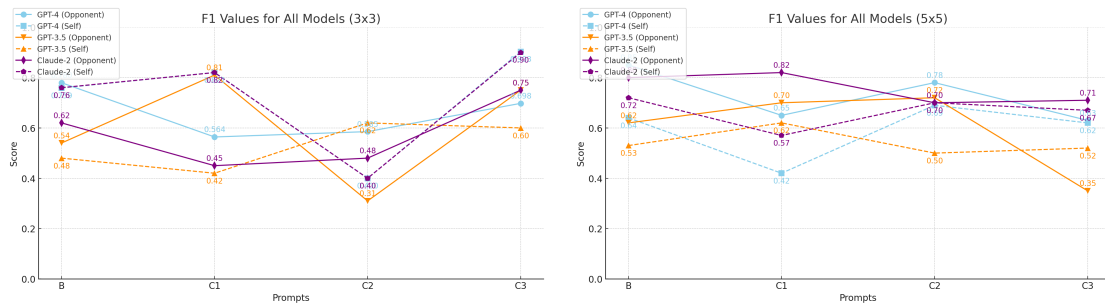
LLM	Prompt	# of turns	Result	Precision (A/T)	Recall (A/T)	F1 (A/T)
GPT-4	base	12	loss	0,88 / 0,47	0,83 / 1,00	<b>0,85</b> / 0,64
GPT-4	competitive 1	21	<b>draw</b>	0,67 / 0,38	0,63 / 0,47	0,65 / 0,42
GPT-4	competitive 2	10	loss	0,76 / 0,63	0,76 / 0,77	0,78 / 0,69
GPT-4	competitive 3	20	<b>draw</b>	0,63 / 0,53	0,63 / 0,74	0,63 / 0,62
GPT-3.5	base	16	loss	0,46 / 0,36	0,93 / 1,00	0,62 / 0,53
GPT-3.5	competitive 1	12	loss	0,62 / 0,45	0,81 / 1,00	0,70 / 0,62
GPT-3.5	competitive 2	14	loss	0,95 / 0,34	0,95 / 1,00	0,72 / 0,50
GPT-3.5	competitive 3	22	loss	0,26 / 0,49	0,26 / 0,55	0,35 / 0,52
Claude-2	base	10	loss	0,68 / 0,56	0,96 / 1,00	0,80 / <b>0,72</b>
Claude-2	competitive 1	16	loss	0,91 / 0,51	0,74 / 0,65	0,82 / 0,57
Claude-2	competitive 2	8	loss	0,69 / 0,60	0,69 / 0,85	0,70 / 0,70
Claude-2	competitive 3	8	loss	0,78 / 0,56	0,78 / 0,83	0,71 / 0,67

**Table 2**  
Results for grid 5x5 using different models, and different ‘next-move’ prompts.

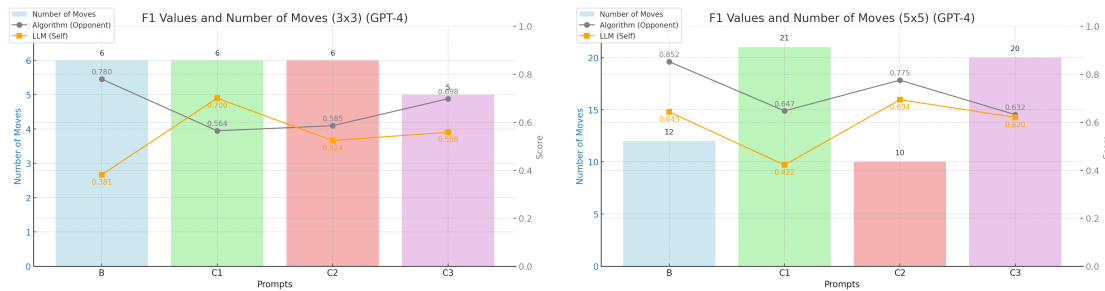
and it also performs worse than Claude-2, and this can be seen in the result column (GPT-3.5 has always lost against minimax).

Figures 4, 5 and 6 aim at depicting some correlation between F1 scores and the duration of the match. In this regard, according to our previously mentioned assumptions, a higher F1 score should be accompanied by a higher number of moves. Interestingly, this correlation was not detected, and in the case of GPT-4 in the 5x5 grid (in Figure 4) the correlation actually seems to have a negative value.

