LLMAO: Evaluating AI Agents powered by LLMs

Kaushal Patil
University of Illinois Chicago
Chicago, USA
kpati@uic.edu

Harshal Hirpara
University of Illinois Chicago
Chicago, USA
hhirp@uic.edu

Andrea Piras
University of Illinois Chicago
Chicago, USA
apiras2@uic.edu

Sai Shridhar Balamurali
University of Illinois Chicago
Chicago, USA
sbalam3@uic.edu

ABSTRACT

The rapid advancements in artificial intelligence have led to the development of specialized algorithms, from heuristic search methods to reinforcement learning-based Deep Q learning, each optimized for specific tasks. However, the emergence of Large Language Models (LLMs) such as ChatGPT, Vicuna and LLama 7B presents new opportunities for enhancing AI agents across diverse environments. This project investigates the integration of LLMs into AI Agents evaluated in various gym environments, aiming to compare their performance with traditional AI agents. A key focus of our research is the implementation of innovative techniques like chain of thought and few-shot learning, aimed at improving the reasoning and adaptability of LLMs. Our goal is to uncover the true potential of LLM-powered AI agents, assessing whether they stand as strong competitors or highlight areas needing improvement. This exploration is pivotal in understanding the capabilities and limitations of LLMs in dynamic and complex scenarios, offering insights into their effective integration and potential transformation of the AI research and application landscape.

KEYWORDS

ChatGPT, LLAMA, Vicuna, Mistral, AI agents, Few-Shot Learning, Chain of Thought Prompting.

The code repository and data for this project can be found at https://github.com/Kaushal1011/Evaluating-AI-Agents-powered-by-LLMs

1 INTRODUCTION

In recent years, the world of AI agents has experienced significant advancements, with specialized algorithms ranging from heuristic search methods to reinforcement learning-based Deep Q learning. These models have traditionally been optimized for specific tasks. However, the emergence of Large Language Models (LLMs) like ChatGPT and LLama 7B presents a novel query: can these LLMs be harnessed to augment AI agents in diverse environments? In our project, we set out to investigate this intersection by embedding LLMs into various gym environments and comparing their performance against conventional AI agents. Building on this foundation, we experimented with enhancing model output through innovative techniques like chain of thought and few-shot learning. These methods, aimed at improving the reasoning and adaptability of LLMs, could potentially redefine their role in complex environments. By integrating these techniques, we seek to push the boundaries of LLM capabilities, exploring how they can not only understand but also interact more effectively within various simulated scenarios.

1.1 Goal

Our primary objective in this venture was to uncover the true potential of Large Language Model (LLM) powered AI agents. We aim to determine whether these LLM-enhanced agents emerge as strong contenders in the realm of artificial intelligence or whether they highlight specific areas that require further improvement. This exploration is crucial in understanding the capabilities and limitations of LLMs when applied to dynamic and complex environments. By rigorously testing these models in various scenarios, we tried to gain valuable insights into how LLMs can be effectively integrated into AI agents, thereby potentially transforming the landscape of AI research and application.

2 ENVIRONMENT SELECTION

To accomplish our research objectives, we utilized three distinct single-objective and one multi-objective simulation environments.

The first environment was a Blackjack game, sourced from the Gymnasium framework [5]. This particular environment operates with a binary action space: ’0’ to signify ’Stand’ and ’1’ for ’Hit’. The primary goal in this environment is to accurately forecast the player’s next move.

For our second environment, we selected a grid-based navigation game, shown in Figure 1. In this game, the objective is to determine the most efficient route from a designated start point, labeled ’S’, to a target destination, marked ’G’. The model navigates by moving Left, Right, Up, or Down, striving to reach the goal using the shortest path possible.

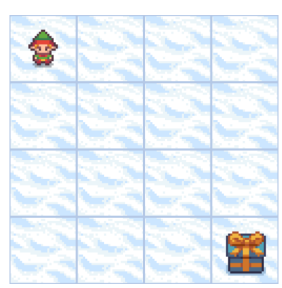

Figure 1: Grid-based navigation game. The player at S=(0,0) needs to reach G=(4,4)

In the third environment: "Decision from Weather Environment" game, agents face the challenge of choosing the right clothing and action based on the external weather conditions. They have three options: applying sunscreen for sunny weather, using an umbrella for rain, or wearing snow boots for snowy conditions. This environment test agents' ability to make appropriate decisions in multi-objective scenarios.

