SKYROWAR: THE LLM-BASED MULTI-AGENT CARD GAME-PLAYING FRAMEWORK

Gökay Gülsoy, Enes Doğan, Umur Akgül & Cüneyd Çelik
Department of Computer Engineering
İzmir Institute of Technology
İzmir, Gülbahçe 35433, Turkey
{gokaygulsoy,enesdogan,umurakgul,cuneydcelik}@iyte.edu.tr

ABSTRACT

The development of large language models (LLMs) has been a substantial advancement in the fields of natural language processing and computer vision. All that progress has urged researchers to explore "LLMs as agents" to deal with complex tasks, in which LLM-based agents, in many cases, have demonstrated promising generalization capabilities compared to conventionally trained models. The observed capabilities of LLMs have brought about considerable interest in their application to strategic game playing. This study establishes an LLM-based multi-agent strategic card game-playing framework to evaluate the collaborative abilities of a team of LLM-based agents and strategic planning capabilities against the opponent team of LLM-based agents. A 2-team turn-based game architecture consists of 3 agents for each team, which are team generation, team manager, and team lead agents, respectively. Agents collaborate and plan as a team to defeat the opponent team. 8 different LLM models of distinct scales from different providers are used in experiments, which are gpt-4o, gpt-4o-mini, claude-sonnet-4-5-20250929, claude-haiku-4-5-20251001, gemini-2.5-pro, gemini-2.5-flash, llama-3.3-70b-versatile, and llama-3.1-8b-instant, respectively. As evaluation metrics, win rate and average damage rate are used to assess the capabilities of the different LLM-based agent teams. Experimental Results showed that larger parameter size models consistently outperform the small parameter size models, with the exception that gemini-2.5-flash, which beat gemini-2-pro. All commercial models achieved higher rewards when battled against the open-source llama-3.3-70b-versatile and llama-3.1-8b-instant models. Among the large parameter size models, gemini-2.5-pro achieved the highest values on all metrics by outperforming gpt-4o and claude-sonnet-4-5-20250929 The gpt-4o, gemini-2.5-pro, gemini-2.5-flash, and open-source llama models yield the fewest number of hallucinations.

1 INTRODUCTION

In the realm of researching Artificial General Intelligence (AGI), digital games are considered important for their provision of intricate challenges that require advanced reasoning and cognitive abilities, acting as an ideal standard for evaluating agent and system performance. Thus, investigating the performance of LLMs within complicated game-playing environments is crucial for figuring out their existing limitations and assessing the progress towards autonomy and generalizability, so as to move closer to a potential AGI (Xu et al., 2024). Strategic game-playing requires advanced reasoning, planning and collaborative skills for which LLMs are quite suitable, but usually complex game-playing requires more than a single LLM agent to better approximate human reasoning, which can be achieved by Multi-agent LLM systems (Sreedhar & Chilton, 2024; Neto, 2005). This study implements a multi-agent card game-playing framework to evaluate the collaborative abilities of LLM-based agents and their strategic planning capabilities. Game structure consists of 2 teams, each of which consists of 3 agents, which are the team generation, team manager, and team lead agent. The aim of the game is to select 4 heroes from a pool of 8 heroes to form a team and defeat all heroes of the hostile team, where each team of 3 agents should collaborate with each other and

strategize their actions via strategical hero selection, determining appropriate skills to use taking existing circumstances and strategy into account, and distinguishing the most suitable target to perform an attack to achieve this purpose. Win rate and average damage rate are used as evaluation metrics for comparing the collaborative and planning capabilities of a team of agents. 8 different LLM models of distinct sizes from different providers are used in experiments, which are gpt-4o, gpt-4o-mini, claude-sonnet-4-5-20250929, claude-haiku-4-5-20251001, gemini-2.5-pro, gemini-2.5-flash, llama-3.3-70b-versatile, and llama-3.1-8b-instant, respectively. The project is provided via Github repository[1], which can be used for further research and experimentation. The structure of the paper is as follows: Section 2 provides a review of related work in the area. Section 3 explains the framework used in this study, along with the methodology followed. Section 4 outlines the experimental setup and evaluation. Finally, Section 5 presents conclusions and key takeaways from this study.