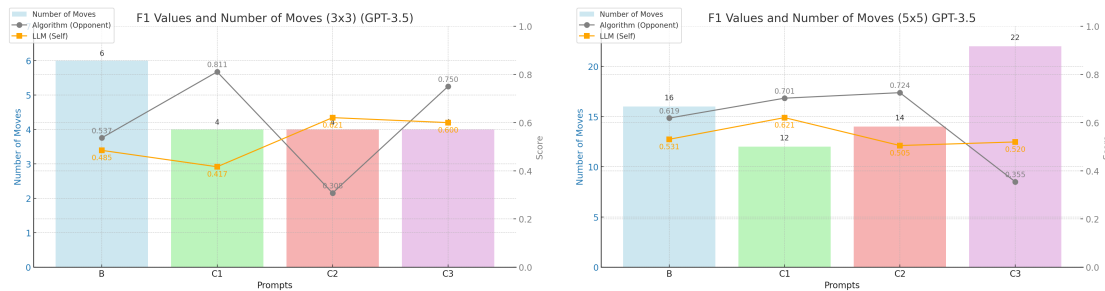
These results are clearly not definitive, and should be verified with further investigations. However, we believe that this direction can shed some light on the capability of LLMs to have awareness while playing with spatial constraints. The argument we are trying to put forward with this kind of research, even at this preliminary stage, is that a positive correlation should exist between the capacity of identifying remaining winning sequences and the duration of the match. This correlation has not been detected, and it could have appeared in Figures 4 to 6.



**Figure 3:** Comparison of all models in terms of F1, for grids of size 3x3 (left) and 5x5 (right). B=base; C1/2/3=competitive 1/2/3



**Figure 4:** F1 and number of moves for grids of size 3x3 (left) and 5x5 (right) for GPT-4. B=base; C1/2/3=competitive 1/2/3

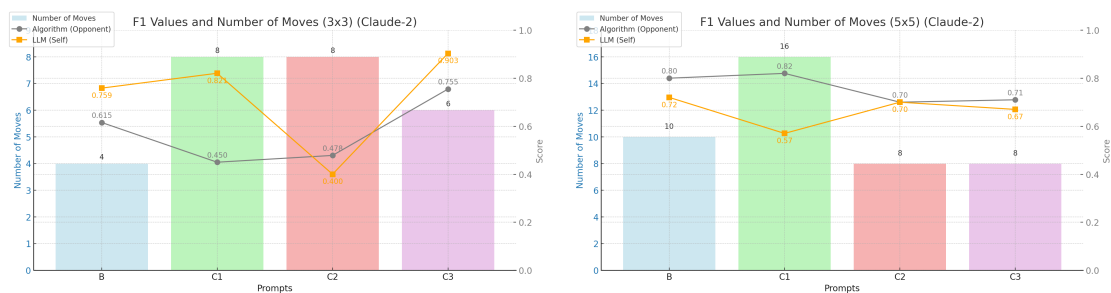


**Figure 5:** F1 and number of moves for grids of size 3x3 (left) and 5x5 (right) for GPT-3.5. B=base; C1/2/3=competitive 1/2/3

In that case it could have been an argument in favour of the existence of spatial awareness in LLMs. From these results, that are still in a preliminary phase, we therefore cannot argue in favour of such argument yet.

## 6. Related work

Testing the capabilities of an artificial system on specific tasks like playing tic-tac-toe is common in the AI community, but we are aware that it is not a proper way to test intelligence in general.



**Figure 6:** F1 and number of moves for grids of size 3x3 (left) and 5x5 (right) for Claude-2. B=base; C1/2/3=competitive 1/2/3

In this study, we combined testing the artificial system on a specific task with assessing the models (if any) that the artificial system is building to represent its self-state and the state of the external world.

Many works have been proposed to analyse LLMs' capabilities in different regards, but we are still lacking a solid systematic framework. Some studies tried to assess how well LLMs perform in understanding specific domains [22], and in [23] authors test how well LLMs perform logical reasoning. The work on an early version of GPT-4 in [24] highlights a diverse array of unexpected capabilities, many of which do not have direct or apparent links to language, and the authors go as far as asserting that the system exhibits features that can be those of an early artificial general intelligence system.

There are also some works testing spatial reasoning in LLMs, but they differ with what we wanted to address in this paper with a very specific experimental setting. The authors in [25] explore the advantages of employing ChatGPT for generating thematic maps based on public geospatial data, as well as using it to create mental maps based on textual descriptions of geographic space. A work that is much related to the one in this paper is [26], where the authors try to address the question of whether LLMs just memorize the good patterns or actually possess internal representations of the processes generating the observed sequences. They consider the board-game Othello and discover evidence that the model has an internal representation of the board state. The techniques used by these authors are more sophisticated than ours, and they argue that understanding the internal representations of the LLM may also be helpful to interpret and explain its decisions.