In the fourth environment: "Traffic Signal Evaluation" game, agents need to optimize traffic flow at intersections, making real-time decisions for signal control based on vehicle and pedestrian traffic data. They consider various factors, such as vehicle count and waiting times in different directions, to efficiently manage North-South and East-West traffic signals and are evaluated based on their ability to reduce congestion and prioritize emergency vehicles.

3 LLMS SELECTION

We decided to use 5 different LLMs, summarized in table **??**. Each model has varying number of parameters, training dataset and task for which they were optimized:

- GPT-4 [8]: the latest iteration of the renowned GPT series, is a natural language processing powerhouse. With a considerable increase in model size and training data, GPT-4 offers advanced language understanding and generation capabilities;
- GPT-3.5 [2]: A highly advanced AI language model by OpenAI, featuring 175 billion parameters, known for its exceptional language understanding and generation skills.
- Mistral Instruct [3]: 7-billion-parameter language model engineered for superior performance and efficiency. Mistral 7B takes a significant step in balancing the goals of getting high performance while keeping large language models efficient;
- Llama : A 13 billion parameter open-source language model developed by Facebook Research, advancing the capabilities of natural language processing;
- Vicuna [7]: an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT;
- OpenChat 3.5 [4]: OpenChat 3.5 is a new model that uses Mistral-7B as its base. It was fine tuned using C-RLFT on on a collection of publicly available high-quality instruction data.According to the authors, its performance was considered comparable to chatgpt-3.5-turbo(March 2023).

Model	Parameters
GPT-4	1.3T (estimate)
GPT-3.5	175B
LLama 2	13B
Mistral	7B
Vicuna	13B
OpenChat	7B

Table 1: LLMs used for the project

4 EXPERIMENTAL SETUP

To begin our experiment, we initially evaluated both reinforcement learning agents and a random policy for baseline comparisons. Later, we utilized LMstudio [6] to create local servers which utitlize openAI's API for the Large Language Models (LLMs) under study. In our methodology, an agent would interact with an environment attempting to complete a certain task. The LLM would act as the "brain" of the agent. We utilized the state of the environment as input data for the LLM by converting them to prompts that it can comprehend.Subsequently, it would generate an appropriate response which would be converted to an action. Following this, the environment determined the subsequent change in the agent state resulting from the implemented action and produced a new state. This iterative process continued until the desired goal was achieved by the agent. After each session, we documented the performance of the model. This data allowed us to effectively gauge the LLMs' performance in the assigned tasks in comparison to original RL agents.

For the baseline the prompts that we used are as follows:

> "In a game of blackjack, the dealer has one card showing and one card hidden.
> The visible card is a 3. I have 2 cards whose sum is 18. You have a usable Ace.
> Given we win close to 21 and lose badly at anything above it (going above 21 should be avoided), Do I hit or stand, hit will lead to drawing one card at random and increasing the total, stand will keep the same total?
> The first line of your answer should be 'Hit.' or 'Stand.' The second line should be the justification."

Prompt for Blackjack.

> "Your current position: (s1,s2), you have to reach (e1,e2). Answer which direction you move in. Options A) Up B)Down C) Left D) Right"

Prompt for Navigation.

> "Consider the following data about weather conditions for the last 7 days:
> Temperatures (in °F): [47.348949541566256, -31.622330311862147, 25.32772017492796, -41.84648929886844, 13.360171709887624, 4.580407868630644, 60.09024980191904] Weather Conditions: ['Sunny', 'Sunny', 'Sunny', 'Snow', 'Snow', 'Sunny', 'Rain']
> Based on the above data, what should be the ideal accessory (Sunscreen, Umbrella, Snowshows) and clothing (Warm, Medium, Summer) choice that I should make tommorow? Please explain your reasoning."

Prompt for Decision from Weather Environment.

Following our initial baseline prompting, we delved into enhancing the accuracy of our model by employing chain of thought and few-shot learning techniques. Our hypothesis is that these methods will significantly improve the Large Language Model's (LLM's) understanding of the game's context. To implement this, we crafted chain of thought prompts that guide the LLM through a more structured reasoning process, as outlined below:

> User:
> You are playing a game of blackjack.
> You have a hand which totals (hand_total) (usable_ace).
> The dealer has a hidden card and a card visible to you. The visible card is (dealer_hand).
> Should you hit or stand? After thinking, give me either hit or stand in a new line.
>
> Assistant:
> Let's think step by step,

Chain of Thought Prompt for Blackjack.