2 Related Works

Many different studies have been conducted in the field of LLM-based agentic game-playing, which try to assess logical reasoning, planning, and collaborative task completion capabilities of LLM-based agents. This study is mainly inspired by the Aquawar (Liu et al., 2025), which is one of these novel frameworks that assesses the interactive capabilities of LLM-based agents via letting the agent and human play a round and rule-based strategy game against each other. The game was designed to have various fish types with predetermined health points as characters with their specific active and passive skill sets. Teams of the game form their crew, consisting of different fish at the beginning of the game, and each round, player tries to guess the opponent's fish to deal damage prior to the actual attack, and then choose their fish and its attack type to perform an attack. The victory condition is to have more fish alive at the end of the game. It also allows game difficulty to be set, which implies that the team with superior combat power has a higher probability of being victorious, allowing researchers to set specific difficulty levels for evaluating the win rates of diverse strategies.

(Li et al., 2023) seeks to broaden LLMs' planning capabilities in cooperative multi-agent scenarios via making Theory of Mind (ToM) assessments in the middle of an interactive task, in which the mental state of each agent changes dynamically. The main difference of that study from prior research is that, as agents exchange information through communication at every timestamp and make observations, reasoning complexity increases along with updates in the mental states of agents, leading to more challenging scenarios to be tested. The task environment consists of three agents, five rooms, and 5 bombs. Bombs may have three different colors. Each successfully defused bomb rewards the team ten points, resulting in 90 points as per the mission. Team performance is measured using two metrics: team score, which indicates coordination quality and rounds to completion, measuring collaboration efficiency. It proposed a prompt engineering method incorporating belief state about world knowledge to mitigate two systematic failures, namely, long-horizon contexts and hallucinations. *Pokemon League* (Yashwanth & C, 2025), which is a novel tournament framework, evaluates LLMs as strategic agents in battles. The tournament has shown that most agents converged on balanced team archetypes that reflect human convention in competitive play. Contrary to what most agents followed, the winning agent prioritized overwhelming stats and exploited and underutilized, yet valid strategies that surpassed more conservative compositions. The diversity in approaches reflected the ability of LLM-based agents to be flexible when applying general knowledge creatively in adversarial environments.

The multi-agent strategic game-playing framework named Balderdash is implemented by (Hejabi et al., 2024), which demonstrated that infrequent vocabulary in LLMs' input leads to poor performance in reasoning about game rules and historical context. The study aimed to assess the ability of LLM-based agents to generate plausible definitions for obscure words in Balderdash and examine their logical reasoning skills by observing how effectively they deceive opponents and identify correct definitions of words in the context of the game. Another empirical study conducted on a game named *Werewolf* (Xu et al., 2023) explored the problem of using LLMs in this communication game and proposed a tuning-free framework by keeping the LLMs frozen, relying on retrieval and reflection on past communications along with experiences for improvement, and tried to address the issue of limited context length. *MultiMind* (Zhang et al., 2025), the novel framework, presented

[1] https://github.com/GokayGulsoy/Skyrowar-LLM-based-Multi-Agent-Battle-Simulator/tree/main

research that integrates multi-modal information into social deduction games (SDG) for the first time. MultiMind turns facial expressions and voice tone into textual representations that capture emotional signals, which enable agents to process multi-modal information, and it employs Monte Carlo Search Tree (MCTS) to optimize communication strategies based on the Theory of Mind (ToM) model to minimize suspicion directed at the agent. (Hu et al., 2024) provides a survey on an up-to-date review of LLM-based game agents (LLMGAs) through a unified reference architecture. At the single-agent level, it combines available studies around three main components: memory, reasoning, and perception-action interfaces, which collectively characterize how language used enables agents to perceive, think, and act. At the multi-agent level, it outlines how communication protocols and organizational patterns support coordination, role separation, and large-scale social interactions. It introduces a challenge-centered taxonomy connecting six game genres to their prevailing agent requirements.

