I am the upgrade: Surpassing RL Agents on Chef's Hat with a Multi-step Reasoning LLM Based Agent

1st Alexandre Pereira
University of Pernambuco
Recife, Brazil
armp@ecomp.poli.br

2nd Bruno Fernandes
University of Pernambuco
Recife, Brazil
bjtf@ecomp.poli.br

3rd Pablo Barros
University of Pernambuco
Recife, Brazil
pablovin@gmail.com

***Abstract*—This study investigates the performance of Llama-3-8B, an open-source large language model (LLM), acting as a player in the competitive environment of the Chef's Hat card game, comparing its results and behavior with traditional reinforcement learning (RL) agents. While RL techniques like Deep Q-Learning (DQL) and Proximal Policy Optimization (PPO) have shown competitive behavior when playing the game, LLMs offer an intriguing alternative due to their ability to generalize across tasks. We demonstrate that they can play at a similar level of skill. The focus of this work is to explore the potential of Llama-3-8B, which, despite being a smaller model, performs competitively as an autonomous player in the game by utilizing basic prompts built on the game rules. Two agents based on Llama-3-8B are developed: Heuristic Llama, which follows predefined strategies, and Global Llama, which creates its own strategies through Chain-of-Thought (CoT). We assess the agents' ability to adapt and demonstrate that they outperform traditional RL agents, including DQL and PPO, and rival an heuristic agent based on a simple yet effective strategy, Larger Value. The study demonstrates that Llama-based agents can generate competitive strategies and exhibit promising generalization capabilities, suggesting the potential for LLMs to complement or even surpass traditional RL in specific domains.**

***Index Terms*—artificial intelligence, large language models, game playing, human-robot interaction**

I. Introduction

Artificial intelligence has increasingly found applications in gaming, where strategy and decision-making are essential for success. Reinforcement learning (RL) agents have shown strong performance in this area over the years [1]. However, the rise of large language models (LLMs) introduces new possibilities for enhancing the gaming experience. Their reasoning and generalization skill allow them to adapt dynamically to complex and competitive environments, such as Chef's Hat, being able to act from interactive NPCs to playing agents [2].

Chef's Hat is a pizzeria-themed multiplayer card game designed for human-robot interaction (HRI) scenarios [3]. It features 66 cards, numbered 1 to 11, and two special Joker cards. Players take turns discarding cards of equal or lower quantity and lower value than those on the board. The game ends when all players discard their cards, and scoring is based on the order in which they do so, earning from 0 to 3 points and influencing advantages in future rounds. The game supports RL-based and generic agents [4], [5].

Among the agents implemented in the game, four stand out due to their capabilities and differences in playstyle: DQL, PPO, Larger Value, and Naive Llama. DQL and PPO are the two highest scoring RL-based agents in Chef's Hat [6]. Deep Q-Learning uses deep neural networks to estimate action values for decision-making [7], while Proximal Policy Optimization refines a policy with controlled updates to ensure stability [8]. Larger Value is a heuristic agent that discards the highest-value cards first, achieving the best scores in the game.

Naive Llama, an LLM-based agent using Llama-3-8B (an open-source 8 billion-parameter model [9]), was presented in our previous paper [10] and yields performance similar to the RL agents; however, it still performs worse than the PPO and Larger Value agents. As no context beyond the board and its current cards is provided to the agent, it struggles to adapt strategies based on the game progression.

This study aims to evaluate the performance of Llama-3-8B as a chef's hat player when provided with a more comprehensive game context and enhanced with Chain-of-Thought (CoT) [11] to aid its reasoning. To evaluate our agents, we have put them to play in sets of 100 matches under predetermined conditions. Heuristic Llama played in four sets using different strategies, and Global Llama played in five sets against different groups of opponents.

The LLM's ability to follow and create strategies has shown overly positive success. While Heuristic Llama was not able to grasp some of the proposed instructions, Global Llama attained scores higher than the RL agents and rivaled Larger Value in both performance and strategy. This indicates that despite the LLM's limitations, it is still capable of being competitive and devising strategies that can surpass RL agents in games.

II. Related Works

In this section we discuss studies related to the development and evaluation of our agents. These works explore modular architectures to address LLM limitations and allow autonomous decision-making, as well as one of our previous works, where we have developed a simple but competitive LLM-based agent for Chef's Hat.

