



UNO Arena for Evaluating Sequential Decision-Making Capability of Large Language Models

Anonymous ACL submission

Abstract

Sequential decision-making refers to algorithms that take into account the dynamics of the environment, where early decisions affect subsequent decisions. With large language models (LLMs) demonstrating powerful capability between tasks, we can't help but ask: *Can Current LLMs Effectively Make Sequential Decisions?* In order to answer this question, we propose the UNO Arena based on the card game UNO to evaluate the sequential decision-making capability of LLMs and explain in detail why we choose UNO. In UNO Arena, We evaluate the sequential decision-making capability of LLMs dynamically with novel metrics based Monte Carlo methods. We set up random players, DQN-based reinforcement learning players, and LLM players (e.g. GPT-4, Gemini-pro) for comparison testing. Furthermore, in order to improve the sequential decision-making capability of LLMs, we propose the **TUTRI** player, which can involves having LLMs reflect their own actions with the summary of game history and the game strategy. Numerous experiments demonstrate that the **TUTRI** player achieves a notable breakthrough in the performance of sequential decision-making compared to the vanilla LLM player.

1 Introduction

In artificial intelligence, sequential decision-making refers to algorithms that take the dynamics of the world into consideration (Frankish and Ramsey, 2014), and it can be described as a procedural approach to decision-making, or as a step by step decision theory. Sequential decision-making has as a consequence the intertemporal choice problem, where earlier decisions influences the later available choices (Amir, 2014).

In recent years, Large language models (LLMs) are gaining increasing popularity in both academia and industry, owing to their unprecedented performances in various applications (Chang et al., 2023),

ranging from chatbots to medical diagnoses (Wang et al., 2023a) to robotics (He et al., 2022). From robots handle complex tasks (Amiri et al., 2020) to entrepreneurial action (McMullen, 2015), sequential decision-making permeates diverse domains. Hence, an interesting question arises: *Can Current LLMs Effectively Make Sequential Decisions?*

To answer this question, we need to design a benchmark to evaluate the sequential decision-making ability of LLMs. However, evaluating LLMs' abilities is not trivial. Many works have been proposed to test LLMs' performances on either a large-scale static benchmark such as MMLU (Hendrycks et al., 2021), or with A/B tests judged by humans (Ganguli et al., 2023). One common and evident limitation of these methods, however, is that the environment for LLMs to be tested is static (Aiyappa et al., 2023; Zhou et al., 2023), which can not reflect the domino effect in sequential decision-making. Besides, data contamination (Sainz et al., 2023; Zeng et al., 2024; Xu et al., 2024), which means the inclusion of test data examples and labels in the pre-training data, also challenges the efficacy of these static benchmarks in differentiating model capabilities.

Unlike static evaluation, dynamic evaluation by treating LLMs as game-playing agents attracted more and more attention of researchers recently, such as beauty contests and private-value second price auctions (Guo et al., 2024a), Werewolf (Xu et al., 2023), Avalon (Wang et al., 2023b; Light et al., 2023), Leduc Hold'em (Guo et al., 2023). However, current attempts do not account for sequential decision-making, and these games are either challenging to evaluate for intermediate results (such as Werewolf) or have too few decision points per round (such as Leduc Hold'em). Meanwhile, we should also note that studies of dynamically evaluating sequential decision-making capability in reinforcement learning, such as games like Go (Silver et al., 2017), Dou Di Zhu (You et al., 2019),

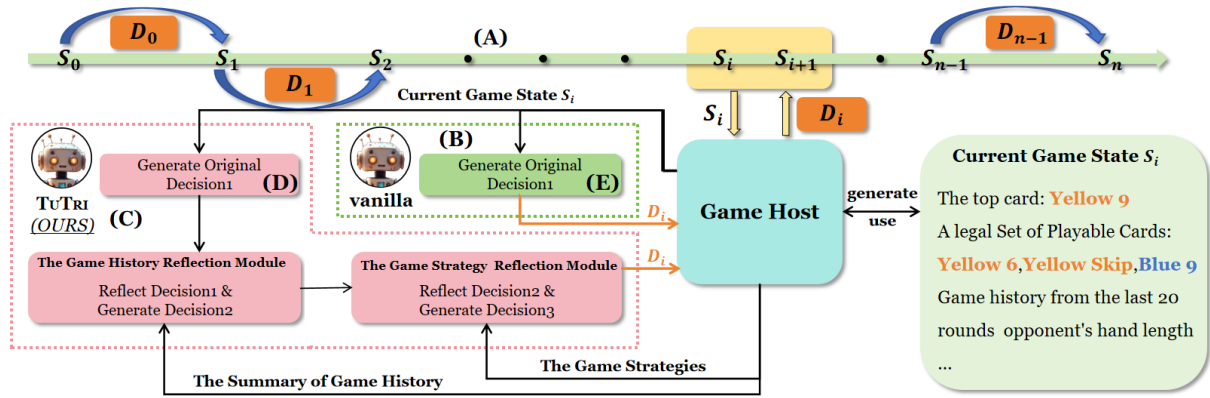


Figure 1: In this figure, (A) demonstrates the sequential decision-making process in UNO Arena, (B) shows the execution process of the vanilla LLM player, and (C) shows the execution process of the TUTRI player. In fact, The Module (D) and the Module (E) are completely identical.

and Mahjong (Li et al., 2020). However, these games present an excessively large action space. For instance, in Dou Di Zhu, players can use any combination of their cards each round, posing significant challenges for current LLMs (Zhai et al., 2024).

Considering the above aspects, we make the following efforts in this paper:

First, we build UNO Arena to dynamically evaluate the sequential decision-making capability of current LLMs. In UNO arena, we allow LLMs to participate as players in the UNO game¹, aiming to play all the cards in their hand as quickly as possible. Compared to games like Leduc Hold'em, which have fewer moves per game, UNO features an average of dozens of moves per game, making it an ideal testbed for sequential decision-making (Pfann, 2021). Additionally, unlike common games in reinforcement learning, legal actions in the UNO Arena are limited, only including drawing cards, playing cards, selecting colors to convert, and choosing whether to challenge the wild draw four card. Furthermore, to monitor the behaviours of LLMs in the UNO arena, we propose some real-time quantitative evaluation metrics by leveraging the Monte Carlo method (Kroese et al., 2014) as the reference, which provide a window to observe the intermediate results and various phenomena (like domino effect) in LLMs' sequential decision-making.

Second, based on the proposed UNO Arena, we set up a family of strong and representative players. In detail, we first build the random player, which makes decisions based on chance rather

than a specific, consistent plan, without considering the game's current state or potential outcomes. Then, we implement the reinforcement learning based player, which leverages DQN (Mnih et al., 2013) to develop sophisticated strategies for playing UNO. Finally, to probing the capability of LLMs in sequential decision-making, we provides the task description and then prompt LLMs, like GPT-4 (Achiam et al., 2023) and Gemini-pro (Team et al., 2023), to generate their reasoning steps that lead to the final action.

Third, to unleash the fully potential capability of LLMs in sequential decision-making, we propose the TUTRI player with reflection mechanism (Shinn et al., 2024), which can involves having LLMs analyze their own actions with the game history and the game strategy. In detail, the proposed agent framework consists of two key reflection modules: the game history reflection module and the game strategy reflection module. In the game history reflection module, we provide the statistical data of game history and then prompt the LLMs to rethink their decision, which simulates the process of card memorization by humans when playing UNO. In the game strategy reflection module, LLMs further take into account the game strategy, like saving wild draw four, and proceed to make the final decision, which simulates the use and adherence to strategies by humans when playing UNO.

