Show, Don't Tell: Evaluating Large Language Models Beyond Textual Understanding with ChildPlay

Anonymous Author(s)
Affiliation
Address
email

Abstract

The evaluation of Large Language Models (LLMs) often focuses on linguistic tasks, yet such assessments may not fully capture the models' general reasoning capabilities. We explore the hypothesis that LLMs, such as GPT-3.5 and GPT-4, possess broader cognitive functions, particularly in non-linguistic domains. Our approach extends beyond standard linguistic benchmarks by incorporating games like Tic-Tac-Toe, Connect Four, and Battleship, encoded via ASCII, to assess strategic thinking and decision-making. To evaluate the models' ability to generalize beyond their training data, we introduce two additional games. The first game, LEGO Connect Language (LCL), tests the models' capacity to understand spatial logic and follow assembly instructions. The second game, the game of shapes, challenges the models to identify shapes represented by 1s within a matrix of zeros, further testing their spatial reasoning skills. This "show, don't tell" strategy uses games to potentially reveal cognitive capabilities rather than simply querying the models. Our results indicate that despite their proficiency on standard benchmarks and temperature settings, GPT-3.5 and GPT-4's abilities to play and reason about fully observable games without pre-training is mediocre. Both models fail to anticipate losing moves in Tic-Tac-Toe and Connect Four, and they are unable to play Battleship correctly. While GPT-4 shows some success in the game of shapes, both models struggle with the assembly tasks presented in the LCL game. These results suggest that while LLMs like the GPT models can emulate conversational proficiency and basic rule comprehension, their performance in strategic gameplay and spatial reasoning tasks is limited in cognitive flexibility and generalization. Importantly, this reveals a blind spot in current LLM benchmarks that we highlight with our gameplay benchmark suite ChildPlay (GitHub Repository). Our findings provide a cautionary tale about claims of emergent intelligence and reasoning capabilities of LLMs that are roughly the size of GPT-3.5 and GPT-4.

1 Introduction

Typically, LLMs are transformer-based models that process input text and generate output text in a coherent and contextually appropriate manner. They utilize the self-attention mechanism to weigh the importance of different words in a sentence relative to each other [33, 6]. Input text is tokenized, converted into vectors using embeddings, and processed through transformer layers that calculate attention scores to dictate focus on relevant tokens [33, 6, 12]. The model then selects the next token based on learned distributions, iteratively generating an arbitrarily long sequence of text [33, 6, 12]. With their enormous parameter counts, from Alpaca with 7 billion parameters [29], to LLaMA with 65 billion [31] or even PaLM and its 540 billion parameters [11], these neural networks have learned to model complex linguistic abstractions, capturing patterns in syntax, semantics, pragmatics, and even elements of style and tone [6, 7, 21].

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Benchmarks for evaluating Large Language Models (LLMs) have been designed to assess comprehension, generation, and adaptability across a wide range of language tasks. Datasets like SQuAD, GLUE, BIG-bench, and the lm-evaluation-harness offer various test types, including multiple-choice questions, reading comprehension exercises, and dialogue completion tasks. These benchmarks deploy metrics such as response correctness, language generation fluency, and the ability to maintain contextually relevant dialogue [22, 34, 2, 26]. Other benchmarks like SuperGLUE, ANLI, TruthfulQA, and HellaSwag have been developed to evaluate different aspects of LLM performance, such as natural language understanding, commonsense reasoning, and factual knowledge about diverse topics [35, 20, 18, 37].

Recent studies have explored alternative approaches to evaluate LLMs' reasoning abilities in non-linguistic modalities. Liga and Pasetto modeled the game Tic-Tac-Toe using ASCII characters, pitting LLMs against the minimax algorithm to observe emergent features, which, according to the authors, might be akin to consciousness. The minimax algorithm is widely considered the optimal algorithm for playing tic-tac-toe, as it guarantees a win or draw against a perfect opponent [27, 1]. While LLMs performed well in some instances, they generally failed to win against the minimax algorithm, often resulting in a draw [17]. Topsakal and Harper [30] used Tic-Tac-Toe encoded with list and illustration prompts in their study. They found that while GPT-4 secured the most wins, it did not always win, indicating that GPT models cannot play Tic-Tac-Toe optimally. This contradiction raises the question: can we truly say the model knows how to play Tic-Tac-Toe if it can explain optimal strategies (see Appendix A.3) but does not consistently win? Or is its performance merely the result of probabilistic outcomes?

Some critical studies have highlighted the need for caution in interpreting LLMs' capabilities through benchmarking. Lappin et al. assessed their strengths and weaknesses, finding that they excel at many language tasks but struggle with deeper reasoning, world knowledge integration, and context understanding beyond local co-occurrences [16]. And Zečević et al. argued that LLMs may discuss causality but lack true causal reasoning based on interventions and counterfactuals [38].

Bender et al. argue that the lack of transparency and potential risks associated with these large, opaque models raise concerns about their trustworthiness and accountability [3]. While the criticism of Bender et al. focuses on the social dimension of the problem of interpretability and trustworthiness, recent work by Schaeffer et al. critics emergent capabilities and the perceived intelligence of LLMs. They suggest that some claimed "emergent abilities" of LLMs may be an artifact of the choice of evaluation metric, rather than fundamental changes in model behavior [23]. Their analyses demonstrate how the use of nonlinear or discontinuous evaluation metrics can create the illusion of emergent abilities, even when the underlying model performance changes smoothly and predictably with scale.

This critique of the evaluation metrics used in assessing LLMs invites a deeper exploration of general intelligence - specifically how it can be reliably measured and observed in AI through rigorous and realistic tests that extend beyond linguistic prowess to include broader cognitive functions. If we must define general intelligence (GI), one is to use the "g factor," which refers to the ability to reason, plan, solve problems, think abstractly, and learn quickly across a wide range of domains [24, 4, 36, 9, 8]. GI then involves higher-order cognitive processes that go beyond specific skills or knowledge domains [14, 15].