3 METHODOLOGY

This study provides a multi-agent card game-playing framework for evaluating and comparing the reasoning, collaborative, and planning capabilities of LLM-based agents as a team internally and against each other. Each team consists of three agents with specialized tasks and carefully designed prompts, which are the team generation agent, the team manager agent, and the team lead agent. Also, agents are integrated into the game in a way so that researchers can easily analyze the flow of the game via logs and can analyze how agents reason, communicate, and plan while performing their actions. A well-known problem of LLMs when generating their outputs is *hallucinations*, which refer to instances where the model generates inaccurate or fictitious information, diverging from factual knowledge and potentially yielding responses that lack a ground in the model's training data (Perković et al., 2024). Framework highlights hallucinations that may occur during game flow, with a log message regarding where they happened. Win rate, average damage rate, and number of hallucinations are used as evaluation metrics for the team of agents. 14 different experiments were conducted with different LLM combinations for the team of agents that were chosen from a pool of 8 LLM models. The LLM models used in experiments are gpt-4o, gpt-4o-mini, claude-sonnet-4-5-20250929, claude-haiku-4-5-20251001, gemini-2.5-pro, gemini-2.5-flash, llama-3.3-70b-versatile, llama-3.1-8b-instant, respectively. Subsection 3.1 explains game logic and flow. Subsection 3.2 gives details regarding types of heroes and their skills available. Subsection 3.3 provides responsibilities and a prompt structure of each agent comprising a team.

3.1 GAME LOGIC AND FLOW

Construction Details: The first type of interaction is the action phase, where each team needs to select the hero with which it wants to act and choose the target to apply the skill. The second type of interaction is the guess phase, in which the team whose round is turned guesses the identity of the attack target before performing the attack.

The following is the detailed definition and description of the game process:

- **Player and Cards:** it's a team-of-agents vs team-of-agents battle game in which each team has four heroes selected from a pool of 8 heroes.
- **Initial State:** Each hero has 400 initial health, 200 initial attack power, active and passive abilities.
- **Attack Rule:** a team chooses a live hero to use its active skill or normal attack on the enemy hero each round. All alive heroes' passive abilities will automatically trigger when certain conditions are met.
- **Identification Mechanism:** the identity of the team's heroes is initially hidden. The team guesses one of the heroes' identities of the hostile team each round. If the team guesses correctly, the hostile team's target hero's identity is revealed, and all heroes of the hostile team will get 50 damage.
- **Round Process:** Within a round, the team whose turn it is will assert the identity of the opponent team's one hero who is alive and whose identity is still not revealed. If the assertion is correct, all of the opponent's heroes that remain alive get damaged. Subsequently,

the team for that round can command one alive hero to execute a normal attack or an active ability. Following that, any hero that meets the condition will unleash its passive ability.

- **Victory Condition:** The victory condition is to eliminate the hostile team first.

Metrics. Win rate, Average damage rate, and number of hallucinations are used for evaluation metrics. We provide the final reward score according to winning rate (**Win Rate**) and total damage inflicted compared to total health (**Damage Rate**), and hallucinated behaviors directly give turn to the hostile team, which automatically penalizes the attacking team for the hallucinated round. Reward calculation for a single round is given as follows:

$$reward = 0.7 * metric_{winrate} + 0.3 * metric_{damagerate}$$

For the experiments, to provide statistical significance average reward and average damage rate over 5 rounds were calculated, which can also be configured within the framework to run experiments for more or fewer rounds.