> User:
> You are a navigator bot provide context of how I want to control navigation. you output directions, Top, Bottom, Right and Left.
>
> Your current position: (0,3), you have to reach (3,1). Answer which direction you move in. Options A) Up B) Down C) Left D) Right.
>
> Assistant:
> Lets think step by step:
> 'Step 1: Current Position: (0,3), Target: (3,1). Analyze: Need to move down to reach row 3 and then left to reach column 1. Next Move: Down.
> Step 2: Current Position: (1,3), Target: (3,1). Analyze: Still need to move down to row 3, but also left to column 1. Next Move: Left.
> Step 3: Current Position: (1,2), Target: (3,1). Analyze: Continue moving down to row 3, then left to column 1. Next Move: Down.
> Step 4: Current Position: (2,2), Target: (3,1). Analyze: Move down to row 3, followed by left to column 1. Next Move: Left.
> Step 5: Current Position: (2,1), Target: (3,1). Analyze: Only need to move down to align with row 3, as column is already aligned. Next Move: Down.'
>
> User:
> You are a navigator bot provide context of how I want to control navigation. you output directions, Top, Bottom, Right and Left.
>
> Your current position: (s1,s2), you have to reach (e1,e2). Answer which direction you move in. Options A) Up B) Down C) Left D) Right.

Chain of Thought Prompt for Navigation.

> Question:
> Consider the following data about weather conditions for the last 7 days:
> Temperatures (in °F): [47.348949541566256, -31.622330311862147, 25.32772017492796, -41.84648929886844, 13.360171709887624, 4.580407868630644, 60.09024980191904]
> Weather Conditions: ['Sunny', 'Sunny', 'Sunny', 'Snow', 'Snow', 'Sunny', 'Rain']
> Rules:
> The ideal accessory choice depends on the most common weather condition:
> If it's mostly sunny, pick 'Sunscreen'.
> If it's mostly raining, pick 'Umbrella'.
> If it's mostly snowing, pick 'Snow Boots'.
> The ideal clothing choice depends on the average temperature:
> Below 32°F: 'Warm' clothing.
> Between 32°F and 70°F: 'Medium' clothing.
> Above 70°F: 'Summer' clothing.
>
> Based on the above data and rules, what should be the ideal accessory and clothing choice? Please explain your reasoning.
> Answer:
> To determine the ideal accessory and clothing choice based on the given data and rules, we need to evaluate the most common weather condition and the average temperature.
> Determine the Most Common Weather Condition:
> Count the occurrences of each weather condition in the data. We have 'Sunny' on 4 days, 'Snow' on 2 days, and 'Rain' on 1 day.
> Therefore, the most common weather condition is 'Sunny'.
>
> Choose the Ideal Accessory:
> Based on Rule 1, since it's mostly sunny, the ideal accessory is 'Sunscreen'.
>
> Calculate the Average Temperature:
> Add up the temperatures: 47.35 - 31.62 + 25.33 - 41.85 + 13.36 + 4.58 + 60.09 = 76.34°F.
> Divide by the number of days (7) to get the average: 76.34 / 7 10.91°F.
>
> Choose the Ideal Clothing:
> Based on Rule 2, since the average temperature (10.91°F) is below 32°F, the ideal clothing is 'Warm'.
>
> In conclusion, the ideal accessory choice is 'Sunscreen', and the ideal clothing choice is 'Warm', based on the most common weather condition and the average temperature.
> Question:
> Consider the following data about weather conditions for the last 7 days:
> Temperatures (in °F): [119.69890062240508, 100.2992556597521, 62.52308069479881, 48.464537253691525, 100.3342855700459, 108.19488214881746, 103.13504138096059]
> Weather Conditions: ['Sunny', 'Rain', 'Rain', 'Snow', 'Rain', 'Rain', 'Rain']
> Rules:

The ideal accessory choice depends on the most common weather condition:
If it's mostly sunny, pick 'Sunscreen'.
If it's mostly raining, pick 'Umbrella'.
If it's mostly snowing, pick 'Snow Boots'.
The ideal clothing choice depends on the average temperature:
Below 32°F: 'Warm' clothing.
Between 32°F and 70°F: 'Medium' clothing.
Above 70°F: 'Summer' clothing.

Based on the above data and rules, what should be the ideal accessory and clothing choice? Please explain your reasoning.

Chain of Thought Prompt for Decision from Weather Environment.