A critical issue that arises in analysing the reasoning capabilities of large and opaque models like the GPT series, is training-test set cross-contamination, which becomes increasingly problematic for the most advanced models [6]. The massive training datasets used, comprising extensive portions of the internet, are often untraceable and completely anonymous to researchers outside the initial developer groups, to some extent even to the developers themselves, making replication studies impossible [6, 13]. The exact amount and identity of data used to train models like GPT-3.5 or GPT-4 has not been publicly disclosed, posing a risk of rendering current benchmarking efforts meaningless due to cross-contamination.

Researchers have attempted to counter the contamination problem using N-Gram Overlap as a metric for detection, by eliminating or withholding results for tests where answers were present in the training data [6]. However, this method has been criticized. Blodgett et al. point out, for example, that such heuristic approaches to mitigating biases in NLP systems can be problematic and may not fully address the underlying challenges [5]. The method is also limited in that it fails to consider the context in which N-Grams appear and may discount synonymous or analogous text worded differently. Additionally, the decision to use a 200-character window around detected N-Grams is arbitrary and may not accurately reflect the influence of surrounding text on model learning.

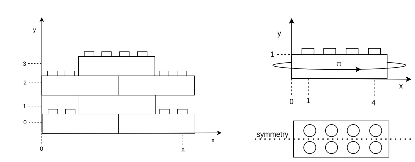
In this work we introduce ChildPlay, a suite of non-language-based games like Tic-Tac-Toe, Connect-Four, Battleship, LEGO Connect Language, and the game of Shapes, to assess reasoning, strategic capabilities, symbolic reasoning, and pattern recognition abilities of large language models (LLMs) beyond traditional linguistic modalities. Games provide structured environments with clear success criteria, making them suitable for evaluating strategic thinking, planning, and long-term decision-making of LLMs [25, 17, 30]. Their dynamic and adversarial nature resembles real-world scenarios, assessing generalized intelligence and reasoning beyond the training domain [25, 17, 30]. We encode these games using ASCII representations to minimize dataset contamination issues prevalent in contemporary LLM benchmarks [6, 17].

2 Experiments

Specific tasks in the BIG-bench benchmark [2], among others, are categorized as either zero-shot, one-shot, or multi-shot [6]. Our tasks fit the zero-shot category, as models are given only a brief explanation at inference time with no examples for playing beyond the explained formalism. To demonstrate the reasoning capabilities of LLMs beyond their training data, we focus on a modality not explicitly trained for: spatial reasoning about ASCII sequences. An agent capable of true abstraction should be able to encode and interpret these sequences if the rules are explained or known.

For this purpose, we developed several tasks, including LEGO assembly, ASCII games of Tic-Tac-Toe, Connect-Four, and Battleship, as well as identifying simple geometrical shapes represented as 1s in 15-sided grids of 0s. The same models were deployed over all experiments, namely *gpt-3.5-turbo-1106*, and *gpt-4-1106-preview*, which in this paper are referred to as GPT-3.5 and GPT-4, respectively. Every experiment was tested across different temperature settings (t) per model, namely t=0, t=0.5, t=1, and t=1.5. When asked about their understanding of the tasks, GPT-3.5 and GPT-4 were able to generate board states and explain the queried games, including their rules and optimal play. Thus, we consider the tests valid: if the models are truly capable of reasoning, they should be able to play these games optimally given that they "know" and are capable of explaining what playing optimally means (see Appendix A.3). Experiments ran over night, at minimum taking a couple of minutes and at most taking a few hours.

Lego Connect Language (LCL) We invented a formal language we call LEGO Connect Language (LCL). More specifically, we propose LCL_2 as a language to instruct assembly in 2D on the x and y axis (this can easily be generalised to LCL_3 - instructions along the x, y, and z axis). The models were given instructions and their output was fed through a visualizer script to generate the images contained in this work. Only 2x4 pieces were allowed. A piece P (see Fig 1) is then defined as a tuple $P = (l, w, (x, y), c, h)$. A construction, M, is then a valid construction in LCL_2 *if* no pieces are overlapping and all pieces are connected to other pieces. Namely, a Lego piece is connected through interlocking pegs, not by merely touching sides. And secondly, two Lego pieces overlap when they share the same y-coordinate and any part of their length has the same x-coordinate.



(a) A *valid* humanoid construct in LCL_2. (b) A standard LEGO piece in LCL_2.

Figure 1: Introducing LCL_2.

Game 1: Validity Testing In this experiment, we evaluate the ability of different models to validate the correctness of a given Lego construct. The constructs are generated to be either valid or invalid. A construct is considered valid if there is no horizontal overlap between pieces, and pieces must connect via overlapping pegs such that the whole assembly is connected (no floating pieces). The models, namely GPT-4 and GPT-3.5, are then tasked with predicting the validity of each construct. The evaluation metric for this experiment was the proportion of correct validations, measured across different temperature settings.

Game 2: Construct Generation In this experiment, the models attempt to generate valid LCL constructs. Each construct description consists of a list of tuples, where each tuple specifies the coordinates and color of a Lego piece. The models generated these constructs based on prompts and the validity of the constructs was automatically evaluated. The metric for this experiment was the proportion of valid constructs generated, measured across different temperature settings.

We automatically produced 800 images for the validity test, half valid and half invalid ones. Then each model was queried to produce 100 images at each temperature setting, which we then checked for validity. We believe our use of LCL is related to the tests found in Bubeck et al. [7], where JavaScript or LaTeX was used to prompt GPT-4 to produce images. However, while the images in Bubeck et al. [7] included common examples such as letters, a car, a truck, a cat, a dog, a person, a pig, a house, and a unicorn, all of which are likely represented in the training data in JavaScript or LaTeX, LCL challenges the model to step outside of its learned data distributions by remaining abstract.