Xi et al. [12] examined the potential of LLMs for developing intelligent agents. They proposed an architecture divided into three modules: brain, perception, and action. This implementation mitigates limitations like catastrophic forgetting and poor task understanding. However, the computational cost might be

too high for simpler applications, especially when scalability is needed.

Wang et al. [13] reviewed autonomous agents based on LLMs, focusing on their construction, applications, and evaluation methods. They introduced an architecture with four modules: profile, memory, planning, and action. This model enables agents to make context-driven decisions based on previous experiences, and be suitable for many tasks, from the simpler to the complex ones.

Pereira et al. [10] evaluated Llama-3-8B as an agent on Chef's Hat, comparing its performance to the traditional RL agents DQL and PPO. The study found that the LLM agent performed similarly or better than the RL agents, making strategic decisions without additional training. The results highlight the potential of LLMs in competitive environments and suggest future exploration of hybrid agent models and improved instructions.

III. Methodology

This section describes the methodology for developing and evaluating LLM-based agents in the Chef's Hat game. It covers the agent architecture, detailing the modules adapted for the game, as well as the experimental methodology used to assess the agents' ability to follow and create strategies, and the setup for the experiments.

A. Agent Architecture

The generic architecture used to develop and experiment on the agents throughout the study followed the model proposed by Wang et al. [13], which is divided in four modules: profile, memory, planning and action. They were adapted to the Chef's Hat environment [14], each containing prompts and, in some cases, mathematical functions, and combined to enable complex reasoning.

The profile module contextualizes the agent by providing essential game information. It consists of a single prompt based on Pereira et al. [10], with additional clarification about the rarity of the card. Fig. 1 shows the prompt, which includes details such as game objectives, card values, distribution, rarity, participant count, rules for discarding cards, and turn-passing consequences.

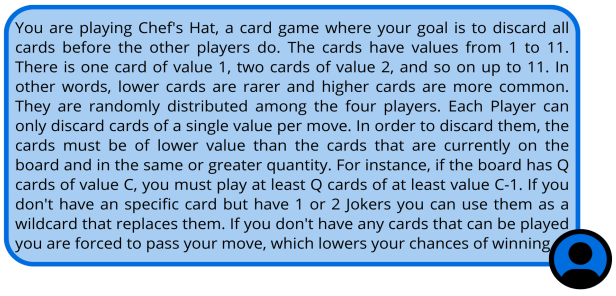

Fig. 1. The Profile module's prompt.

The memory module stores data from all four players, including previous moves, card counts, scores, and rankings. Fig. 2 presents an example during a match. The prompt dedicates a paragraph to each player: the first covers the agent's data, while the others feature data from opponents, labeled with capital letters for consistency and to prevent output changes [15].

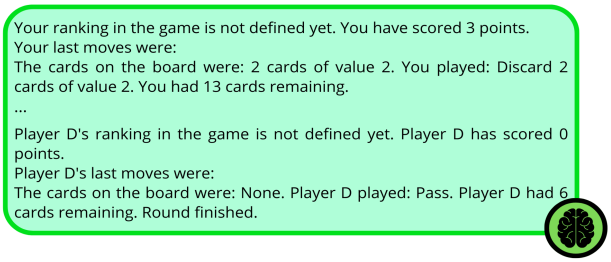

Fig. 2. A simplified memory module's prompt extracted from a match.

The planning module allows the agent to develop its strategies by combining the prompts from the earlier modules with three new ones, split into two CoT steps. The first prompt introduces the game state, while the second (Q1) and third (Q2) guide the agent to choose the rarity and quantity of the cards it wants to discard, respectively, as shown in Fig. 3. In order to speed up the LLM's response, brackets ("[") are added at the start of each completion to a multiple-choice question, encouraging single-token answers. [16].

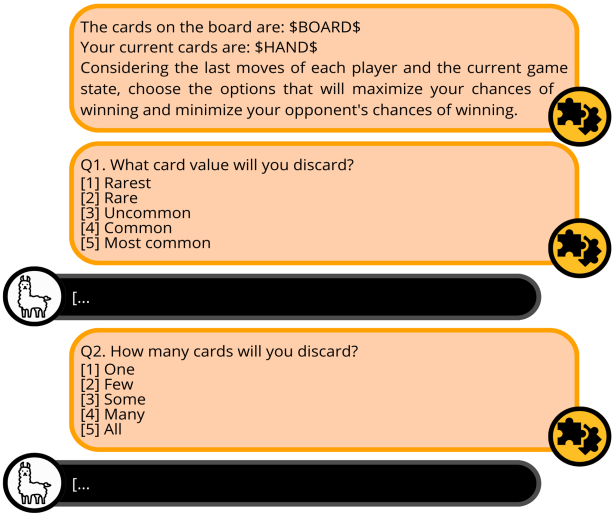

Fig. 3. Planning module's prompts illustrated as a chat. Each square bracket represents a step of the CoT where the LLM will make a choice.