The integration of few-shot learning proved to be a valuable technique, showing promising results in our project. This approach, which involves training the model with a small set of examples to rapidly adapt and learn, significantly enhanced the model's performance. The few-shot prompts, designed to provide concise yet comprehensive learning examples, are outlined as follows:

"User: You are playing a game of blackjack.
You have a hand which totals 20 .
The dealer has a hidden card and a card visible to you. The visible card is 4. Should you hit or stand? After thinking, give me either hit or stand in a new line.
Assistant:
Let's think step by step,
In this scenario, you have a very strong hand with a total of 20. The dealer's visible card is a 4, which is considered a weak card for the dealer. Here's the thought process:

1. **Assess Your Hand**: With a total of 20, you are in a very strong position. The only hand that can improve yours is an Ace, which would give you a perfect 21. However, any other card would either make no difference (if it's a 10 or a face card) or would worsen your hand by causing you to bust.
2. **Consider the Dealer's Card**: The dealer has a 4 showing, which is not a good starting card. The dealer has a higher chance of busting with a 4 as the upcard because they must hit until they reach a total of at least 17, and there are many card values that can lead to a bust.
3. **Evaluate the Risk of Busting**: If you hit, you have a very high chance of busting since you can only take an Ace without going over 21. There are only 4 Aces in a standard deck, and the likelihood of drawing one is not in your favor.
4. **Consider the Odds**: Standing on 20 is almost always the correct play. It's one of the strongest hands you can have, and the odds of the dealer beating a 20 are quite low, especially when their upcard is a 4.
5. **Basic Strategy**: Basic blackjack strategy would advise you to stand on any total of 17 or higher, regardless of what the dealer is showing. With a total of 20, you are well above that threshold.

Given these considerations, the most strategically sound decision is to stand. You want to avoid the risk of busting and give the dealer the chance to try to improve their hand, which could lead to them busting.
Stand"

Few Shot Prompt for Blackjack.

" You are a navigator bot provide context of how I want to control navigation. you output directions, Top, Bottom, Right and Left.

Look at the following data and its Answers:
Your current position: (2,3), you have to reach (1,1). Answer : left
Your current position: (2,2), you have to reach (1,1). Answer : left
Your current position: (2,1), you have to reach (1,1). Answer : up
Your current position: (1,3), you have to reach (2,0). Answer : left
Your current position: (1,2), you have to reach (2,0). Answer : left
Your current position: (1,1), you have to reach (2,0). Answer : left

Now Answer:
Your current position: (s1,s2), you have to reach (e1,e2). Answer which direction you move in. Options A) Up B) Down C) Left D) Right. "

Few Shot Prompt for Navigation.

Consider the following data about weather conditions for the last 7 days:
Temperatures (in °F): [47.348949541566256, -31.622330311862147, 25.32772017492796, -41.84648929886844, 13.360171709887624, 4.580407868630644, 60.09024980191904]
Weather Conditions: ['Sunny', 'Sunny', 'Sunny', 'Snow', 'Snow', 'Sunny', 'Rain']
Rules:
The ideal accessory choice depends on the most common weather condition:
If it's mostly sunny, pick 'Sunscreen'.
If it's mostly raining, pick 'Umbrella'.
If it's mostly snowing, pick 'Snow Boots'.
The ideal clothing choice depends on the average temperature:
Below 32°F: 'Warm' clothing.
Between 32°F and 70°F: 'Medium' clothing.
Above 70°F: 'Summer' clothing.

Based on the above data and rules, what should be the ideal accessory and clothing choice? Please explain your reasoning.

Few Shot Prompt for Decision from Weather Environment.

5 DATASET GENERATION

Utilizing the A* Algorithm, we successfully developed a data generator capable of producing 'n' number of data points. This generator is specifically designed to convert these data points into prompts suitable for both fine-tuning and few-shot learning applications. This advancement represents a significant step in enhancing the efficiency and effectiveness of our model training processes.

The dataset would look like this:

	Prompt	Answer
0	Your current position: (1,1), you have to reach (3,3). Answer which direction you move in. Options A) Up B) Down C) Left D) Right	right
1	Your current position: (1,2), you have to reach (3,3). Answer which direction you move in. Options A) Up B) Down C) Left D) Right	down
2	Your current position: (2,2), you have to reach (3,3). Answer which direction you move in. Options A) Up B) Down C) Left D) Right	right
3	Your current position: (2,3), you have to reach (3,3). Answer which direction you move in. Options A) Up B) Down C) Left D) Right	down

Figure 2: Dataset Generated using A* Algorithm.