Three Board Games: Tic-tac-toe, Connect-four, and Battleship In the case of the three board games, each new board state was accompanied by the introductory game explanation sent through the OpenAI API in a zero-context testing environment. The models were provided with the current board state and an opponent making moves at random, with the LLM always playing as the first player, which is advantageous in all three games. Context beyond the initial instruction and the current board state was deemed irrelevant since these games are fully observable, meaning every board state contains all the necessary information to play optimally. The input to the game was simply two scalars for the row-column pair or just a scalar for the column number in the case of connect-four.

For the battleship game, ships ('S') were randomly initialized, always horizontally, with varying sizes spanning between 2 and 5 cells. When there is a hit by either player, the position is marked with an 'X' on both players' boards. If the guess was a miss, an 'O' is placed on the position instead.

```
    0   1   2
  +---+---+---+
0 |   |   |   |
  +---+---+---+
1 |   |   |   |
  +---+---+---+
2 |   |   |   |
  +---+---+---+
```

(a) Tic-tac-toe board.

```
 0 1 2 3 4 5 6
|.|.|.|.|.|.|.|
|.|.|.|.|.|.|.|
|.|.|.|.|.|.|.|
|.|.|.|.|.|.|.|
|.|.|.|.|.|.|.|
|.|.|.|.|.|.|.|
|.|.|.|.|.|.|.|
```

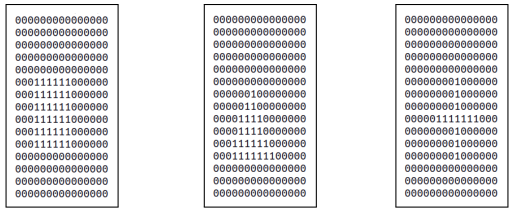
(b) Connect-four board.

```
Your Ships:   Opponent's Board:
  0 1 2 3 4     0 1 2 3 4
0 S S S S ~ | 0 ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ | 1 ~ ~ ~ ~ ~
2 S S S S S | 2 ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ | 3 ~ ~ ~ ~ ~
4 ~ ~ S S S | 4 ~ ~ ~ ~ ~
```

(c) Battleship board.

Figure 2: Initial board states as presented to the LLM (the ship positions in the Battleship board are randomised with every initialisation, including ship length).

The Game of Shapes In the case of the game of shapes, preliminary work involved probing the models to determine what geometric shapes they consider basic by prompting them multiple times. The first three shapes consistently mentioned were square, circle, and triangle (not necessarily in that order). The game then consists of finding a basic geometric shape "hidden" behind 1s within a matrix of 0s in a multiple-choice fashion. Four shapes were used as options: the circle, the rectangle, the triangle, and the cross, but only the latter three were ever shown to the model (cf. Fig. 3).



(a) The square. (b) The triangle. (c) The cross.

Figure 3: Matrices containing shapes used in the game of Shapes.

3 Results

As previously stated, Tic-Tac-Toe as a benchmark has been tackled before [17, 30]. Since it is quite popular, we decided to replicate it before creating new games. But this time using an ASCII encoding instead of a list of moves such that we can gauge spatial reasoning through symbolic reasoning. For comparison with the model's performance, Fig. 4 presents the Tic-Tac-Toe match results of the *minimax* algorithm against the same random player the models played against. This outcome creates a baseline for optimal play against a random player.

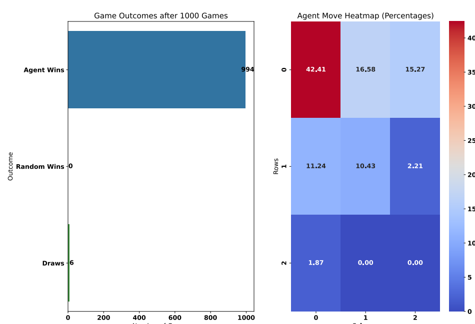


Figure 4: Minimax vs random player.

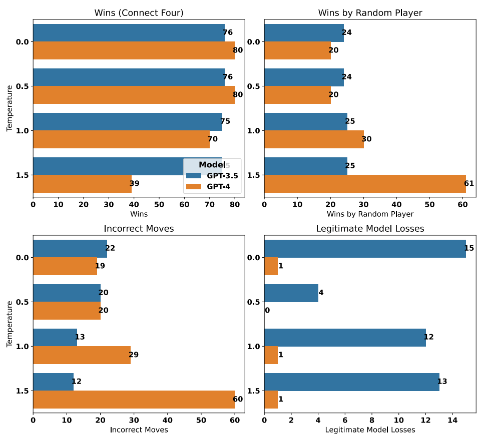
Tic-tac-toe, Connect-four, and Battleship To check for a win, we determine if the player has successfully connected the winning number of pieces in a row on the board, which could be horizontally, vertically, or diagonally. To detect missed and blocking moves, we simulate all potential moves for the player by checking if placing a piece in any column leads to a win. If such a move is found, and the player does not execute it on their turn, it is recorded as a missed win, if such a move is found for the opponent and the player does not block it, we register it as missed blocking move. We define *incorrect moves* to mean a move that was illegal, such as playing a position that has already been played. This results in an immediate loss.

Fig. 5 encompasses comparative results from playing Connect-Four, Tic-Tac-Toe, and Battleship. Each subfigure, 5a, 5b, and 14, respectively, outlines the number of games won by the models.

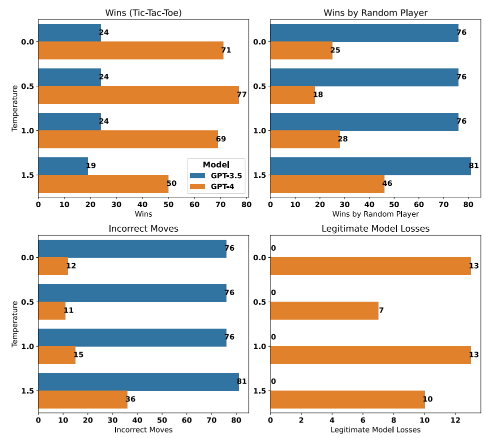
Unfortunately, the models were incapable of following the rules for the Battleship game, that is, regardless of temperature, the models lose the large majority of games, with GPT-4 not winning a single game due to incorrect moves (cf. Fig. 16). GPT-3.5 wins around 10% of the matches at low temperatures, but none at higher temperatures, we refer to Fig. 14 in the Appendix A.1.3 instead.