The action module is responsible for translating the agent's decision into an actual game action. We have implemented two variations of this module, as shown on Fig. 4. The LLM-assisted action (a) forces the model to follow a predefined strategy when choosing an action, having its marker "$HEURISTIC$" replaced by the desired strategy. On the other hand, the function-assisted action (b) uses a mathematical function to choose the agent's moves. It takes into account the outputs of the planning module and ensures that the strategy is executed correctly.

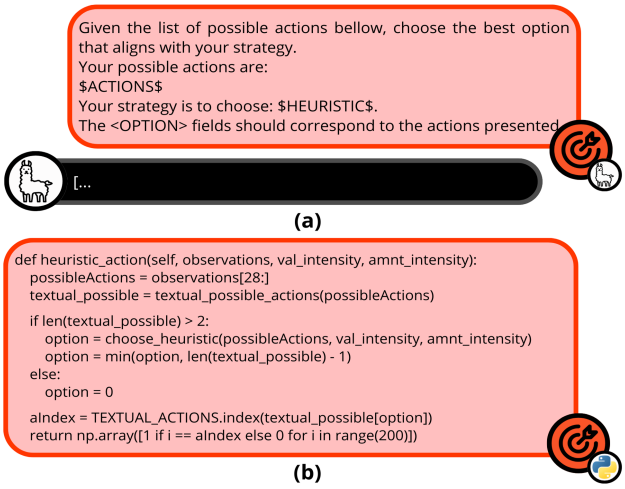

Fig. 4. The two variations of the action module. (a) LLM-assisted action presented as a chat. (b) Function-assisted action implemented in Python.

B. Experimental Methodology

Developing an agent that can both plan and follow a strategy is a complex task, so we divided our study into two agents: Heuristic Llama and Global Llama. This allowed us to evaluate whether Llama-3-8B could follow and develop a strategy, respectively. This division was necessary because, to surpass RL opponents, the agent needed to create and execute a successful strategy, which are two bottlenecks. By splitting our focus, we conducted experiments with greater accuracy.

The Heuristic Llama was used as an intermediary agent to determine whether the LLM was able to follow a given strategy successfully or not. As shown on Fig. 5, its architecture is simpler, containing only two of the four proposed modules: profile and LLM-assisted action. This is intended because this agent's only goal is to follow a given strategy, so no previous information or reasoning is required.

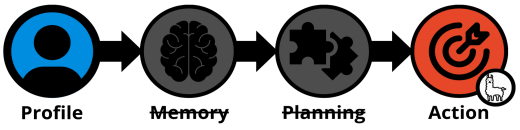

Fig. 5. Heuristic Llama's architecture with the profile and action modules.

The four strategies used to evaluate Heuristic Llama's capabilities are displayed in Table I. These simple heuristics explore extreme decisions for discarding cards based on value and quantity. For each strategy, 100 games were played using Random agents as opponents. Each action was scored on a 0-1 scale, depending on how closely it matched the ideal move defined by the heuristic, offering a detailed evaluation of strategy compliance.

The Global Llama was the study's goal agent. Its architecture, represented in Fig. 6, comprises four modules with a function-assisted action. This agent was designed to evaluate Llama-3-8b's ability to develop strategies capable of outperforming opponents, potentially surpassing RL and even other Chef's Hat agents.

TABLE I
STRATEGIES USED TO EVALUATE THE HEURISTIC LLAMA

Strategy Text ($HEURISTIC$)	Quantity	Value
discard <LOWEST> cards of value <LOWEST>	min	min
discard <HIGHEST> cards of value <LOWEST>	max	min
discard <LOWEST> cards of value <HIGHEST>	min	max
discard <HIGHEST> cards of value <HIGHEST>	max	max

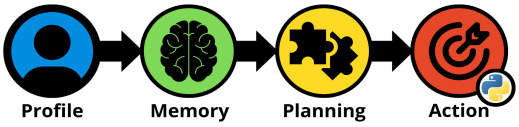

Fig. 6. Global Llama's architecture with all modules.