In the experiment, we comprehensively evaluate some mainstream LLMs' ability of sequential decision-making, including GPT-3.5 (OpenAI, 2022), GPT-4 (Achiam et al., 2023), Gemini-pro (Team et al., 2023), Llama 2 (Touvron et al., 2023), and ChatGLM3 (Du et al., 2021; Zeng

¹[https://en.wikipedia.org/wiki/Uno_\(card_game\)](https://en.wikipedia.org/wiki/Uno_(card_game))

| | | | |
|-----|--|--|-----|
| 153 | et al., 2022). Our experiments show that among | and translation agents. LLM-Vectorizer (Taneja | 203 |
| 154 | these LLMs, GPT-4 is the most effective sequential | et al., 2024) uses multiple agents to generate vec- | 204 |
| 155 | decision-maker. | torized code by leveraging large language mod- | 205 |
| 156 | In summary, our contributions are as follows: | els and test-based feedback. We tailored a special | 206 |
| 157 | • We propose a dynamic evaluation method | framework for UNO, featuring self-refinement and | 207 |
| 158 | named UNO Arena for assessing the sequen- | iterative thinking. | 208 |
| 159 | tial decision-making capability of large lan- | Sequential Decision-Making Capability: Sequen- | 209 |
| 160 | guage models (LLMs) based on the card game | tial decision-making refers to the process of making | 210 |
| 161 | UNO. This method supports the evaluation | a series of decisions over time, where each decision | 211 |
| 162 | of 2-10 LLM players, reinforcement learning | may impact future choices and outcomes (Amir, | 212 |
| 163 | players, or random players engaged in a single | 2014). Though certain algorithms or reinforce | 213 |
| 164 | UNO game. | learning provide solutions for some sequential | 214 |
| 165 | • We introduce multiple unique evaluation met- | decision-making problems (Littman, 1996), LLM- | 215 |
| 166 | rics based on the Monte Carlo method for | based sequential decision-making are only em- | 216 |
| 167 | evaluating the sequential decision-making ca- | ployed in limited field like recommendation (Wang | 217 |
| 168 | pabilities of players in UNO Arena. | et al., 2023c). In our work, we utilized UNO, which | 218 |
| 169 | • To improve the sequential decision-making | is not an easy one even for human (Demaine et al., | 219 |
| 170 | capabilities of LLMs and enhance their per- | 2014), to explore the sequential decision-making | 220 |
| 171 | formance in the highly dynamic and complex | ability of LLMs. With certain methods like integrat- | 221 |
| 172 | UNO game, we have developed the TUTRI | ing past experiences and expert advice or demon- | 222 |
| 173 | player and compared it horizontally with the | strations (Chen et al., 2023), we made efforts to | 223 |
| 174 | vanilla LLM player. | maximally leverage the decision making ability as | 224 |
| 175 | | possible in a sequential manner. | 225 |
| 176 | 2 Related Work | | 226 |
| 177 | Evaluate LLMs Dynamically with Game: LLMs | 3 The UNO Arena | 227 |
| 178 | has presented increasingly emerging ability on | In this section, we first provide a brief overview | 228 |
| 179 | game-playing (Brookins and DeBacker, 2023 ; | of the version of UNO we adopt in the subsection | 229 |
| 180 | Akata et al., 2023) in recent development and iter- | §3.1. Then, we present the four different types of | 230 |
| 181 | tions. Wang et al. (2023b) use the Avalon, which | players in the UNO arena in the subsection §3.2. | 231 |
| 182 | contains elements of deception, to evaluate the ca- | Next, we detail how to use Monte Carlo methods to | 232 |
| 183 | pability of LLMs to recognize and handle decep- | determine whether a player has made an optimal de- | 233 |
| 184 | tive information. Gong et al. (2023) leverage the | cision in subsection §3.3. In the end, we introduce | 234 |
| 185 | CuisineWorld and Minecraft to assess the planning | our evaluation metrics in subsection §3.4. | |
| 186 | and emergency cooperation capabilities of LLMs. | 3.1 The UNO Game | 235 |
| 187 | Guo et al. (2024a) employ beauty contests and auc- | We select the UNO as the foundation within our | 236 |
| 188 | tion games to evaluate the rationality, strategic rea- | arena due to its widespread popularity, simplicity | 237 |
| 189 | soning capability, and adherence to instructions | and mathematical value. There are various versions | 238 |
| 190 | of LLMs. Xu et al. (2023) use the game Were- | of the UNO game. In this section, we briefly in- | 239 |
| 191 | wolf to evaluate the capability of LLMs to infer | troduce the rules of the version we adopt in this | 240 |
| 192 | player roles. Despite evaluating with game be- | work. | 241 |
| 193 | coming a popular trend, exploring into sequential | UNO Cards: A deck of UNO cards comprises | 242 |
| 194 | decision-making capability is still of scarcity in | a total of 108 cards. UNO cards are divided into | 243 |
| 195 | current works. | three types: number cards, function cards, and wild | 244 |
| 196 | Development of Agent Framework: LLM agents | cards. A number card is composed of a color (Red, | 245 |
| 197 | have been perceived as a promising way to realiz- | Blue, Yellow and Green) and a number (ranging | 246 |
| 198 | ing Artificial General Intelligence(AGI) (Xi et al., | from 0 to 9). A function card is composed of a | 247 |
| 199 | 2023) and recently have shown emergent abili- | color (Red, Blue, Yellow and Green) and a function | 248 |
| 200 | ties to execute various tasks in complex environ- | (Skip, Reverse, Draw Two). The wild cards has no | 249 |
| 201 | ment (Wei et al., 2022). SiLLM (Guo et al., 2024b) | color and is only composed of Wild cards and Wild | 250 |
| 202 | merges large language models with synchronous | Draw Four cards. The effects of the function cards | 251 |
| | machine translation, using policy decision agents | | |

and wild cards are shown in the Table 1.

UNO Process: First, deal each player 7 initial cards in clockwise order, then continue drawing cards until a number card is drawn and set as the top card of the initial discard pile. All players take rounds playing cards in clockwise order(it will be reversed by a reverse card) until a player runs out of his cards or the draw pile is exhausted, signaling the end of the game.

UNO Action: From the beginning to the end of the game, players continuously take actions in UNO. In our work, UNO includes the following types of actions:

- **Select Card:** When a player comes his playing round, they need to play a card that matches the color, number, or function of the top card in the discard pile, or play a Wild card. If they don't have a card to play, they must draw one card.
- **Select Color:** After a player plays a Wild card or a Wild Draw Four card, they need to change the color of the current top card to one of Red, Yellow, Blue or Green.
- **Select ChallengeFlag:** After a player's previous opponent plays a Wild Draw Four card, the player needs to decide whether to challenge the legality of the previous opponent's Wild Draw Four card.

For more details about the UNO games, please refer to Appendix A. The Figure 1 (A) shows the workflow diagrams of UNO Arena.

3.2 Players in the UNO Arena

In the UNO Arena, we initially involve three types of players: random player, reinforcement learning based player, vanilla LLM player. To further unleash the potential capability of LLMs in sequential decision-making, we propose TuTRI player, which involves reflection mechanism.

Random Player: As like its name suggests, the random player performs all actions randomly, such as randomly selecting a regulative card to play when it's their turn. The random player can be considered the baseline of the UNO Arena, mainly serving to maintain the flow of the UNO game. If some players outperform the random player, we can infer that these players are consciously playing UNO with an understanding of the game rules.

Reinforcement Learning Based Player: Previous research has sought breakthroughs in UNO using reinforcement learning models (Pfann, 2021).






| Card | Sample | Effect |
|----------------|--|--|
| Skip |  | The next player in sequence misses a round. |
| Reverse |  | Order of play switches directions (clockwise to counterclockwise, or vice versa). |
| Draw Two |  | The next player in sequence draws two cards and misses a round. |
| Wild |  | Player declares the next color to be matched (it can be used on any round even if the player has any card of matching color). |
| Wild Draw Four |  | Player declares the next color to be matched. The next player in sequence draws four cards and misses a round. May be legally played if the player has cards of the current color. |

Table 1: The effects of function and wild cards.