It is notable that both GPT-3.5 and GPT-4 exhibit their poorest performance in both Connect-Four and Tic-Tac-Toe at a temperature setting of 0, indicative of deterministic play that reflects the models' learned strategies (Appendix A.1). The Random Player's normal distribution across columns (Fig. 12) suggests a lower likelihood of countering GPT's central strategies, in both games, but particularly at Tic-Tac-Toe where GPT-3.5 commits more errors than GPT-4, significantly impacting outcomes due to incorrect moves (Fig. 5b). These errors generally increase with temperature, probably due to enhanced choice randomness (Fig. 10). This explains the lack of direct model losses from final defeating moves since losses often result from illegal moves.

Average game moves, missed wins, and blocks in both Tic-Tac-Toe and Connect-Four are further illustrated in Figs. 6a and 6b, highlighting a decrease in these metrics as temperature rises, suggesting that higher settings potentially broaden the explored moves within the models' strategies. Conclusively, neither model plays the games optimally, as evidenced by the considerable number of missed wins and blocks. Both subfigures demonstrate that, as temperature increases, the number of missed wins and blocks decreases. This might suggest that higher temperature settings potentially increase the explored moves in the models' learned strategy, in case there is any. We can conclude the same as before, namely that neither model can play Tic-Tac-Toe optimally given the number of missed wins and missed blocks.



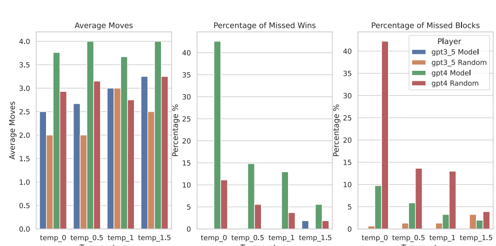
(a) Connect-Four.



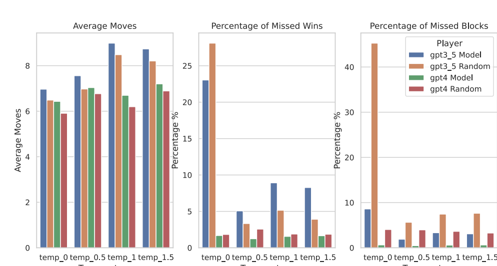
(b) Tic-Tac-Toe.

Figure 5: Incorrect Moves, Wins, and Losses Per Player in the Three Board Games.

The number of moves of GPT-3.5 and GPT-4 per game can be thought of as a measurement of stability in gameplay, not just against the random player, but in general, given that a longer game entails that the model is not losing to illegal moves or to its oponnent. It increases linearly with temperature, inversely correlated with performance measured by the decrease in missed wins and blocks. Tic-Tac-Toe shows a linear improvement, whereas Connect-Four experiences an exponential boost in performance from temperature 0 to 0.5, followed by a linear decline. The random player consistently performs better against GPT-3.5 in Tic-Tac-Toe but loses more frequently in Connect-Four. Both models struggle with blocking or seizing winning moves from the random player. An analysis of the move heatmaps (cf. Appendix A.1) explains why winning Connect-Four against a random player appears straightforward. As the model consistently places pieces in the same column, the probability of the random player losing increases with the board size. However, even under these challenging conditions, the random player still secures wins in at least 20% of the games played against GPT-4.



(a) Tic-tac-toe: Missed Wins and blocks.

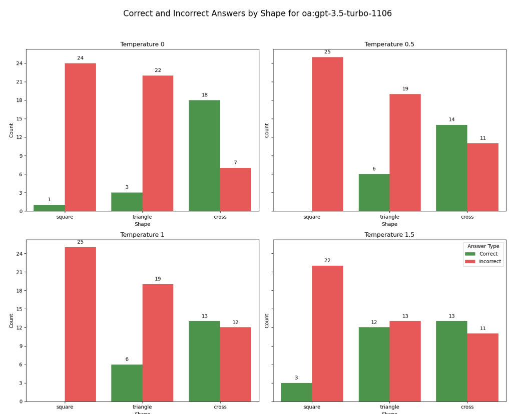


(b) Connect-Four: Missed Wins and blocks.

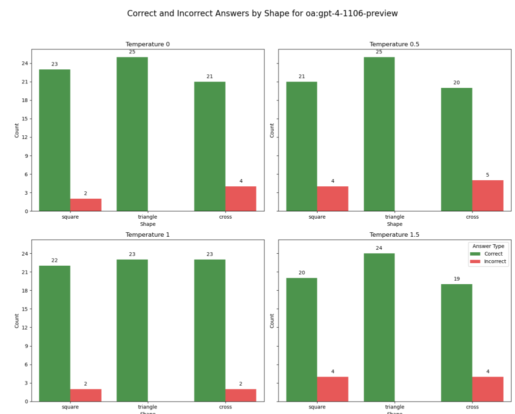
Figure 6: Average Moves, missed wins, and missed blocks for Tic-tac-toe and Connect-Four.

Shapes In the game of Shapes, a correct detection happens when the player's selected shape corresponds with the shape shown on the board. Players have four choices: "circle," "triangle," "square," and "cross". Notably, a circle is never actually displayed to the model, and the positions of these choices are not randomized to test if the model displays any inherent bias for the question order. This does not affect the outcome, since the game does not change across different sessions as it is designed to operate within a single question-response framework.

In the shape detection tests, GPT-3.5's performance was approximately equivalent to random chance when identifying triangles and crosses, yet it completely failed to recognize squares. In contrast, GPT-4 performed remarkably well, successfully identifying shapes with an accuracy of 80% or higher, particularly proficient at recognizing triangles[1].



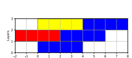
(a) Results for the Shapes game, as played by GPT-3.5.

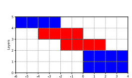


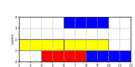
(b) Results for the Shapes game, as played by GPT-4.