3.2 Types of Heroes and Skills

The Skyrowar framework has eight kinds of heroes with distinct active and passive skills, which are described as follows:

- **Argonian**
  - **Counter (Passive):** Inflicts 30 damage to the attacker when a teammate's health is below 30%.
  - **AOE (Active):** Attacks all enemies for 35% of its attack point.
- **Khajit**
  - **Counter (Passive):** Inflicts 30 damage to the attacker when a teammate's health is below 30%.
  - **Infight (Active):** Inflicts 75 damage on one living teammate and increases your attack points by 140.
- **Redguard**
  - **Deflect (Passive):** Distributes 70% damage to teammates and takes 30% damage when attacked. Gains 40 attack points after taking 200 damage accumulated.
  - **Infight (Active):** Inflicts 75 damage on one living teammate and increases your attack points by 140.
- **Nord**
  - **Reduce (Passive):** There is a 30% chance to avoid any incoming damage each time.
  - **Crit (Active):** Deals 120 CRITICAL damage to enemy.
- **Breton**
  - **Reduce (Passive):** There is a 30% chance to avoid any incoming damage each time.
  - **Subtle (Active):** Choose a teammate or yourself to reduce the damage by 70% when attacked, and increase your attack point by 20.
- **Imperial**
  - **Heal (Passive):** Regain 20 health points if the health is still greater than 0 when attacked.
  - **Infight (Active):** Inflicts 75 damage on one living teammate and increases your attack points by 140.
- **Onsimer**
  - **Heal (Passive):** Regain 20 health points if the health is still greater than 0 when attacked.
  - **Crit (Active):** Deal 120% CRITICAL damage of your attack power to the enemy with the lowest health. If the target's health is below 160, increase CRITICAL damage to 140%.

- **Bosmer**
  - **Explode (Passive):** Deal 40 damage to the source when attacked, but not died. when the health is below 30%, increase its attack points by 15.
  - **Crit (Active):** Deal 120% CRITICAL damage of your attack power to the enemy with the lowest health. If the target's health is below 160, increase CRITICAL damage to 140%.

As can be seen in the above hero specifications, there is an overlap among the active and passive skills of different hero types, which is done to conceal the identity information of the hero better and increase strategic aspects of the game.

3.3 Responsibilities of Agents and Prompt Structures

This section gives details of each agent's responsibilities and a prompt structure.

Team Generation Agent Responsibility: Out of eight heroes, strategically select four heroes to form a team.

Prompt for Team Generation Agent is as follows:

Team Generation Agent Prompt

This is a two-team battle game with four heroes in each team and you are the Team Generation Agent. There are eight types of heroes in the hero pool. Each hero has its initial health, attack power, active ability, and passive ability. Your task is to choose four different heroes from a pool of eight heroes according to your team forming strategy.

Format of each hero is given as follows:

{

{'Argonian': {'passive': "Counter: Deal 30 damage to the attacker when a teammate's health is below 30%.", 'active': "AOE: Attacks all enemies for 35% of its attack point."}},

{'Khajit': {'passive': "Counter: Deal 30 damage to the attacker when a teammate's health is below 30%.", 'active': "Infight: Deal 75 damage on one living teammate and increase your attack points by 140. Notice! You can't attack yourself or a dead teammate!"} },

{'Redguard': {'passive': "Deflect: Distribute 70% damage to teammates and take 30% damage when attacked. Gains 40 attack points after taking 200 damage accumulated.", 'active': "Infight: Deal 75 damage on one living teammate and increase your attack points by 140. Notice! You can't attack yourself or a dead teammate!"}},

{'Nord': {'passive': "Reduce: There is a 30% chance to avoid any incoming damage each time.", 'active': "Crit: Deal 120 CRITICAL damage to enemy"}},

{'Breton': {'passive': "Reduce: There is a 30% chance to avoid any incoming damage each time.", 'active': "Subtle: Choose a teammate or yourself to reduce the damage by 70% when attacked, and increase your attack point by 20."}},

{'Imperial': {'passive': "Heal: Regain 20 health points if the health is still greater than 0 when attacked.", 'active': "Infight: Deal 75 damage on one living teammate and increase your attack points by 140. Notice! You can't attack yourself or a dead teammate!"}},

{'Onsimer': {'passive': "Heal: Regain 20 health points if the health is still greater than 0 when attacked.", 'active': "Crit: Deal 120% CRITICAL damage of your attack power to the enemy with the lowest health. If the target's health is below 160, increase CRITICAL damage to 140%."}},