Similar to Heuristic Llama, Global Llama was also tested against other agents. However, this evaluation focused on its game performance rather than strategy adherence. As shown in Table II, the tests involved five groups of opponents, each comprising 100 matches, with three different agents per group. In addition to Random agents, Global Llama also faced DQL, PPO and Larger Value. To ensure adaptability, the agent's memory was reset after each match, and agent positions were rotated every five matches to maintain fairness and eliminate biases from prior performance or turn order.

TABLE II
SCENARIOS USED TO EVALUATE THE GLOBAL LLAMA

Opponents	Target Opponent
3x Random	highest score
1x DQL and 2x Random	DQL
1x PPO and 2x Random	PPO
1x Larger Value and 2x Random	Larger Value
1x Larger Value, 1x PPO and 1x DQL	highest score

IV. RESULTS AND DISCUSSION

A. Strategy following analysis: Heuristic Llama

The Heuristic Llama agent was able to follow the correct strategy with an average accuracy of 71.85% for the four different scenarios. Despite being a seemingly good result, when breaking the analysis down for each individual strategy, it becomes clear that Llama-3-8b is not suited to follow some of them correctly.

As shown in Fig. 7, the model performed better in strategies that discarded cards with the same intensity for quantity and value, and failed to comprehend conflicting strategies despite the prompt engineering efforts. This behavior happens due to the LLM's poor capability of understanding sequential instructions [15], which may also be aggravated because of its relatively low number of parameters.

As the Heuristic Llama failed to follow a given strategy, it became evident that the Global Llama agent would likely

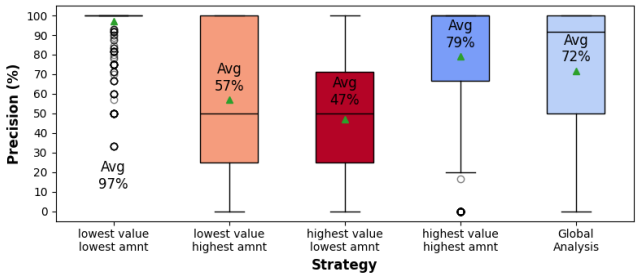

Fig. 7. Box plot of Heuristic Llama's precision in following strategies.

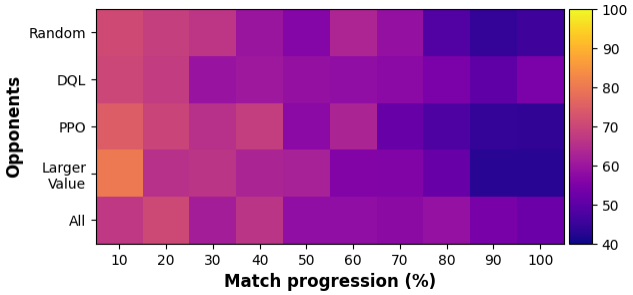

Fig. 8. Heatmap of the average card values discarded by Global Llama throughout the games in each scenario.

face similar difficulties in executing its own strategies, compromising any meaningful performance evaluation in Chef's Hat. To address this, the function-assisted action module was adopted instead of the LLM-assisted counterpart, proving to be a promising alternative.

B. Strategy creation analysis: Global Llama

Global Llama was able to score higher than both of the most competitive RL agents implemented in the game: DQL and PPO, as evidenced on Table III. It also nearly surpassed Chef's Hat highest scoring agent, Larger Value, missing only by two points, a statistically insignificant result (less than 1%).

TABLE III
GLOBAL LLAMA RESULTS

Opponents	Global Llama	Target Opponent
3 Random	292	112
DQL and 2 Random	292	146
PPO and 2 Random	283	187
Larger Value and 2 Random	245	247
Larger Value, PPO and DQL	216	245

Although the agent performed well in the first four scenarios, it faced challenges in the final one, where it competed against multiple non-random agents. This difficulty may be attributed to both, Llama-3-8b limitations and the low number of past actions supplied by the memory module (10 for each agent) due to our computational limitations.

The agent's strategies can be studied by analyzing the card values and quantities discarded throughout the matches. Figure 8 illustrates the average card values discarded across 100 games for each of the five scenarios, broken down into 10 time slots from start to finish. Brighter colors represent instances where the agent discarded the highest value cards it had (more common), while darker colors indicate the selection of the lowest value cards (rarer).