6 RESULTS

6.1 Blackjack

6.1.1 Baseline. The RL agent developed an optimal strategy for the game based on the cards available to both the player and the dealer. Specifically, the developed strategy is the following: This

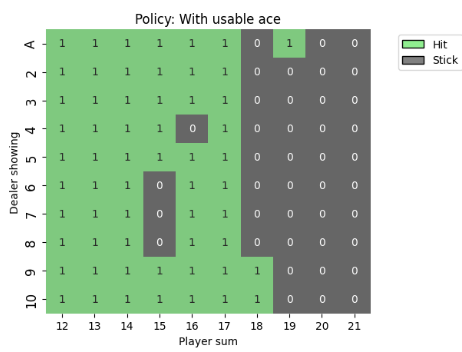

Figure 3: Strategy developed by RL agent

allowed the RL agent to reach a win ratio of around 44%, which is coherent to the theoretical most optimal strategy ratio.
The LLMs were then tested with the baseline prompts. The math in the answers was satisfactory for most LLMs, while the reasoning was sound for just the gpt models.
Furthermore, we noticed some bias in the models, either towards hitting or standing. Again, the only model who followed a reasoning similar to the RL agent strategy was gpt.

6.1.2 Chain of Thought. By applying COT prompting, the overall performance improved for both math and reasoning for almost all models.

6.1.3 Few Shot. By using Few Shot prompting, the overall evaluation for almost all models increased furthermore. For this method, we also considered the models ability to follow the instructions provided by the small examples and compared them to the overall reasoning and math capabilities.
Overall, the only models that did not show an improvement with any techniques were the lowest parameters models, i.e. Mistral and Rocket, proving to not be sufficient to perform strategy reasoning.

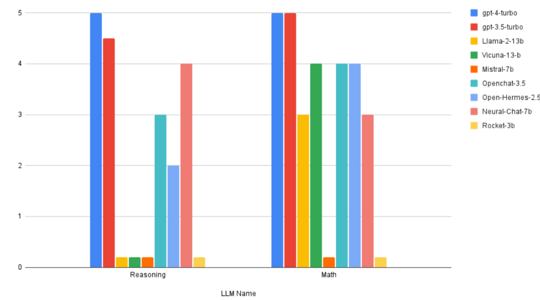

Figure 4: Baseline LLMs evaluation for Blackjack

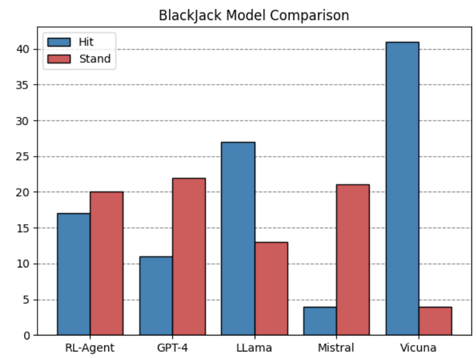

Figure 5: Frequency of 'Hit' and 'Stand' actions for different models over 25 episodes

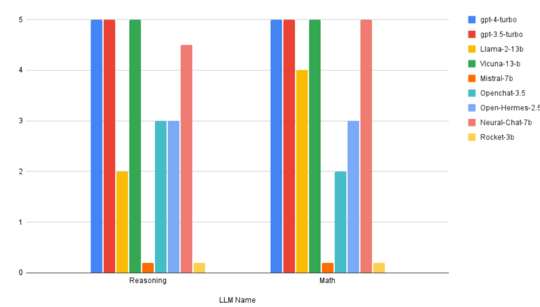

Figure 6: LLMs evaluation with Chain of Thought Prompting for Blackjack

The only exception was Neural-Chat, which surprisingly performed similarly to more complex models.

6.2 Grid Navigation

6.2.1 Baseline. For Grid Navigation game, we tested the average number of steps required by the models to reach the goal, as well as reasoning, math and prediction capabilities. The RL agent was able to reduce the average number of steps needed to complete the game in just over 100 episodes, reaching performance similar to the most optimal one dictated by the A* algorithm. The models performance tended to be inferior and not able to reach the theoretical minimum

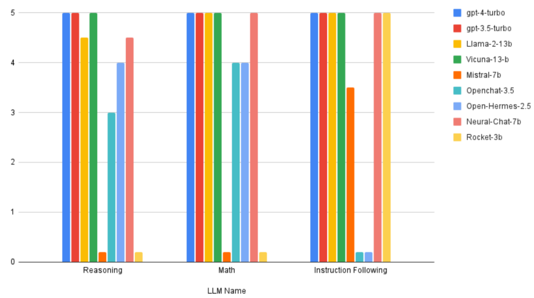

Figure 7: LLms evaluation with Few Shot Prompting for Blackjack

Model	Win	Tie/Loss
RL Agent	5	5
gpt-4.5-turbo	5	5
gpt-3.5-turbo	4	6
Vicuna 13b	4	6
Neural-Chat-7b	3	7