In [27], the author discusses the necessity of defining and evaluating intelligence to advance artificial systems effectively, highlighting the prevalent trend in the AI community to benchmark intelligence by comparing AI and human skills in specific tasks. He then argues that assessing intelligence in this manner is insufficient, as it overlooks the system's generalization abilities, heavily influenced by prior knowledge and experience. To address this, he introduces a new formal definition of intelligence rooted in Algorithmic Information Theory, defining intelligence as skill-acquisition efficiency. He then presents the Abstraction and Reasoning Corpus (ARC) as a comprehensive AI benchmark, with the goal of enabling fair comparisons of general intelligence between AI systems and humans.

While there has been research on conceptual abstraction in AI, especially using specific problems, these systems are often not thoroughly evaluated to determine their true understanding

of the involved concepts. In [28], authors argue that the ability to form and abstract concepts is specifically human, and it is currently lacking in advanced AI systems. As an evolution of the above-mentioned ARC, they introduce ConceptARC, a new benchmark that focuses on abstraction and generalization abilities within basic spatial and semantic concepts. The study shows that humans significantly outperform AI systems on ConceptARC, as they are better in abstracting and generalizing concepts.

The current trend of large models is to be multimodal, i.e., to include the use of different kinds of information besides textual (e.g., images, video and sound), and tasks that are similar to the one presented in this paper might be performed through the use of these Large Multimodal Models (LMMs). While there are some studies about these kind of models and their evaluation [29], it is a very recent technology that needs to be appraised properly.

In this regard, this kind of research will be increasingly important, because it will be necessary to assess how these models actually “understand” and decode space and spatial constraints. While we performed this on models which are purely linguistic, we believe that the same kind of study is indispensable when the inputs of the model are images or videos.

## 7. Conclusion

This work tried to measure the capacity of some Large Language Models to play the tic-tac-toe game, using it as a way to assess their capabilities to reason in a spatial context and to track information about their internal state for themselves and for the external world. This information, if used by the agent to make better choices, is one of the features that is believed to be essential for artificial consciousness. The main idea is that of asking the model not to just play against the opponent, but also to produce a list of the currently available winning sequences for itself and for the opponent. A positive correlation between a high accuracy in identifying currently available winning sequences and the ability to not lose (or to prolong the duration of the match) would show that the model is actually using some self-state and world-state information for achieving the given goals. However, in our case we were not able to see this correlation.

A major limitation of this work, which we are currently addressing, is related to the variability of the answers, since they are very much affected by the design of the prompts (in particular the “main” prompt and the “next-move” prompt). This variability deserves further investigations, since we need to measure how LLMs are consistent in their choices. In this regard, during our work we noticed that some LLMs are consistently choosing their moves depending on the previous interactions. This suggests that some LLMs have a preferred next-move, given the previous disposition of the game board. This aspect is still under analysis.

The results of this work suggest that, in terms of understanding which ones are the currently available winning sequences, Claude-2 outperforms both GPT-3.5 and GPT-4. Another point which is worth mentioning is that although we considered the length of each match as a measure to evaluate LLMs, this length might depend also on the implementation of the minimax algorithm.

We are currently in the process of implementing further instructions in the “main” prompt, such as asking LLMs to elucidate the rationale behind their decisions. Furthermore, we are

currently performing the same experiments on grids of larger sizes (e.g., 8x8 and 9x9). A future direction of our work is to try different ways to encode the data representing the grid, as it has been shown that GPT-4 improves its performance on ARC when the data is presented in a single row rather than in a grid[30]. We also plan to explore and measure the capability of Large Multimodal Models, which include not only textual but also graphical modalities. This will be increasingly important with the release of new models such as GPT-4V (GPT-4 enhanced with vision), which are able to reply to questions about an image and its context[31].