We built our reinforcement learning player with DQN (Mnih et al., 2013) model based on the open-source project RLcard (Zha et al., 2019).

Vanilla LLM Player: During the vanilla LLM player's turn, the game host transmits all publicly available information through a prompt to the LLM. The LLM then returns a JSON containing the decision and reasoning as required by the prompt. The Figure 1 (B) shows the workflow diagrams of vanilla LLM player.

TuTRI Player: While LLMs do not always generate the best output on their first try just as human (Madaan et al., 2023), iterative feedback and refinement could be a necessity for a better agent framework. Moreover, human-like thinking patterns, such as introspective reflections foster divergent thinking processes (Zhang et al., 2023), inspires us to propose the TuTRI player. This advanced framework is designed to navigate the intricacies of UNO game play, offering a more structured approach to strategic sequential decision-making. The original decision for TuTRI player is exactly the same as the vanilla LLM player's decision, after that are two additional reflection modules.

- **The Game History Reflection Module:** In the module, we provide statistical information about game history to TuTRI player, and they are told to *reflect the action you just selected* with these auxiliary information. Just like human thinking when playing UNO, if there is a large number of green cards that have already been played in the game's history, it is very

advantageous for players to play a green card. LLM should output both reflection thoughts and updated action.

- **The Game Strategy Reflection Module:** In the module, we provide additional useful game strategies to TUTRI players, and they are again told to *reflect the action you just selected* based on game strategies. For example, since wild cards can be played at any situations and disrupt other players, saving the wild cards in your hand as long as possible is a very useful game strategy. LLM should output both reflection thoughts and updated action (the final action).

It must be emphasized that the TUTRI player should work in a conversational manner, with exactly 3 times Q&A per round. Moreover, the TUTRI players may keep their original decision, in other words, literally updating the action is not a necessity, nevertheless, the reflection process, instead of simple I-O prompting of interaction, providing more opportunities for mistake correcting and divergent thinking. The Figure 1(C) shows the workflow diagrams of TUTRI player.

3.3 Monte Carlo Simulation Method for Monitoring Players' Behavior

In the game play, the change in each player's winning rate after making a decision is the key for tracking. In the classical combinatorial games, like Nim (Bouton, 1901) or Wythoff's Game (Wythoff, 1907), positions space are limited and thus computationally affordable, while UNO is more intricate, where positional space exponentially increases as cards number increases and the calculation gets tougher (Demaine et al., 2014).

To make a plausible ranking mechanism of the candidate decisions, we define the concept of **optimal decision**, meaning the state transferred by the decision from last state, has a highest winning rate concerning all subsequent outcomes, and thus adopt Monte Carlo Simulation (Mooney, 1997) to calculate the estimated winning rate.

Detailedly, with S_i representing the state of the game after the i -th step taken, $D_{i,j}$ representing the j -th legal decision candidates at state S_i , \mathcal{T} representing the state transfer function, \mathcal{E} representing the estimate function of state, thereby we have the definition of the optimal decision $D_{i,opt}$ at the i -th step where

$$opt = \arg \max_j \mathcal{E}(\mathcal{T}(S_{i-1}, D_{i,j})) \quad (1)$$

In calculation of $\mathcal{E}(S_i)$, we massively randomly generate the subsequent decision sequence $\{D_{i+1}, D_{i+2}, \dots\}$ and thus obtain the subsequent state sequence $\{S_{i+1}, S_{i+2}, \dots\}$. Then $\mathcal{E}(S_i)$ is assigned to the ratio of number of sequences where the player plays the state S_{i-1} comes as the winner, to the total number of sequences simulated.

As the times we simulate the subsequent sequence increases, the approximate value $\mathcal{E}(S_i)$ gets more precise, though we could not enumerate all the possible situations. To balance the time expenditure and the precision of the metrics, we control the simulation times in a certain range. Additionally, a threshold parameter p is set to identify critical decisions. We say a decision $D := D_i$ is **critical** if among its all candidate choices D_j

$$\max \mathcal{E}(\mathcal{T}(S, D_j)) - \min \mathcal{E}(\mathcal{T}(S, D_j)) \geq p \quad (2)$$

Actual decisions made on critical positions may have a huge effect on the winning rate, which is consistent with the game nature.

3.4 Evaluation Metrics in the UNO Arena

In our work, we design three evaluation metrics, including WR, ODHR@K and ADR@K, in conjunction with the UNO game to comprehensively evaluate the sequential decision-making capability of LLMs. Among these metrics, ODHR@K and ADR@K can off a glimpse into the intermediate results in the sequential decision-making of LLMs.

Winning Rate (WR). WR denotes the proportion of player wins to total game innings, and can be represented as:

$$WR = \frac{N_{Win}}{N_{Game}} \quad (3)$$

where N_{Win} represents the total number of times the given player has won, and N_{Game} denotes the total game innings.

Optimal Decision Hit Rate at K Decision Points (ODHR@K) : This metric measure the proportion of times players make the best decision to all decision times, when facing K decision points:

$$ODHR@K = \frac{N_{Hit@K}}{N_{Decision@K}} \quad (4)$$

where $N_{Hit@K}$ is the number of times the player makes the optimal decision when facing K optional decision points, and $N_{Decision@K}$ represents the total number of times the agent player makes decision when it faces K optional decision points.

Average Decision Rank at K Decision Points (ADR@K). This metric looks at the rank of output decision made by the player and can be denoted as:

$$\text{ADR@K} = \frac{\sum_{i=1}^{N_{\text{Decision@K}}} \text{Rank}(D_i)}{N_{\text{Decision@K}}} \quad (5)$$

where $\text{Rank}(D_i)$ represents the rank from best to worst among all legal decisions in its decision-making process, and $N_{\text{Decision@K}}$ represents the total number of times the agent player makes decision when it faces K optional decision points.

For metrics ODHR@K and ADR@K, according to the characteristics of UNO, we only focus on the situations where K is equal to 2, 3 or 4, because the vast majority of decisions in UNO do not exceed 4 (Pfann, 2021).

4 Experiments

In this section, we first conduct preliminary experiments with vanilla LLM players, RL players, and random players in subsection §4.1. Then, we have multiple different LLM-based vanilla players compete in UNO Arena to identify the best LLM in subsection §4.2. Next, we test the superiority of the TUTRI players compared to the vanilla LLM players in subsection §4.3. Finally, we perform ablation experiments on the TUTRI player in subsection §4.4.

To ensure the generalization of the experiments, we take the mainstream LLMs mentioned in the introduction: (1) gpt-3.5-turbo-16k-0613 (OpenAI, 2022); (2) gpt-4-1106-preview (Achiam et al., 2023); (3) Gemini-pro (Team et al., 2023); (4) Llama-2-7b-chat (Touvron et al., 2023); (5) ChatGLM3-6b (Du et al., 2021; Zeng et al., 2022).

4.1 1v1 UNO Arena between vanilla LLM players, RL players and random players

In order to verify the rationality of using UNO Arena to evaluate the sequential decision-making ability of LLMs, we first conduct experiments on vanilla LLM players, RL players and random players in 1V1 UNO Arena. We randomly generate 500 sets of UNO initial decks. Each vanilla LLM player or RL player have to play with the random player in these 500 initial decks. In addition, the random players are the first to play cards in all games. The results are shown in Table 2.