Table 2: Wins and Ties/Losses for different models over 10 games

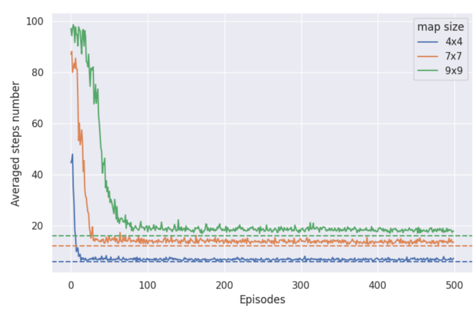

Figure 8: RL agent performance for multiple grid sizes over episodes

number of steps, with the exception of the gpt models. LLama2 and Mistral Instruct frequently confused between sending the model Down instead of sending it Up as it sometimes assumed that the down action would increment the Y coordinate. Due to this reason, they have significantly higher average steps than GPT4. We also noticed behaviour from the LLMs where they would answer the prompt but also start outputting subsequent made-up questions based our prompt and answer them in a single call.
The overall performance was in fact satisfying for only the gpt models, while the other showed to be lacking in both math and prediction capabilities.

6.2.2 Chain of Thought. Using COT technique, the performance of Vicuna improved noticeably in all metrics, while Llama started including some propert math reasoning into its answers. On the other hand, the performance of Mistral dropped in both prediction and math reasoning.

Model	Average Steps in 4x4 Matrix
A* Algorithm	6
Random Policy	45
RL Agent	6
GPT-4	6
GPT-3.5	6
LLama 2	20
Mistral Instruct	16
Vicuna	10

Table 3: Average steps needed by models to reach goal

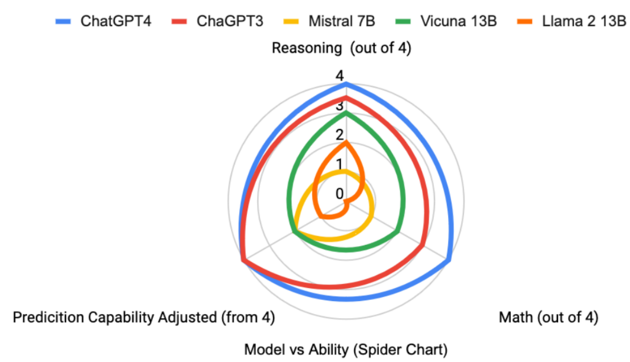

Figure 9: Baseline LLms evaluation for Grid Navigation

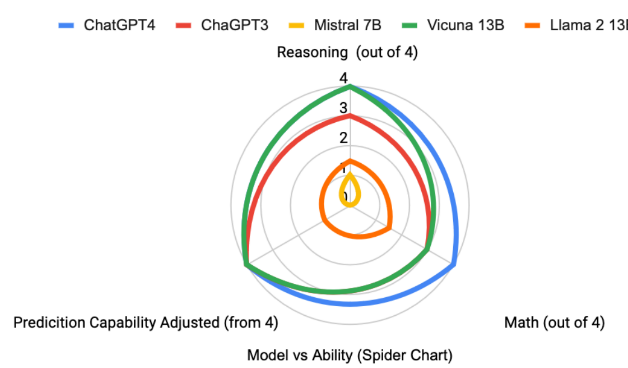

Figure 10: LLMs evaluation with Chain of Thought Prompting for Grid Navigation

6.2.3 Few Shot. A similar result to COT was obtained when we applied few shot prompting, proving again the limited spatial awareness of most lower parameters LLMs.

6.3 Decision from Weather Enviroment

6.3.1 Baseline. The trained RL agent model exhibited limitations in complexity, rendering it not able to grasp the mathematical information and decision-making requirements of the environment.

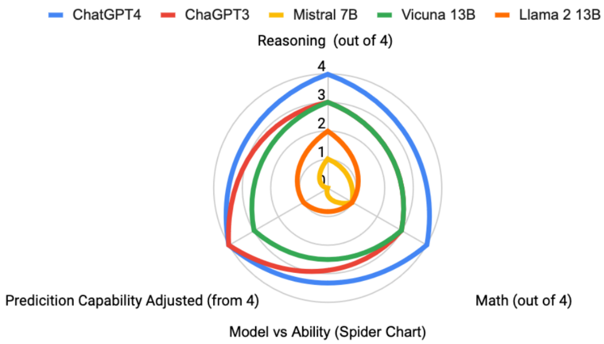

Figure 11: LLMs evaluation with Few Shot Prompting for Grid Navigation

Considering a possible maximum reward of 100 per iteration, the agent obtained an average of 35.4.