In general, Global Llama followed a consistent discard strategy, prioritizing the highest value cards first before moving to the lower value ones, with some variations. This approach was most pronounced against the PPO and Larger Value agents, although it was also observed, to a lesser degree, against the Random and DQL agents.

A possible explanation for the weaker strategy is the relatively poor performance of those opponents compared to LLM-based agents [10]. However, the pattern was also less evident in the final scenario, confirming that Global Llama struggled to perform effectively against a more capable group of opponents, as discussed earlier.

Regarding the amount of cards discarded, the agent showed little variation across matches, rounds, and scenarios, consistently opting to discard the maximum possible amount of each card, with an average of 99.62%. When combined with its discard value strategy, it becomes clear that the agent attempts to mimic unintentionally the strategy of the Larger Value agent, while keeping its actions more flexible and adaptable.

V. CONCLUSION

This study has provided a detailed evaluation of Llama-3-8b abilities on following and developing strategies on Chef's Hat. Heuristic Llama, our intermediate agent, has demonstrated that despite Llama-3-8b being a very competitive and capable model, as shown with the Naive Llama agent, it is not able to correctly follow some of the simple heuristics proposed, especially the conflicting ones, which highlights that the LLM may not be suitable for navigating some gaming scenarios of greater or similar complexity.

By analyzing the model's deficiencies, we developed the Global Llama agent. Although this agent does not rely on the model to play its moves, its strategy was crafted with insights from the LLM, enhanced by a memory module that considers past decisions, and further refined through a Chain-of-Thought approach on the planning module. This agent successfully outperformed both RL agents it was tested against and even rivaled Larger Value, the best Chef's Hat agent.

The results indicate that despite some flaws, when prompted with in-game context and breaking convoluted reasoning in smaller steps, a lightweight model such as Llama-3-8b is able to provide results superior to those of RL agents in human-robot-interaction scenarios, which may extend to other areas, even beyond gaming.

It is important to study how LLMs adapt to those scenarios. Future works should include deeper studies about how game information impacts LLM based agent responses, investigate multiple games and LLMs, analyze how visual large language models react, and test other technologies besides LLMs, such as embeddings.

REFERENCES

[1] K. Souchleris, G. K. Sidiropoulos and G. A. Papakostas, "Reinforcement Learning in Game Industry—Review, Prospects and Challenges," MDPI 2023, February 2023.

[2] R. Gallotta et al. "Large Language Models and Games: A Survey and Roadmap," IEEE Transactions on Games 2024, September 2024.

[3] P. Barros et al. "It's food fight! designing the chef's hat card game for affective-aware hri," HRI 2021, March 2021.

[4] P. Barros et al. "You were always on my mind: introducing chef's hat and copper for personalized reinforcement learning," Front. Robot. AI, July 2021.

[5] P. Barros, "ChefsHatGYM," GitHub. [Online]. Available: https://github.com/pablovin/ChefsHatGYM.

[6] P. Barros, A. Tanevska and A. Sciutti, "Learning from Learners: Adapting Reinforcement Learning Agents to be Competitive in a Card Game," ICPR 2020, January 2021.

[7] Z. Yang, Y. Xie and Z. Wang, "A Theoretical Analysis of Deep Q-Learning," L4DC 2019, January 2019.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, July 2017.

[9] Meta, "Introducing meta llama 3: the most capable openly available LLM to date," unpublished.

[10] A. R. M. Pereira, B. Fernandes, P. Barros, "There's no Human in Charge: Playing Chef's Hat with a Large Language Model Based Agent," ACII 2024, July 2024.

[11] J. Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," NeurIPS 2022, November 2022.

[12] Z. Xi et al. "the rise and potential of large language model based agents: A Survey," unpublished.

[13] L. Wang et al. "Survey on large language model based autonomous agents," Frontiers of Computer Science, December 2024.

[14] P. Barros et al. "The chef's hat simulation environment for reinforcement-learning-based agents," unpublished.

[15] X. Chen, R. A. Chi, X. Wang and D. Zhou," Premise order matters in reasoning with large language models," ICML 2024, February 2024.

[16] J. Robinson, C. M. Rytting and D. Wingate "Leveraging large language models for multiple choice question answering," ICLR 2023, February 2023.