From the Table 2, we can find that (1) Except for ChatGLM3, the WR of other vanilla LLM players and RL players are all above 50.00%; (2) The performance of GPT-4 is the best, and GPT-4 performs

| Metrics | Vanilla LLM Players & RL Player with DNQ | | | | | |
|------------|--|--------------|------------|---------|----------|--------------|
| | GPT-3.5 | GPT-4 | Gemini-Pro | Llama 2 | ChatGLM3 | DNQ |
| WR (↑) | 55.80 | 63.20 | 53.80 | 53.60 | 48.80 | <u>57.40</u> |
| ODHR@2 (↑) | 57.34 | 61.47 | 53.94 | 53.69 | 49.75 | <u>54.96</u> |
| DAR@2 (↓) | 1.427 | 1.385 | 1.461 | 1.463 | 1.503 | <u>1.450</u> |
| ODHR@3 (↑) | 32.15 | 39.30 | 34.42 | 33.84 | 34.45 | <u>35.98</u> |
| DAR@3 (↓) | 2.010 | 1.904 | 2.017 | 1.994 | 2.034 | <u>1.947</u> |
| ODHR@4 (↑) | 27.20 | 36.99 | 31.05 | 27.39 | 25.36 | <u>37.74</u> |
| DAR@4 (↓) | 2.399 | 2.142 | 2.331 | 2.436 | 2.460 | <u>2.247</u> |

Table 2: Statistical results of random player VS vanilla LLM player or RL player with DNQ. The decision threshold p for critical decision in ODHR@K and ADR@K is 0.15. Bold indicates the best result, underline the second best result, and the Table 3 below follows this pattern.

| Metrics | Vanilla LLM Players | | | | |
|------------|---------------------|--------------|------------|--------------|----------|
| | GPT-3.5 | GPT-4 | Gemini-Pro | Llama 2 | ChatGLM3 |
| WR (↑) | <u>22.80</u> | 24.20 | 20.40 | 20.00 | 15.60 |
| ODHR@2 (↑) | 52.57 | 54.77 | 49.88 | <u>54.08</u> | 50.52 |
| DAR@2 (↓) | 1.474 | 1.452 | 1.501 | <u>1.459</u> | 1.495 |
| ODHR@3 (↑) | <u>39.56</u> | 41.41 | 33.14 | 34.78 | 33.13 |
| DAR@3 (↓) | <u>1.889</u> | 1.885 | 2.034 | 1.978 | 2.043 |
| ODHR@4 (↑) | <u>26.75</u> | 29.03 | 25.74 | 24.90 | 25.04 |
| DAR@4 (↓) | <u>2.407</u> | 2.366 | 2.516 | 2.471 | 2.477 |

Table 3: Statistical results of competition among 5 vanilla LLM players in UNO Arena. The decision threshold p for critical decision in ODHR@K and ADR@K is 0.00.

excellently on the 7 evaluation metrics. Especially, the WR of GPT-4 is 63.20%, 13.20% higher than 50.00%.

4.2 5-players UNO Arena with 5 LLMs

To find the best LLM, we place 5 LLMs in a 5-players UNO Arena to compete against each other. We fix the initial playing order of UNO Arena in the sequence of GPT-3.5, GPT-4, Gemini-Pro, Llama 2, and ChatGLM3. We conduct experiment on 200 decks generated randomly. All players are the vanilla LLM players. The results are shown in Table 3.

From the Table 3, we can find that (1) GPT-4 has the best performance, with a WR of 24.20%, 4.2% higher than the average (20.00%) and 1.4% higher than the second highest ranked GPT-3.5. Not only that, GPT-4 also performs the best in other 6 evaluation metrics; (2) ChatGLM3 has the worst performance, with a WR of 15.60%, which is 4.4% lower than the average (20.00%) and 8.6% lower than the highest ranked GPT-4. Not only that, ChatGLM3 also performs the worst in ODHR@2, DAR@2, DAR@3, ODHR@4, and DAR@4.

| LLM | WR (\uparrow) | ODHR@2 (\uparrow) | DAR@2 (\downarrow) | ODHR@3 (\uparrow) | DAR@3 (\downarrow) | ODHR@4 (\uparrow) | DAR@4 (\downarrow) |
|----------------------|-------------------|-----------------------|------------------------|-----------------------|------------------------|-----------------------|------------------------|
| GPT-3.5 (vanilla) | 48.00 | 53.05 | 1.4695 | 34.97 | 1.9508 | 34.47 | 2.2340 |
| GPT-3.5 (TuTRI) | 52.50 (+4.50%) | 54.01 (+0.06%) | 1.4599 (-0.06%) | 43.13 (+8.16%) | 1.8563 (-4.73%) | 32.92 (-1.55%) | 2.2667 (+1.09%) |
| GPT-4 (vanilla) | 49.00 | 56.27 | 1.4373 | 39.38 | 1.9375 | 36.24 | 2.2140 |
| GPT-4 (TuTRI) | 51.00 (+2.00%) | 56.60 (+0.33%) | 1.4340 (-0.33%) | 40.14 (+0.76%) | 1.8592 (-3.92%) | 36.33 (+0.09%) | 2.1510 (-2.10%) |
| Gemini-pro (vanilla) | 44.00 | 50.62 | 1.4938 | 37.04 | 2.0159 | 25.44 | 2.4737 |
| Gemini-pro (TuTRI) | 56.50 (+12.50%) | 53.64 (+3.02%) | 1.4636 (-3.02%) | 34.13 (-2.91%) | 1.9461 (-3.49%) | 30.36 (+4.92%) | 2.3482 (-4.18%) |
| Llama 2 (vanilla) | 47.00 | 49.54 | 1.5046 | 33.11 | 1.9595 | 29.11 | 2.3944 |
| Llama 2 (TuTRI) | 54.00 (+7.00%) | 55.07 (+5.53%) | 1.4493 (-5.53%) | 37.31 (+4.20%) | 1.8507 (-5.44%) | 26.75 (-2.36%) | 2.4650 (+2.35%) |
| ChatGLM3 (vanilla) | 47.00 | 55.82 | 1.4418 | 29.05 | 2.0541 | 31.84 | 2.2935 |
| ChatGLM3 (TuTRI) | 54.00 (+7.00%) | 57.24 (+1.42%) | 1.4276 (-1.42%) | 39.51 (+10.46%) | 1.8642 (-9.50%) | 30.62 (-1.22%) | 2.4689 (+5.85%) |

Table 4: Statistical results of vanilla LLM players VS TuTRI players. The decision threshold p for critical decision in ODHR@K and ADR@K is 0.00. Red annotations indicate favorable experimental results, while blue annotations indicate unfavorable experimental results.

| LLM | WR (\uparrow) | ODHR@2 (\uparrow) | DAR@2 (\downarrow) | ODHR@3 (\uparrow) | DAR@3 (\downarrow) | ODHR@4 (\uparrow) | DAR@4 (\downarrow) |
|----------------------|-------------------|-----------------------|------------------------|-----------------------|------------------------|-----------------------|------------------------|
| Gemini-pro (TuTRI) | 56.50 | 53.64 | 1.4636 | 34.13 | 1.9461 | 30.36 | 2.3482 |
| Gemini-pro + TuTRI' | 52.50 (-4.00%) | 54.33 (+0.69%) | 1.4567 (-0.69%) | 29.88 (-4.25%) | 2.0610 (+5.75%) | 31.25 (+0.89%) | 2.4219 (+2.46%) |
| Gemini-pro + TuTRI'' | 53.50 (-3.00%) | 54.59 (+0.95%) | 1.4541 (-0.95%) | 31.95 (-2.18%) | 2.0384 (+4.62%) | 27.59 (-2.77%) | 2.3824 (-1.14%) |

Table 5: Statistical results of the ablation study. Where TuTRI' represents the TuTRI player which remove the game history reflection module, and TuTRI'' represents the TuTRI player which remove the game strategy reflection module. The decision threshold p for critical decision in ODHR@K and ADR@K is 0.15. Red annotations indicate favorable experimental results, while blue annotations indicate unfavorable experimental results.