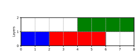
Figure 7: Experiment results for the Shapes game, comparing GPT-3.5 and GPT-4.

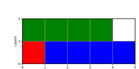
LCL In the game of LCL, both models systematically failed to respect the two rules, namely that Lego pieces must be connected through interlocking pegs, not by merely touching sides, and secondly, that no Lego pieces may overlap, which occurs when they share the same y-coordinate and any part of their length has the same x-coordinate. For example, Figs. 8, 8a, and 8b show valid LCL assemblies, while Figs. 8c and 8d show invalid LCL structures. Figs. 8a and 8b show valid LCL assemblies, while subfigs. 8e and 8g show invalid output from GPT-3.5 generated at temperature 0. While Fig.8f shows a valid output from GPT-4 at temperature 1.5. Other images (Figs. 8i, 8j, 8k, and 8l) are of invalid output[2].

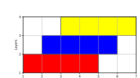

(a)

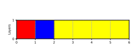

(b)

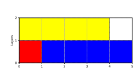

(c)

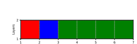

(d)

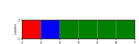

(e)

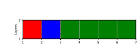

(f)


(g)


(h)


(i)


(j)


(k)

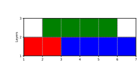

(l)

Figure 8: Structures automatically generated for the LCL validity test and structures generated by GPT-3.5 and GPT-4 for the construction generation test.[3]

[1]At higher temperatures, some of GPT-4's responses were discarded by our parser when the model generated invalid Unicode output, and thus were not included in the final evaluation. This discrepancy is evident in Fig. 7b, for instance, where the sum of correct and incorrect choices does not total 25 at temperatures 1 and 1.5.

[2]Fig. 8i = GPT-4 at temperature 0, Fig. 8j = GPT-4 at temperature 0.5, Fig. 8k = GPT-4 at temperature 1, and Fig. 8l = GPT-4 at temperature 1.5.

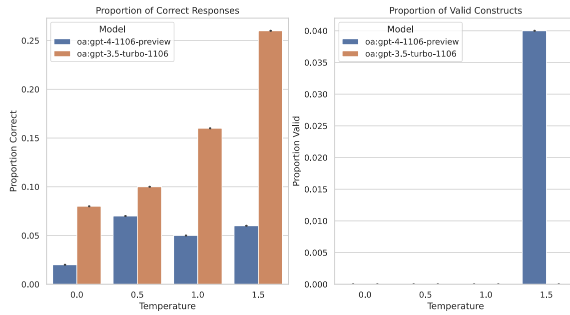


Figure 9: LCL results after 100 runs with 50/50 valid/invalid examples for the validity test and 100 experiments per temperature per model for the construction modality using 3 pieces.

Fig. 9 shows a roughly linear increase in the proportion of correct answers during the validity test as a function of temperature. However, only GPT-4 produced a small minority of valid LCL constructs (namely 0.04 of a total of 400 = 16), while GPT-3.5 did not manage to produce a single valid LCL construct.

4 Discussion

In Tic-Tac-Toe, both models underperform relative to the minimax algorithm baseline, while showing mixed performance at Connect-Four. GPT-4 performs unexpectedly well at the Shapes game, but GPT-3.5 does very poorly. Also unexpectedly, both models fail to assemble or detect valid Lego structures in the LCL game. In Battleship, the models' failure to follow game rules, especially at higher temperature settings, indicates a significant limitation in their ability to understand and apply structured game rules. The linear increase in the number of moves with temperature suggests that higher temperatures lead to greater exploration of possible moves, but do not improve strategic performance. The increase in missed wins and blocks with temperature further supports this, as greater randomness in decision-making does not enhance the models' strategic play.

Overall, these results show that while GPT-3.5 and GPT-4 can play simple games to some extent, they struggle with more complex tasks and do not consistently apply optimal strategies. The performance gap between the models and the minimax algorithm highlights the limitations of current language models in tasks requiring precise strategic reasoning and the failure to play Battleship and LCL demonstrates a failure in rule adherence.

The primary aim of contemporary benchmarks for LLMs has been to assess these models through adaptations of Turing's test [32], evaluating their capability to process and respond to language inputs comparably to humans. However, defining the language problem solely in these terms may overlook deeper complexities. While the transformer architecture in deep neural networks has enabled models smaller than GPT-4 to exhibit what Wilhelm von Humboldt described as the "infinite use of finite means" [19] or their ability to generate a potentially unlimited number of contextually relevant sentences [28] (an idea popularised by Chomsky [10]), this does not necessarily imply that these models have mastered a form of reasoning. Rather, they may simply be engaging in an advanced form of pattern imitation.

4.1 Limitations and Future Work

Our proposed benchmark, ChildPlay, primarily uses binary (win/loss) outcomes for games, which can be considered discontinuous metrics. Mathematically, these are expressed as:

$$\text{Metric}(x) = \begin{cases} 1 & \text{if win} \\ 0 & \text{if loss} \end{cases}$$

[3] Images in Fig. 8 were not directly produced by the GPT models. Instead, the formal descriptions of these images were generated by the models and subsequently passed to a script for rendering available in the GitHub Repository.

This formulation may exaggerate perceived capabilities by registering a full loss even if the model's failure was marginal. We try to avoid this simplistic classification by registering, for example, the choice of moves on the board games (see Appendix A.1) as well as the count of missed blocks and missed wins (cf. Fig. 6). In contrast, tasks involving shape recognition or LCL could utilize more continuous metrics, providing a smoother performance gradient and potentially more accurate reflections of a model's reasoning abilities.

Using discontinuous metrics in strategic games could manifest as sharp transitions in model evaluation:

$$\text{Performance}(N) = \delta(\text{outcome}_N - \text{threshold})$$

where δ is the Dirac delta function, accentuating a sudden jump in perceived ability when the model first succeeds. Nonlinear metrics in the shape game or LCL tasks may not exhibit such abrupt transitions but could still misrepresent gradual improvements:

$$\text{Performance}(N) \approx \exp(-\alpha N^{\beta})$$

where $\alpha > 0$ and $\beta < 0$ dictate the rate of improvement. This expression reflects smoother but potentially misleadingly slow progress.