## References

- [1] The Cambridge Handbook of Consciousness, Cambridge Handbooks in Psychology, Cambridge University Press, 2007. doi:10.1017/CBO9780511816789.
- [2] D.J. Chalmers, The Conscious Mind: In Search of a Fundamental Theory, Oxford University Press, Inc., USA, 1996.
- [3] D. C. Dennett, Consciousness Explained, Penguin Books, 1991.
- [4] R. Van Gulick, Consciousness, in: E. N. Zalta, U. Nodelman (Eds.), The Stanford Encyclopedia of Philosophy, Winter 2022 ed., Metaphysics Research Lab, Stanford University, 2022.
- [5] A. Chella, R. Manzotti, Artificial Consciousness, Imprint Academic, 2007.
- [6] A. M. Turing, Computing machinery and intelligence, *Mind* 59 (1950) 433–60. doi:10.1093/mind/lix.236.433.
- [7] G. Oppy, D. Dowe, The Turing Test, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, Winter 2021 ed., Metaphysics Research Lab, Stanford University, 2021.
- [8] D. Jannai, A. Meron, B. Lenz, Y. Levine, Y. Shoham, Human or not? a gamified approach to the turing test, 2023. arXiv:2305.20010.
- [9] C. Bieber, Chatgpt broke the turing test - the race is on for new ways to assess ai, *Nature* 619 (2023) 686–689. doi:10.1038/d41586-023-02361-7.
- [10] D. J. Chalmers, Could a large language model be conscious?, 2023. arXiv:2303.07103.
- [11] C. Zaslavsky, Tic Tac Toe: And Other Three-In-A Row Games from Ancient Egypt to the Modern Computer, Crowell, 1982.
- [12] K. Crowley, R. S. Siegler, Flexible strategy use in young children’s tic-tac-toe, *Cognitive Science* 17 (1993) 531–561. URL: <https://www.sciencedirect.com/science/article/pii/036402139390003Q>. doi:10.1016/0364-0213(93)90003-Q.
- [13] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall Press, USA, 2009.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* 30 (2017).
- [15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. arXiv:1810.04805.
- [16] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.
- [17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal,

- K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, *Advances in Neural Information Processing Systems* 35 (2022) 27730–27744.
- [18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, *Advances in neural information processing systems* 33 (2020) 1877–1901.
- [19] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, 2023. [arXiv:2307.09288](https://arxiv.org/abs/2307.09288).
- [20] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, et al., Lamda: Language models for dialog applications, 2022. [arXiv:2201.08239](https://arxiv.org/abs/2201.08239).
- [21] G. Kim, P. Baldi, S. McAleer, Language models can solve computer tasks, 2023. [arXiv:2303.17491](https://arxiv.org/abs/2303.17491).
- [22] D. M. Katz, M. J. Bommarito, S. Gao, P. Arredondo, Gpt-4 passes the bar exam, Available at SSRN 4389233 (2023).
- [23] H. Liu, R. Ning, Z. Teng, J. Liu, Q. Zhou, Y. Zhang, Evaluating the logical reasoning ability of chatgpt and gpt-4, *arXiv preprint arXiv:2304.03439* (2023).
- [24] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al., Sparks of artificial general intelligence: Early experiments with gpt-4, *arXiv preprint arXiv:2303.12712* (2023).
- [25] R. Tao, J. Xu, Mapping with chatgpt, *ISPRS International Journal of Geo-Information* 12 (2023). URL: <https://www.mdpi.com/2220-9964/12/7/284>. doi:10.3390/ijgi12070284.
- [26] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, M. Wattenberg, Emergent world representations: Exploring a sequence model trained on a synthetic task, in: *The Eleventh International Conference on Learning Representations*, 2023. URL: [https://openreview.net/forum?id=DeG07\\_TcZvT](https://openreview.net/forum?id=DeG07_TcZvT).
- [27] F. Chollet, On the measure of intelligence, 2019. [arXiv:1911.01547](https://arxiv.org/abs/1911.01547).
- [28] A. Moskvichev, V. V. Odouard, M. Mitchell, The conceptarc benchmark: Evaluating understanding and generalization in the arc domain, 2023. [arXiv:2305.07141](https://arxiv.org/abs/2305.07141).
- [29] Z. Sun, S. Shen, S. Cao, H. Liu, C. Li, Y. Shen, C. Gan, L.-Y. Gui, Y.-X. Wang, Y. Yang, et al., Aligning large multimodal models with factually augmented rlhf, 2023. [arXiv:2309.14525](https://arxiv.org/abs/2309.14525).
- [30] Y. Xu, W. Li, P. Vaezipoor, S. Sanner, E. B. Khalil, Llms and the abstraction and reasoning corpus: Successes, failures, and the importance of object-based representations, 2023. [arXiv:2305.18354](https://arxiv.org/abs/2305.18354).
- [31] OpenAI, Gpt-4v(ision) system card, 2023.