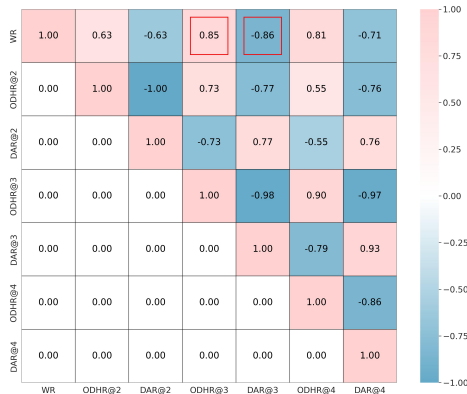


Figure 2: The Pearson Correlation Heatmap among WR, ODHR@K (K=2,3,4), and DAR@K (K=2,3,4).

4.3 Validation of the superiority of the TuTRI player compared to the vanilla LLM player

To verify that our TuTRI player can improve the sequential decision-making ability of LLMs, we compare the vanilla LLM players (baseline) with TuTRI players. We let 5 LLMs serve as the backend LLMs for both the vanilla LLM players and TuTRI players, and play two-players UNO Arena on 200 decks generated randomly. The results are shown in Table 4.

From the Table 4, we can find that: (1) All LLMs (the TuTRI player) are better than LLM (the vanilla LLM player) on WR, ODHR@2, and DAR@3.

Gemini-Pro (the TuTRI player) has a 12.50% higher than Gemini-Pro (the vanilla LLM player) on WR; (2) For ODHR@3, except for Gemini-Pro which performed slightly worse (-2.91%), the other 4 LLMs achieved good results. For ODHR@4 and DAR@4, GPT-4 and Gemini-Pro both performed well. It can be seen that the TuTRI player based on reflection can significantly improve its abilities of sequential decision-making after two rounds of reflection on the summary of game history and the game strategies. The experimental results strongly support the superiority of our TuTRI player based reflection over the vanilla LLM player.

4.4 Ablation studies on TuTRI player

To illustrate the necessity of the two reflection modules in the TuTRI player, we conduct ablation study. We remove the game history reflection module and the game strategy reflection module from the TuTRI players, and conduct two-players UNO Arena with vanilla LLM player respectively. The results are shown in Table 5.

From the Table 5, we can find that: (1) After removing the game history reflection module, the WR of decreased by 4%, the ODHR@3 of decreased 4.25%, and the DAR@3 of increase by 5.75%. (2) After removing the game strategy reflection module, the WR of decreased by 3%, the ODHR@3 of decreased 2.18%. and the DAR@3 of

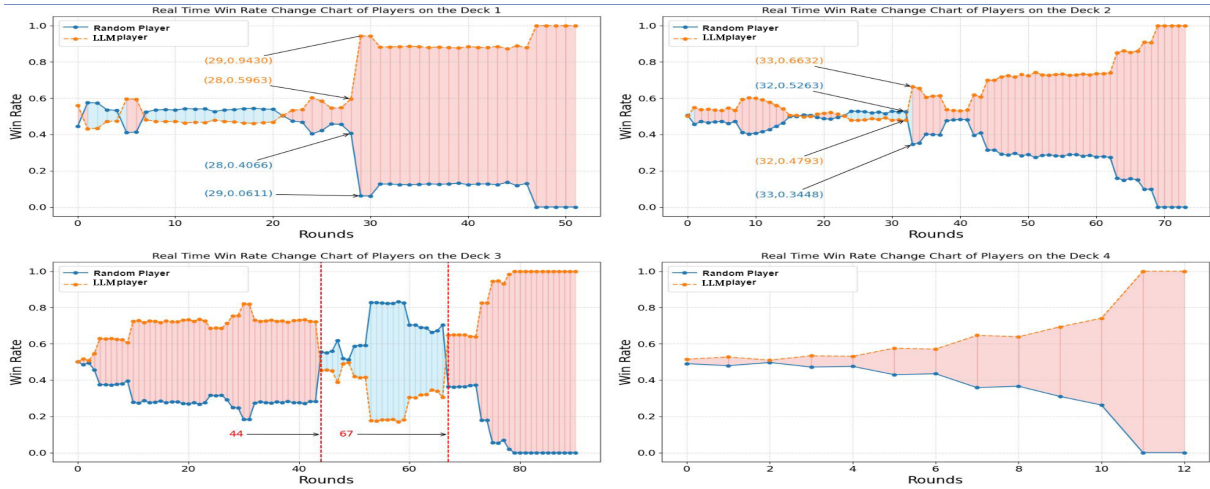


Figure 3: GPT-4 (the vanilla LLM player) real-time winning rate variations on 4 decks.

increase by 4.62%. The game history holds significant potential information for incomplete information games. Therefore, removing the game history reflection module has a greater adverse impact on the TUTRI player.

5 Discussion

5.1 Further Exploration of ODHR@K and ADR@K

To better analyze the relationship between our unique evaluation metrics (ODHR@K and ADR@K), and the evaluation metric WR, we conduct a Pearson correlation analysis of the experimental results from the Table 2. The results are shown in Figure 2. From the Figure 2, we can find that (1) WR shows a positive correlation with ODHR@K (K=2,3,4), and simultaneously, WR shows a negative correlation with ADR@K (K=2,3,4); (2) The strongest positive correlation, reaching 0.85, exists between WR and ODHR@3, while the strongest negative correlation, reaching -0.86, exists between WR and DAR@3. Overall, our unique ODHR@K and ADR@K have a good correlation with WR, so they can serve as reference evaluation metrics for evaluating LLMs in UNO Arena.

5.2 Case Study

In order to more intuitively see the advantages of LLM versus random player, we conduct a case study. We utilized GPT-4 as the backend LLM for the vanilla LLM player to engage in the game across 4 decks generated randomly, with the random player plays first. We recorded all decision

points (for both the vanilla LLM player and the random player) and employed the Monte Carlo method to calculate the real-time percentage change in winning rate for both sides following each decision point. The results are shown in Figure 3.

From the Figure 3, we can find that: (1) In UNO Arena, winning rates fluctuate significantly. For example, in deck 1, from round 28 to 29, the random player’s winning rate dropped by 34.5%, while the vanilla LLM player by 34.67%; (2) Turning points, like rounds 44 and 67 in deck 3, show shifts in dominance. Initially, the vanilla LLM player leads until round 44, then loses advantage until round 67, before regaining control; (3) Brief game durations occur, notably in deck 4, where the agent player consistently makes exceptional decisions, steadily increasing its winning rate until achieving victory. These findings underscore LLM’s adeptness at identifying crucial decision junctures and exploiting its capabilities, highlighting its potential in sequential decision-making scenarios.

6 Conclusion

In conclusion, LLMs possess the capability for sequential decision-making, as evidenced by the experimental results of LLMs playing the UNO game. Our proposed UNO Arena and unique evaluation metrics enable LLMs to compete with each other in the same UNO Arena game, thereby providing a better dynamic assessment of LLMs’ sequential decision-making abilities. Furthermore, we propose that the TUTRI player effectively addresses how to enhance LLMs’ sequential decision-making abilities for better performance in playing UNO Arena.

609 Limitations

610 The method of dynamically evaluating the sequen-
611 tial decision-making ability of LLMs using the
612 UNO Arena, as well as the TuTRI player, is only
613 applicable to LLMs that support *chat*. The unique
614 evaluation metrics, ODHR@K and ADR@K, in-
615 troduced in this paper are only applicable to games
616 or tasks with a limited action space.

617 References

618 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama
619 Ahmad, Ilge Akkaya, Florencia Leoni Aleman,
620 Diogo Almeida, Janko Altenschmidt, Sam Altman,
621 Shyamal Anadkat, et al. 2023. Gpt-4 technical report.
622 [arXiv preprint arXiv:2303.08774](#).

623 Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-
624 Yeol Ahn. 2023. [Can we trust the evaluation on
625 chatgpt?](#)

626 Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon
627 Oh, Matthias Bethge, and Eric Schulz. 2023. Playing
628 repeated games with large language models. [arXiv
629 preprint arXiv:2305.16867](#).