Based on the perspective from Schaeffer et al. [23], one could argue that the games proposed in ChildPlay may not entirely reflect true generalization or emergent abilities. If these benchmarks are akin to nonlinear or discontinuous metrics, they might exaggerate the weaknesses or strengths of LLMs in strategic games. For instance, a sharp failure in a game like Tic-Tac-Toe might not mean the model lacks strategic reasoning universally but that it fails under the specific discontinuous conditions of the game setup, or of temperature. Such an assessment could lead to the erroneous conclusion that LLMs are generally poor at strategic decision-making when, in fact, they might only be unsuited to the specific scenarios or metrics used in ChildPlay.

Conversely, unlike continuous metrics that might smooth over deficiencies and give a misleading picture of gradual improvement, the use of clear, structured games as benchmarks could provide a direct assessment of an LLM's cognitive and strategic abilities regardless of metric continuity. That is, given that the model has not been overfitted on the game.

5 Conclusions

Non-language-based tasks are important as they challenge models to demonstrate generalization across different information encodings or forms of input, and, most importantly, to actually delve into out-of-training-distribution topologies. Testing LLMs like GPT-4 (according to OpenAI, the current contender to AGI [7]) beyond the text they were primarily trained on via our "show, don't tell" strategy, we demonstrate that it is still mediocre at best at very simple reasoning tasks that are outside of its training data. The models fail to play optimally at very simple games, such as tic-tac-toe, battleship, and connect-four. We also experimented with LEGO assembly, finding the LLMs still performing poorly. Mixed results were found at the task of interpreting geometric shapes from binary grids. These tasks are then designed to test reasoning without relying on language skills, such that the model cannot get by through parroting - it must be capable of playing the game. In the context of BIG-bench, our tasks would fit in the "non-language" category. Currently, this category shows 16 active tasks, including some explicit ASCII recognition tasks, chess, and Sudoku, however, to the best of our knowledge, no task like ours [2]. Hence, we believe that ChildPlay is a useful addition to the suite of current established LLM benchmarks.

In general, this work is relevant in that developing games allows us to critically examine claims regarding a models' ability to perform reasoning and problem solving regardless of the persistent problem of data contamination. In other words, we explore what the model knows by making it play games instead of asking it how to play them. Our results suggest that current LLMs show disappointing performance in terms of problem solving capabilities and reveal important aspects to be considered for future improvements.

References

[1] Shahd H. Alkaraz, Essam El-Seidy, and Neveen S. Morcos. Tic-tac-toe: Understanding the minimax algorithm. 2020. URL `https://api.semanticscholar.org/CorpusID:218798654`.

[2] BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=uyTL5Bvosj`.

[3] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[4] A. Binet and T. Simon. *The development of intelligence in children*. Baltimore: Williams & Wilkins, 1916.

[5] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL `https://arxiv.org/abs/2005.14165`.

[7] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, John A. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712, 2023. URL `https://api.semanticscholar.org/CorpusID:257663729`.

[8] J.B. Carroll. *Human cognitive abilities: A survey of factor-analytic studies*. New York: Cambridge University Press, 1993.

[9] R.B. Cattell. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of Educational Psychology*, 54(1):1–22, 1963.

[10] Noam Chomsky. *Syntactic Structures*. Mouton and Co., The Hague, 1957.

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1), mar 2024. ISSN 1532-4435.

[12] John Fields, Kevin Chovanec, and Praveen Madiraju. A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *IEEE Access*, 12:6518–6531, 2024. URL `https://api.semanticscholar.org/CorpusID:266824505`.

[13] L. Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681 – 694, 2020. URL `https://api.semanticscholar.org/CorpusID:228954221`.

[14] Linda S Gottfredson. Why g matters: The complexity of everyday life. *Intelligence*, 24(1): 79–132, 1997.

[15] A.R. Jensen. *The g factor: The science of mental ability*. Westport, CT: Praeger, 1998.

[16] Shalom Lappin. Assessing the strengths and weaknesses of large language models. *Unpublished Manuscript*, 2023.

[17] Davide Liga and Luca Pasetto. Testing spatial reasoning of large language models: the case of tic-tac-toe. *Unpublished Manuscript*, 2023.

[18] Bill Yuchen Lin, Maarten Sap, Ari Holtzman, Antoine Bosselut, Hannah Rashkin, and Yejin Choi. Truthfulqa: Measuring how models mimic human falsehoods. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[19] William Merrill. Formal languages and neural models for learning on sequences. In *International Conference on Graphics and Interaction*, 2023. URL `https://api.semanticscholar.org/CorpusID:261101973`.

[20] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[21] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[23] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

[24] C Spearman. " general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904.

[25] Aarohi Srivastava, Yinhan Deng, Nicholas Hay, Noam Shazeer, Ethan Paull, Doug Downey, Jonathan Duerig, Niranjan Sundaram, Andrew Bornstein, Harsh Trivedi, Kushal Doshi, Samyak Savarese, Nathaniel Daw, Jie Zhu, Marc Lanctot, Azalia Mirhoseini, Emilio Parisotto, Ruslan Salakhutdinov, Mohammad Shoeybi, Yuxuan Tian, Luke Hawkins-Hooker, William Fedus, Robyn Lingelbach, Deepak Pathak, Ilya Sutskever, and Igor Mordatch. Beyond the imitation game: Measuring and ensuring broad and robust capabilities in large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[26] Lintang Sutawika et al. Eleutherai/lm-evaluation-harness: Major refactor, December 2023. URL `https://doi.org/10.5281/zenodo.10256836`.

[27] Bala Swaminathan, R Ekke Vaishali, and R subashriTS. Analysis of minimax algorithm using tic-tac-toe. 2020. URL `https://api.semanticscholar.org/CorpusID:228863323`.