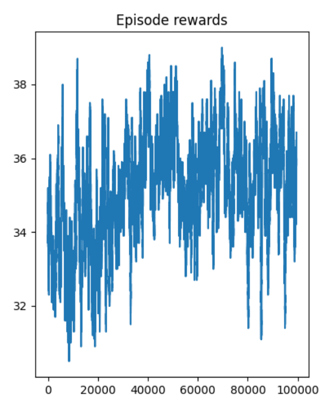

Figure 12: RL Agent average reward over episodes

The LLMS, on the other hand, proved to have a better performance than the RL agent, even in the case of lower parameters ones. Without being explicitly prompted for one, almost all models provided little to no mathematical reasoning.

6.3.2 Chain of Thought. When including COT in the prompt, the performance of gpt models increased drastically, including a sound mathematical reasoning to the answers. A similar result was obtained for LLama2, which showed improvement for almost all metrics.

6.3.3 Few Shot. When using Few Shot prompting, the gpt models performanced improved similarly to COT. Vicuna, while not showing improved mathematical reasoning managed to obtain excellent prediction capabilities.

6.4 Traffic Signal Evaluation

6.4.1 Baseline. Except for gpt4, all models provided an unsatisfactory performance in all metrics, thus showing the inability of smaller models to navigate the complexities inherent in this environment.

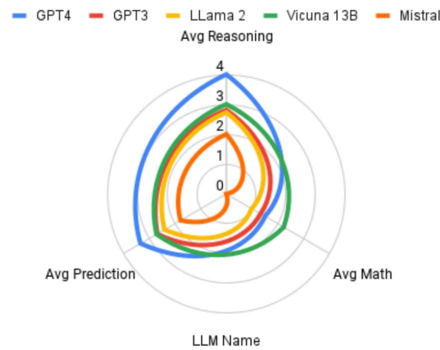

Figure 13: Baseline LLms evaluation for Weather Environment

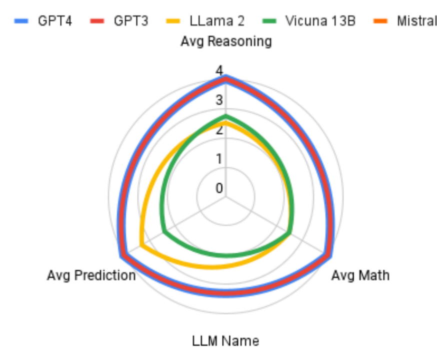

Figure 14: LLMs evaluation with Chain of Thought Prompting for Weather Environment

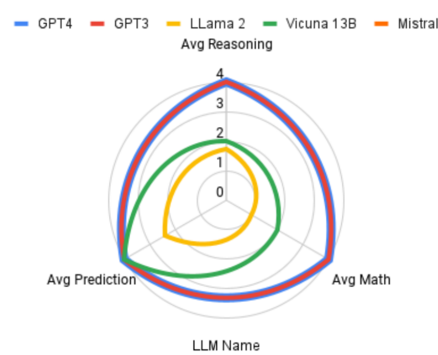

Figure 15: LLMs evaluation with Few Shot Prompting for Weather Environment

6.4.2 Chain of Thought. When performing COT, the performance for mistral slightly improved in all metrics but math reasoning. For Vicuna and LLama2, the overall reasoning improved, albeit at the expenses of losing their predictive capacity drastically. Surprisingly, the mathematical reasoning behind Vicuna improved greatly.

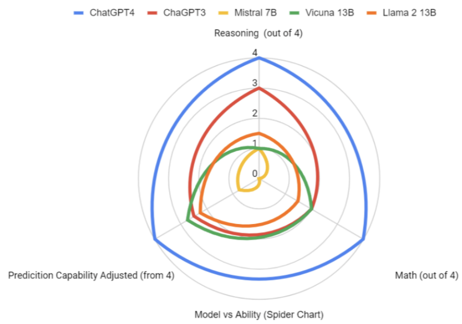

Figure 16: Baseline LLMs evaluation for Traffic Signal Evaluation

Figure 17: LLMs evaluation with Chain of Thought Prompting for Traffic Signal Evaluation

6.4.3 Few Shot. With the specific prompts utilized for Few Shot learning, the mathematical side of the problem was ignored by all models. Nonetheless, they all showed an increase reasoning and prediction, with the only expection of Mistral, which perfomed worse than the baseline.