630 Eyal Amir. 2014. Reasoning and decision making. [The
631 Cambridge handbook of artificial intelligence](#), pages
632 191–212.

633 Saeid Amiri, Mohammad Shokrolah Shirazi, and Shiqi
634 Zhang. 2020. Learning and reasoning for robot
635 sequential decision making under uncertainty. In
636 [Proceedings of the AAI Conference on Artificial
637 Intelligence](#), volume 34, pages 2726–2733.

638 Charles L Bouton. 1901. Nim, a game with a complete
639 mathematical theory. [The Annals of Mathematics](#),
640 3(1/4):35–39.

641 Philip Brookins and Jason Matthew DeBacker. 2023.
642 Playing games with gpt: What can we learn about
643 a large language model from canonical strategic
644 games? [Available at SSRN 4493398](#).

645 Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu,
646 Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan
647 Yi, Cunxiang Wang, Yidong Wang, et al. 2023.
648 A survey on evaluation of large language mod-
649 els. [ACM Transactions on Intelligent Systems and
650 Technology](#).

651 Liting Chen, Lu Wang, Hang Dong, Yali Du, Jie Yan,
652 Fangkai Yang, Shuang Li, Pu Zhao, Si Qin, Saravan
653 Rajmohan, et al. 2023. Introspective tips: Large lan-
654 guage model for in-context decision making. [arXiv
655 preprint arXiv:2305.11598](#).

656 Erik D Demaine, Martin L Demaine, Nicholas JA
657 Harvey, Ryuhei Uehara, Takeaki Uno, and Yushi
658 Uno. 2014. Uno is hard, even for a single player.
659 [Theoretical Computer Science](#), 521:51–61.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding,
Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021.
Glm: General language model pretraining with
autoregressive blank infilling. [arXiv preprint
arXiv:2103.10360](#).

Keith Frankish and William M Ramsey. 2014. [The
Cambridge handbook of artificial intelligence](#). Cam-
bridge University Press.

Deep Ganguli, Nicholas Schiefer, Marina Favaro, and
Jack Clark. 2023. [Challenges in evaluating AI sys-
tems](#).

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane
Durante, Yusuke Noda, Zilong Zheng, Song-Chun
Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. 2023.
Mindagent: Emergent gaming interaction. [arXiv
preprint arXiv:2309.09971](#).

Jiaxian Guo, Bo Yang, Paul Yoo, Bill Yuchen
Lin, Yusuke Iwasawa, and Yutaka Matsuo. 2023.
[Suspicion-agent: Playing imperfect information
games with theory of mind aware gpt-4](#).

Shangmin Guo, Haoran Bu, Haochuan Wang, Yi Ren,
Dianbo Sui, Yuming Shang, and Siting Lu. 2024a.
Economics arena for large language models. [arXiv
preprint arXiv:2401.01735](#).

Shoutao Guo, Shaolei Zhang, Zhengrui Ma, Min Zhang,
and Yang Feng. 2024b. Sillm: Large language
models for simultaneous machine translation. [arXiv
preprint arXiv:2402.13036](#).

Zexue He, Yu Wang, Julian McAuley, and Bod-
hisattwa Prasad Majumder. 2022. Controlling bias
exposure for fair interpretable predictions. [arXiv
preprint arXiv:2210.07455](#).

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou,
Mantas Mazeika, Dawn Song, and Jacob Steinhardt.
2021. [Measuring massive multitask language under-
standing](#).

Dirk P Kroese, Tim Brereton, Thomas Taimre, and
Zdravko I Botev. 2014. Why the monte carlo method
is so important today. [Wiley Interdisciplinary
Reviews: Computational Statistics](#), 6(6):386–392.

Junjie Li, Sotetsu Koyamada, Qiwei Ye, Guoqing Liu,
Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan
Liu, and Hsiao-Wuen Hon. 2020. Suphx: Mastering
mahjong with deep reinforcement learning. [arXiv
preprint arXiv:2003.13590](#).

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu.
2023. [Avalonbench: Evaluating llms playing the
game of avalon](#).

Michael Lederman Littman. 1996. [Algorithms for
sequential decision-making](#). Brown University.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler
Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon,
Nouha Dziri, Shrimai Prabhumoye, Yiming Yang,
et al. 2023. Self-refine: Iterative refinement with
self-feedback. [arXiv preprint arXiv:2303.17651](#).

| | | |
|-----|---|-----|
| 715 | Jeffery S McMullen. 2015. Entrepreneurial judgment as empathic accuracy: A sequential decision-making approach to entrepreneurial action. <i>Journal of Institutional Economics</i> , 11(3):651–681. | 769 |
| 716 | | 770 |
| 717 | | 771 |
| 718 | | 772 |
| 719 | Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. <i>arXiv preprint arXiv:1312.5602</i> . | 773 |
| 720 | | 774 |
| 721 | | 775 |
| 722 | | 776 |
| 723 | | 777 |
| 724 | Christopher Z Mooney. 1997. <i>Monte carlo simulation</i> . 116. Sage. | 778 |
| 725 | | 779 |
| 726 | OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt . Accessed: 2023-09-30. | 780 |
| 727 | | 781 |
| 728 | Bernhard Pfann. 2021. Tackling the uno card game with reinforcement learning. <i>Towards Data Science: tackling-uno-card-game-with-reinforcement-learning</i> . | 782 |
| 729 | | 783 |
| 730 | | 784 |
| 731 | Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. <i>NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark</i> . In <i>Findings of the Association for Computational Linguistics: EMNLP 2023</i> , pages 10776–10787, Singapore. Association for Computational Linguistics. | 785 |
| 732 | | 786 |
| 733 | | 787 |
| 734 | | 788 |
| 735 | | 789 |
| 736 | | 790 |
| 737 | | 791 |
| 738 | Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. <i>Advances in Neural Information Processing Systems</i> , 36. | 792 |
| 739 | | 793 |
| 740 | | 794 |
| 741 | | 795 |
| 742 | | 796 |
| 743 | David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. <i>nature</i> , 550(7676):354–359. | 797 |
| 744 | | 798 |
| 745 | | 799 |
| 746 | | 800 |
| 747 | | 801 |
| 748 | Jubi Taneja, Avery Laird, Cong Yan, Madan Musuvathi, and Shuvendu K Lahiri. 2024. Llm-vectorizer: Llm-based verified loop vectorizer. <i>arXiv preprint arXiv:2406.04693</i> . | 802 |
| 749 | | 803 |
| 750 | | 804 |
| 751 | | 805 |
| 752 | Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. <i>arXiv preprint arXiv:2312.11805</i> . | 806 |
| 753 | | 807 |
| 754 | | 808 |
| 755 | | 809 |
| 756 | | 810 |
| 757 | | 811 |
| 758 | Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. <i>arXiv preprint arXiv:2302.13971</i> . | 812 |
| 759 | | 813 |
| 760 | | 814 |
| 761 | | 815 |
| 762 | | 816 |
| 763 | | 817 |
| 764 | Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. 2023a. Chatcad: Interactive computer-aided diagnosis on medical image using large language models. <i>arXiv preprint arXiv:2302.07257</i> . | 818 |
| 765 | | 819 |
| 766 | | 820 |
| 767 | | 821 |
| 768 | | 822 |
| | Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2023b. <i>Avalon’s game of thoughts: Battle against deception through recursive contemplation</i> . | |
| | | 774 |
| | | 775 |
| | | 776 |
| | | 777 |
| | | 778 |
| | Yu Wang, Zhiwei Liu, Jianguo Zhang, Weiran Yao, Shelby Heinecke, and Philip S Yu. 2023c. Drdt: Dynamic reflection with divergent thinking for llm-based sequential recommendation. <i>arXiv preprint arXiv:2312.11336</i> . | |
| | | 779 |
| | | 780 |
| | | 781 |
| | | 782 |
| | | 783 |
| | Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. <i>arXiv preprint arXiv:2206.07682</i> . | |
| | | 784 |
| | | 785 |
| | Willem A Wythoff. 1907. A modification of the game of nim. <i>Nieuw Arch. Wisk</i> , 7(2):199–202. | |
| | | 786 |
| | | 787 |
| | | 788 |
| | | 789 |
| | | 790 |
| | Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. <i>arXiv preprint arXiv:2309.07864</i> . | |
| | | 791 |
| | | 792 |
| | | 793 |
| | Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024. <i>Benchmark data contamination of large language models: A survey</i> . | |
| | | 794 |
| | | 795 |
| | | 796 |
| | | 797 |
| | Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023. <i>Exploring large language models for communication games: An empirical study on werewolf</i> . | |
| | | 798 |
| | | 799 |
| | | 800 |
| | Yang You, Liangwei Li, Baisong Guo, Weiming Wang, and Cewu Lu. 2019. Combinational q-learning for dou di zhu. <i>arXiv preprint arXiv:1901.08925</i> . | |
| | | 801 |
| | | 802 |
| | | 803 |
| | | 804 |
| | | 805 |
| | Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. <i>arXiv preprint arXiv:2210.02414</i> . | |
| | | 806 |
| | | 807 |
| | | 808 |
| | | 809 |
| | Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. 2024. <i>Mr-gsm8k: A meta-reasoning revolution in large language model evaluation</i> . | |
| | | 810 |
| | | 811 |
| | | 812 |
| | | 813 |
| | Daochen Zha, Kwei-Herng Lai, Yuanpu Cao, Songyi Huang, Ruzhe Wei, Junyu Guo, and Xia Hu. 2019. Rlcard: A toolkit for reinforcement learning in card games. <i>arXiv preprint arXiv:1910.04376</i> . | |
| | | 814 |
| | | 815 |
| | | 816 |
| | | 817 |
| | | 818 |
| | Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. 2024. <i>Fine-tuning large vision-language models as decision-making agents via reinforcement learning</i> . | |
| | | 819 |
| | | 820 |
| | | 821 |
| | | 822 |
| | Jintian Zhang, Xin Xu, and Shumin Deng. 2023. Exploring collaboration mechanisms for llm agents: A social psychology view. <i>arXiv preprint arXiv:2310.02124</i> . | |