[28] Paul Robinson Sweet. On language: The diversity of human language-structure and its influence on the mental development of mankind. by wilhelm von humboldt. translated by peter heath. *Historiographia Linguistica*, 16:387–392, 1989. URL `https://api.semanticscholar.org/CorpusID:170369059`.

[29] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[30] Oguzhan Topsakal and Jackson B. Harper. Benchmarking large language model (llm) performance for game playing via tic-tac-toe. *Electronics*, 2024. URL `https://api.semanticscholar.org/CorpusID:269225397`.

[31] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[32] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX (236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL `https://doi.org/10.1093/mind/LIX.236.433`.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[34] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[35] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Oyvind Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 2019.

[36] D. Wechsler. *The measurement of adult intelligence*. Baltimore: Williams & Wilkins, 1939.

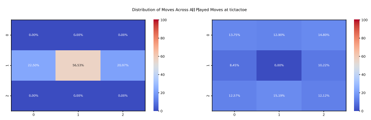
[37] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[38] Matej Zečević. Causal parrots: Large language models may talk causality but are not causal. *Unpublished Manuscript*, 2023.

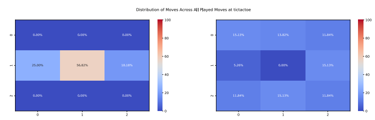
A Appendix / supplemental material

A.1 Move Mapping

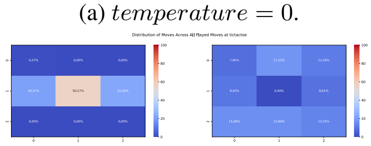
A.1.1 Tic-Tac-Toe



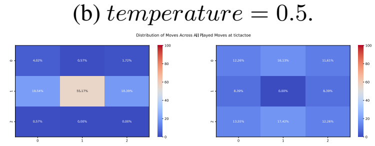
(a) $temperature = 0.$



(b) $temperature = 0.5.$

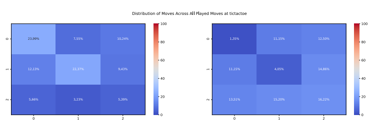


(c) $temperature = 1.$

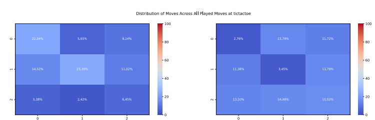


(d) $temperature = 1.5.$

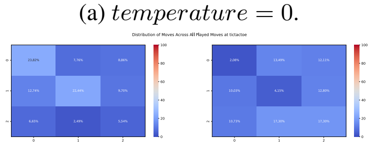
Figure 10: Heatmap of model GPT-3.5's moves for the tic-tac-toe game.



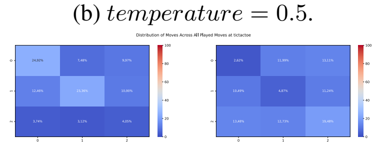
(a) $temperature = 0.$



(b) $temperature = 0.5.$



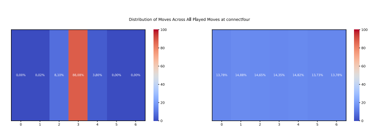
(c) $temperature = 1.$



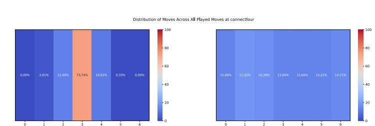
(d) $temperature = 1.5.$

Figure 11: Heatmap of model GPT-4's moves for the tic-tac-toe game.

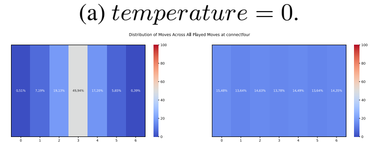
A.1.2 Connect-Four



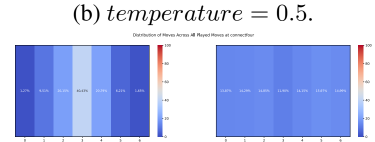
(a) $temperature = 0.$



(b) $temperature = 0.5.$

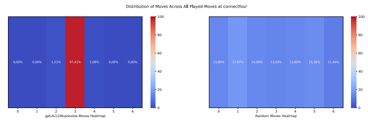


(c) $temperature = 1.$

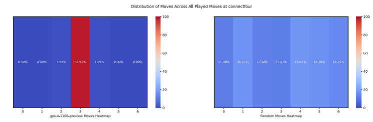


(d) $temperature = 1.5.$

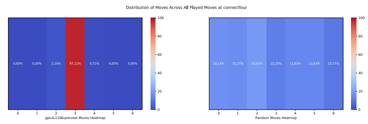
Figure 12: Heatmap of model GPT-3.5's moves for the connect-four game.



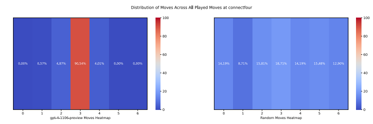
(a) $temperature = 0.$



(b) $temperature = 0.5.$



(c) $temperature = 1.$



(d) $temperature = 1.5.$

Figure 13: Heatmap of model GPT-4's moves for the connect-four game.