Figure 18: LLMs evaluation with Few Shot Prompting for Traffic Signal Evaluation

7 CHALLENGES FACED

A significant challenge in our experiment was addressing the 'hallucinations'—inaccuracies and irrelevant responses—exhibited by LLMs. Since these models aren't specifically trained on game data, they frequently suggest invalid actions within the game environment. To mitigate this issue, we had to do detailed prompt engineering defining the environment and action space to the model. Another prominent challenge involved the extraction and refinement of responses from the LLMs. These models often produce varied explanations and unpredictable response formats to a given prompt. To tackle this, we developed several post-processing functions. These functions are designed to cleanse the LLM output, effectively discerning and translating the suggested actions into the action space defined by the RL gym environment.

8 RECENT DEVELOPMENTS

In the recent surge of interest in our domain, subsequent research endeavors have significantly expanded the landscape. Aghzal et al. [13] delved deeper into the evaluation of Large Language Models (LLMs) in the pathfinding task, demonstrating notable improvements through fine-tuning, aligning with our initial hypotheses. Zhou et al. ([10]) have introduced an innovative approach that integrates the planning capabilities of LLMs with Reinforcement Learning (RL). Their hierarchical agent employs LLMs to encode world knowledge, guiding a high-level policy in addressing long-horizon tasks. Wu et al. [11] contributed to the field by introducing SMARTPLAY, an Evaluation Benchmark designed for assessing LLMs in game reasoning. This automated pipeline encompasses diverse game environments, including rock-paper-scissors, tower of hanoi, and two-arm bandits. Concurrently, novel prompting methods have emerged, showing promise in enhancing results without resorting to fine-tuning. Sel et al. [9] introduced the Algorithm of Thoughts (AoT) prompting, encouraging algorithmic thinking in models to circumvent the iterative process of stopping, modifying prompts, and rerunning, as noted in [12] and [1].

These recent developments not only enrich the current landscape but also serve as inspiration for us to refine and elevate our work, aiming to achieve superior results in our ongoing research endeavors.

9 CONCLUSIONS AND FUTURE WORKS

The evaluation of Large Language Models (LLMs) in game environments serves as a valuable proxy for assessing their reasoning and decision-making capabilities. Consequently, it presents a robust methodology for the same. Our experiments reveal a positive correlation between the number of parameters in the base model and its reasoning proficiency, indicating that larger models tend to exhibit superior abilities. Furthermore, the performance of the base model is amenable to improvement through the strategic utilization of effective prompting techniques. To address existing performance challenges in LLMs concerning reasoning tasks, we intend to conduct experiments incorporating novel prompting techniques, as detailed in Section 8. Additionally, we aim to assess the models post-fine-tuning, specifically tailored for the targeted task. Expanding our scope, we plan to investigate the capabilities of these models across a greater diversity of games. In pursuit of more objective evaluations, we aspire to develop a benchmarking environment for game reasoning that moves away from traditional human-centric evaluations, paving the way for a more comprehensive and automated assessment framework.

REFERENCES

[1] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687* (2023).

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[3] Jiang et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[4] Wang et al. 2023. OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. *arXiv preprint arXiv:2309.11235* (2023).

[5] Farama Foundation. 2017. Gymnasium Library. (2017). gymnasium.farama.org/

[6] LMstudio. 2022. lmstudio. (2022). https://lmstudio.ai/

[7] lmsys. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90 perc. ChatGPT Quality. (2023). https://lmsys.org/blog/2023-03-30-vicuna/

[8] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* abs/2303.08774 (2023). https://api.semanticscholar.org/CorpusID:257532815

[9] Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Lu Wang, Ruoxi Jia, and Ming Jin. 2023. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379* (2023).

[10] Arjun Vaithilingam Sudhakar, Prasanna Parthasarathi, Janarthanan Rajendran, and Sarath Chandar. 2023. Language Model-In-The-Loop: Data Optimal Approach to Learn-To-Recommend Actions in Text Games. *arXiv preprint arXiv:2311.07687* (2023).

[11] Yue Wu, Xuan Tang, Tom M Mitchell, and Yuanzhi Li. 2023. SmartPlay: A Benchmark for LLMs as Intelligent Agents. *arXiv preprint arXiv:2310.01557* (2023).

[12] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601* (2023).

[13] Zihao Zhou, Bin Hu, Pu Zhang, Chenyang Zhao, and Bin Liu. 2023. Large Language Model is a Good Policy Teacher for Training Reinforcement Learning Agents. *arXiv preprint arXiv:2311.13373* (2023).