| | | | |
|-----|---|--|-----|
| 823 | Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, | terclockwise, or from counterclock- | 869 |
| 824 | Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong | wise to clockwise). | 870 |
| 825 | Wen, and Jiawei Han. 2023. Don't make your llm an | – Draw Two: The player's next player | 871 |
| 826 | evaluation benchmark cheater. | draws two cards and skips this round | 872 |
| 827 | Appendix | of play. | 873 |
| 828 | A UNO Game | • The Wild Cards: the wild card includes 4 | 874 |
| 829 | In this section of appendix, you will learn what | Black Wild cards and 4 Black Wild Draw | 875 |
| 830 | UNO Game is, and in order to facilitate the | Four cards, totaling 8 cards. | 876 |
| 831 | evaluation of LLMs with the UNO Game, we | – Wild: the player selects one | 877 |
| 832 | have made some slight modifications to it. | color from the COLOR set | 878 |
| 833 | A.1 Game Objective | { <i>Red, Blue, Yellow, Green</i> } as | 879 |
| 834 | In the modified UNO game, We simply set | the new color for the top card in the | 880 |
| 835 | the game objective as to be the first player to | discard pile. | 881 |
| 836 | clear out the hand. Players play alternatively(2 | – Wild Draw Four: the player se- | 882 |
| 837 | players) or in circle manner(3 or more players) | lects one color from the COLOR set | 883 |
| 838 | and strive to achieve the unique goal. It should | { <i>Red, Blue, Yellow, Green</i> } as the | 884 |
| 839 | be noted that if the cards in the deck are ex- | new color for the top card in the dis- | 885 |
| 840 | hausted by players, the player with the fewest | card pile, and the player's next player | 886 |
| 841 | number of cards in hand wins, so there may be | draws 4 cards. | 887 |
| 842 | multiple winners in the same game. | A.3 Game Progress | 888 |
| 843 | A.2 UNO Cards | First, deal each player 7 initial cards in clock- | 889 |
| 844 | UNO comprises 3 categories of cards: number | wise order, then continue drawing cards until a | 890 |
| 845 | cards, function cards, and wild cards. In total, | number card is drawn and set as the top card of | 891 |
| 846 | UNO features 108 cards. | the initial discard pile. All players take rounds | 892 |
| 847 | • The Number Cards: the number cards | playing cards in clockwise order(it will be re- | 893 |
| 848 | can be expressed in the form of COLOR | versed by a reverse card) until a player runs | 894 |
| 849 | + NUMBER, where COLOR is belong | out of his cards or the draw pile is exhausted, | 895 |
| 850 | to the set { <i>Red, Blue, Yellow, Green</i> }, | signaling the end of the game. | 896 |
| 851 | and NUMBER is an integer from 0 to 9. | A.4 Legal Decision(Action) | 897 |
| 852 | It is important to note that there is only | In every round of the game, player in charge | 898 |
| 853 | one 0-number card per color, while there | can using rules to match the top card of the | 899 |
| 854 | are two 1-9 number cards per color. There | discard pile otherwise pick up a new card into | 900 |
| 855 | are a total of 76 number cards. | hand. The rules, or say, the legal decisions | 901 |
| 856 | • The Function Cards: the function cards | consists of several sorts: | 902 |
| 857 | can be expressed in the form of COLOR | • Draw Card: If a player does not have any | 903 |
| 858 | + FUNCTION, where COLOR is belong | cards to play during their playing round, | 904 |
| 859 | to the set { <i>Red, Blue, Yellow, Green</i> }, | they must draw a card, or their previous | 905 |
| 860 | and FUNCTION is belong to the set | player used Draw Two or Wild Draw Four | 906 |
| 861 | { <i>Skip, Reverse, DrawTwo</i> }. There are | cards to make the player draw multiple | 907 |
| 862 | two cards of the same COLOR for each | cards. | 908 |
| 863 | FUNCTION. There are a total of 24 func- | • Select Card: In a player's playing round, | 909 |
| 864 | tion cards. | they need to play a card that matches ei- | 910 |
| 865 | – Skip: the player's next player skips | ther the COLOR, NUMBER, or FUNC- | 911 |
| 866 | this round of play. | TION of the top card in the discard pile, | 912 |
| 867 | – Reverse: the player reverses the or- | or play a Wild card(include Wild Draw | 913 |
| 868 | der of play (from clockwise to coun- | Four card) to match. The card played by | 914 |
| | | the player then becomes the new top card. | 915 |

- **Select Color:** After selecting either Wild Card or Wild Draw Four Card, the player needs to convert the color of the current top card to one of $\{Red, Blue, Yellow, Green\}$.
- **Select ChallengeFlag:** The use of the Wild Draw Four card may be illegal. After a player plays a Wild Draw Four card, their next player can choose to challenge its use. If the player who played the Wild Draw Four card still holds non-Wild cards matching the color of the current top card, the use of the Wild Draw Four card is illegal. Possible scenarios are as follows: (1) If the player’s play is illegal and their next player challenges it, the player must draw 4 cards, and their next player faces no penalty; (2) If the player’s play is legal and their next player challenges it, the player’s next player must draw 6 cards. (3) If the player’s next player does not challenge, regardless of the legality of the player’s play, the player’s next player must draw 4 cards. Note that Challenge is not a stand-alone action to complete turns, it should be accompanied by a card draw or card match action.

B Prompt

Here is the prompt design for the entire experiment.