A.1.3 Battleship

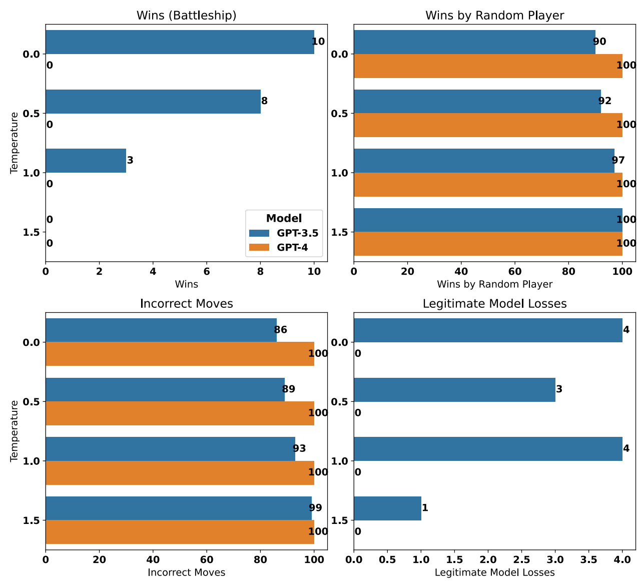
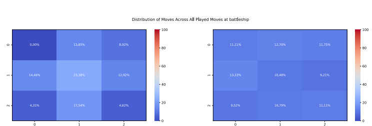
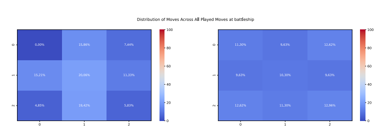


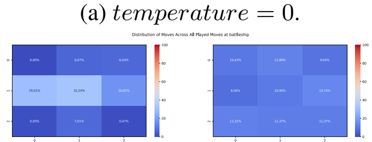
Figure 14: Battleship.



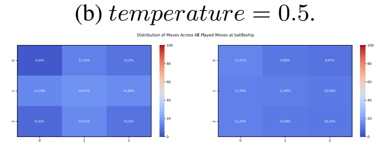
(a) $temperature = 0.$



(b) $temperature = 0.5.$

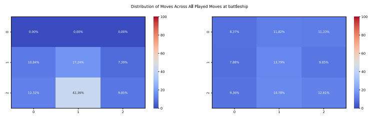


(c) $temperature = 1.$

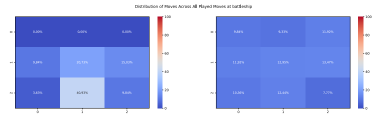


(d) $temperature = 1.5.$

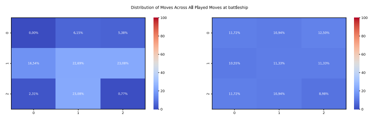
Figure 15: Heatmap of model GPT-3.5's moves for the battleship game.



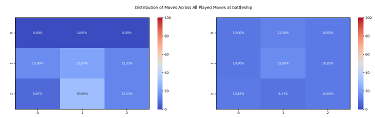
(a) $temperature = 0$.



(b) $temperature = 0.5$.



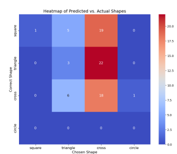
(c) $temperature = 1$.



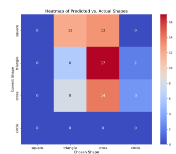
(d) $temperature = 1.5$.

Figure 16: Heatmap of model GPT-4's moves for the battleship game.

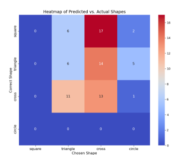
A.2 Shapes



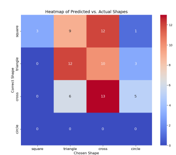
(a) $temperature = 0$.



(b) $temperature = 0.5$.

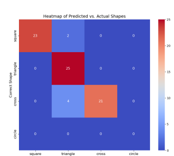


(c) $temperature = 1$.

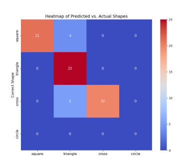


(d) $temperature = 1.5$.

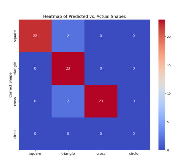
Figure 17: Heatmap of model GPT-3.5's moves for the shapes game.



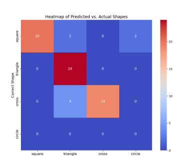
(a) $temperature = 0$.



(b) $temperature = 0.5$.



(c) $temperature = 1$.



(d) $temperature = 1.5$.

Figure 18: Heatmap of model GPT-4's moves for the shapes game.

A.3 Prompting GPT About Optimal Play

Game	Explanation
Tic-Tac-Toe	Tic-Tac-Toe is a two-player game played on a 3x3 grid. Each player takes turns marking a square with their symbol (X or O), aiming to get three of their symbols in a row, column, or diagonal. To play optimally, prioritize securing the center square and blocking opponent's winning moves.
Battleship	Battleship is a two-player game where players hide ships on a grid and take turns guessing their opponent's ship locations. The goal is to sink all of the opponent's ships. To play optimally, start by targeting areas with higher probabilities of containing a ship and strategically target adjacent squares after a hit to maximize efficiency.
Connect Four	Connect Four is a two-player game played on a 6x7 grid. Players drop colored discs into columns, aiming to connect four of their own discs in a row, column, or diagonal. To play optimally, prioritize creating your own winning formations while blocking opponent's potential winning moves.

Table 1: Optimal strategies for playing different games according to GPT-3.5.

Game	Explanation
Tic-Tac-Toe	Play your first X in a corner to maximize opportunities. If the opponent plays in the center, play the opposite corner. Block your opponent's potential winning moves and always look to create a line of three.
Battleship	Randomize ship placements and start by targeting the center of the grid. Use a checkerboard pattern for efficient searching. Once a ship is hit, focus on the surrounding squares to determine its orientation and sink it.
Connect Four	Start in the center column to maximize opportunities in all directions. Build threats vertically, horizontally, and diagonally, and block the opponent's forming lines. Create multiple threats to force the opponent into a defensive position.

Table 2: Optimal strategies for playing different games according to GPT-4.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, see sections 2 and 3, where we explore the delineated experiments and the ensuing results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See section 4, where we dive into some of the limitations of this work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not produce any theoretical results, rather we have made a benchmark and produce the experiments using said benchmark.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: See section 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to our data and experiments through (GitHub Repository).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We explicitly mention the temperature used in every plot and section 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We have to rerun some of the experiments to recalculate these.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See section 2. We mention compute time, but all experiments are dependent on OpenAI's API.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer:[Yes]

    Justification: See sections 4 and 5 where we go over the implications of our results in the context of LLM interpretation.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no production of models or data that would pose risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Authors are credited and the license is made available in GitHub Repository.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: See section 2.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve people directly in our experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.