B.1 Select Card

The input1 prompt of the select card shared by the vanilla LLM player and the TUTRI player is shown in the Figure 4. The game history Reflection module prompt of the select card for the TUTRI player is shown in the Figure 5. The game strategy reflection module prompt of the select card for the TUTRI player is shown in the Figure 6.

B.2 Select Color

The input1 prompt of the select color shared by the vanilla LLM player and the TUTRI player is shown in the Figure 7. The game history reflection module prompt of the select color for the TUTRI player is shown in the Figure 8. The game strategy reflection module prompt of the select color for the TUTRI player is shown

in the Figure 9.

B.3 Select ChallengeFlag

The input1 prompt of the select challenge-Flag shared by the vanilla LLM player and the TUTRI player is shown in the Figure 10. The game history reflection module prompt of the select challengeFlag for the TUTRI player is shown in the Figure 11. The game strategy reflection module prompt of the select challengeFlag for the TUTRI player is shown in the Figure 12.

Select Card Input1 Prompt

You are playing a two-player UNO game.

- You are the player{player_id}, and your opponent is the player{opponent_id}.
- Currently, there are {len_deck} cards in the deck, and the discard pile has {len_discard_pile} cards.
- The number of cards in the hand of your opponent is {len_opponent_hand}.
- The game history of the last {len_history} rounds is {history}.
- Your entire hand consists of {hand}.
- The cards you can play are: {playable_card}.

Please note that you are not playing a normal UNO game, if the cards in the deck are depleted, the person who has the minimum cards will win directly. The goal of the game is to minimize the number of cards in your possession. In order to win the UNO game, you must consider all the provided information and select the best card from the cards you can play.

The output should strictly be a JSON object with two keys: 'thoughts' and 'action'.

- In this context, the value corresponding to the 'thoughts' key represents your thoughts and considerations, with its data type being a string.
- Simultaneously, the value corresponding to the 'action' key is a int which represents the card index of the card you have selected.

Figure 4: The input1 prompt of the select card shared by the vanilla LLM player and the TuTRI player.

Select Card Reflection1 Prompt

Here is the statistical data of the game history:

{history_summary}

Now, in order to win the game, you must consider the statistical data carefully and reflect the action you just selected.

You should strictly output a JSON object with two keys: 'reflection' and 'action'.

- The value corresponding to the 'reflection' key is your reflection.
- The value corresponding to the 'action' key is the card index you currently select.

Figure 5: The game history reflection module prompt of the select card for TuTRI player.

Select Card Reflection2 Prompt

Here is an useful tip that you can follow:

- The card values range from low to high, starting with number cards 0, followed by number cards (1-9), reverse cards, skip cards and wild cards.
- It is better to start with low-value cards before playing high-value cards.
- Unless your opponent is on the verge of victory, it is time to play some high-value cards to disrupt your opponent's strategy.

Now, in order to win the game, you should reflect the action you just selected based on the tip.

You should strictly output a JSON object with two keys: 'reflection' and 'action'.

- The value corresponding to the 'reflection' key is your reflection.
- The value corresponding to the 'action' key is the final card index you currently select.

Figure 6: The game strategy reflection module prompt of the select card for TuTRI player.

Select Color Input1 Prompt

You are playing a two-player UNO game.

- You are the player{player_id}, and your opponent is the player{opponent_id}.
- Currently, there are {len_deck} cards in the deck, and the discard pile has {len_discard_pile} cards.
- The number of cards in the hand of your opponent is {len_opponent_hand}.
- The game history of the last {len_history} rounds is {history}.
- Your entire hand consists of {hand}.

Please note that you are not playing a normal UNO game, if the cards in the deck are depleted, the person who has the minimum cards will win directly. The goal of the game is to minimize the number of cards in your possession. In order to win the UNO game, you just played a {wild_type} card, and you must consider all the provided information and select the best color from Red, Yellow, Blue and Green to switch.

The output should strictly be a JSON object with two keys: 'thoughts' and 'action'.

- In this context, the value corresponding to the 'thoughts' key represents your thoughts and considerations, with its data type being a string.
- Simultaneously, the value corresponding to the 'action' key is one of Red, Yellow, Blue or Green, indicating the color you have selected.

Figure 7: The input1 prompt of the select color shared by the vanilla LLM player and the TUTRI player.

Select Color Reflection1 Prompt

Here is the statistical data of the game history:

{history_summary}

Now, in order to win the game, you must consider the statistical data carefully and reflect the action you just selected.

You should strictly output a JSON object with two keys: 'reflection' and 'action'.

- The value corresponding to the 'reflection' key is your reflection.
- The value corresponding to the 'action' key is the color you currently select.

Figure 8: The game history reflection module prompt of the select color for TUTRI player.

Select Color Reflection2 Prompt

Here are some useful tips that you can follow:

- It is better to select the color with the highest frequency of occurrence in your hand.
- It is better to avoid selecting the color with the lowest frequency of occurrence in your hand.
- Consider carefully which color of cards is relatively more frequent in your opponent's hand and try to avoid selecting that color.

Now, in order to win the game, you should reflect the action you just selected based on these tips.

You should strictly output a JSON object with two keys: 'reflection' and 'action'.

- The value corresponding to the 'reflection' key is your reflection.
- The value corresponding to the 'action' key is the final color you currently select."

Figure 9: The game strategy reflection module prompt of the select color for TUTRI player.

Select ChallengeFlag Input1 Prompt

You are playing a two-player UNO game.

- You are the player{player_id}, and your opponent is the player{opponent_id}.
- Currently, there are {len_deck} cards in the deck, and the discard pile has {len_discard_pile} cards.
- The number of cards in the hand of your opponent is {len_opponent_hand}.
- The game history of the last {len_history} rounds is {history}.
- Your entire hand consists of {hand}.

Your opponent played a Wild Draw Four card, and changed the color of the current discard pile's top card to {new_color}. But the use of the Wild Draw Four card may be illegal, when your opponent still has cards in {old_color}. Please note that you are not playing a normal UNO game, if the cards in the deck are depleted, the person who has the minimum cards will win directly. The goal of the game is to minimize the number of cards in your possession. In order to win the UNO game, you must consider all the provided information and select whether to challenge the use of the Wild Draw Four card which played by your opponent.

The output should strictly be a JSON object with two keys: 'thoughts' and 'action'.

- In this context, the value corresponding to the 'thoughts' key represents your thoughts and considerations, with its data type being a string.
- Simultaneously, the value corresponding to the 'action' key is 'Yes' or 'No', indicating that you select to challenge or not to challenge, respectively."

Figure 10: The input1 prompt of the select challengeFlag shared by the vanilla LLM player and the TuTRI player.

Select ChallengeFlag Reflection1 Prompt

Here is the statistical data of the game history:

{history_summary}

Now, in order to win the game, you must consider the statistical data carefully and reflect the action you just selected.

You should strictly output a JSON object with two keys: 'reflection' and 'action'.

- The value corresponding to the 'reflection' key is your reflection.
- The value corresponding to the 'action' key is the choice you currently select.

Figure 11: The game history reflection module prompt of the select challengeFlag for TuTRI player.

Select ChallengeFlag Reflection2 Prompt

Here are some useful tips that you can follow:

- Please remember the penalty for a failed challenge: you must draw 6 cards.
- Please remember the benefits of a successful challenge: your opponent must draw 4 cards.
- Wild Draw Four is only illegal if your opponent has cards of {old_color} color in his hand.

Please carefully consider whether your opponent's Wild Draw Four card is genuinely illegal. Now, in order to win the game, you should reflect the action you just selected based on these tips.

You should strictly output a JSON object with two keys: 'reflection' and 'action'.

- The value corresponding to the 'reflection' key is your reflection.
- The value corresponding to the 'action' key is the final choice you currently select.

Figure 12: The game strategy reflection module prompt of the select challengeFlag for TuTRI player.