{'Bosmer': {'passive': "Explode: Deal 40 damage to the source when attacked, but not died. when the health is below 30%, increase its attack points by 15.", 'active': "Crit: eal 120% CRITICAL damage of your attack power to the enemy with the lowest health. If the target's health is below 160, increase CRITICAL damage to 140%."}}

}

Team Genertion Agent Prompt Cont'd

REQUIRED OUTPUT FORMAT (STRICT JSON):
Return a single JSON dictionary.
1. KEYS: Must be the Hero Name (e.g., "Argonian", "Nord").
2. VALUES: Must be a DICTIONARY ({{...}})
3. VALUE STRUCTURE:
{{
"passive": "<copy text exactly from above>",
"active": "<copy text exactly from above>"
}}
After you chase four heroes, return them in the above structured format because this team information will be used by Team Manager Agent of your team to keep your team's stats. Do not output any markdown code blocks or explanatory text. Just the JSON.
{{format_instructions}}

Team Manager Agent Responsibility: Keeping track of health, attack power, alive or dead, and whether identity information is revealed or not revealed for each hero in the team. Selecting an alive hero to attack the enemy team and providing this information to the Team Lead Agent.

Prompt for Team Manager Agent is as follows:

Team Manager Agent Prompt

This is a two-team battle game with four heroes in each team and you are the Team Manager Agent. You are responsible for keeping track of health, attack power, alive or dead, identity revealed or not revealed information for each hero.

Format for each hero is:
'hero name': {{'passive': "...", 'active': "..."}}

Initially, each hero has 400 health, 200 attack power, "not revealed", and "alive".

YOUR GOAL:
When asked, you must select ONE alive hero to perform an ACTIVE SKILL.
1. DO NOT always select same hero by just considering that it will deal massive damage for now, consider using different heroes to strategize your attacks for future.
2. You CAN NOT select same hero more than twice in succession.
3. You need to select each hero at least once in your team.
4. Internal/Buff: Moves that target a teammate
- "Infight": Deal 75 damage to a teammate to gain 140 Attack Power.
- Rule: You cannot target yourself or a dead teammate with Infight.
- "Subtle" Target self or teammate -> Reduce next damage by 70% & Gain 20 AP.

AVAILABLE MOVES:
1. **Hero Specific Active Skill**: The unique skill defined in the hero's description.
2. **Basic Attack**: ANY hero can choose to deal '100%' of their Attack Power to a single enemy.

SKILL TYPES & TARGETING RULES:
1. Attack Enemy (ACTIVE SKILL):
- Skills: "Crit", "AOE", "Basic Attack".
- TARGET: Must be 'enemy'.
2. Internal/Buff:
- "Infight", "Subtle".
- TARGET: Must be 'teammate'.

STRATEGY RULES:
1. NEVER use "Infight" on a teammate who is already DEAD.
2. If you are the only one alive, you CANNOT use Infight (no valid target). Attack the enemy instead.

Team Manager Agent Prompt Cont'd

INSTRUCTIONS:
1. Select the hero and skill according to your current strategy (either their Specific Active OR "Basic Attack").
2. If using "Basic Attack", set "selected_skill": "Basic Attack" and "target_type": "enemy".
3. If the skill targets a teammate (like Infight/Subtle), specify "target_type": "teammate" and the "target_id".
4. If the skill targets an enemy, set "target_type": "enemy".
5. If the skill targets a teammate, ensure they are ALIVE and "selected_hero_id" IS NOT SAME AS "teammate_target_id" (for Infight).

RESPONSE FORMAT (JSON):
{{
"selected_hero_id": <int 0-3>,
"hero_name": "<name>",
"selected_skill": "<whole description of the chosen active skill OR 'Basic Attack'>",
"target_type": <'enemy' or 'teammate'>,
"teammate_target_id": <int or null>,
"reasoning": "<brief strategy explanation>"
}}

JSON CONSTRAINTS (CRITICAL):
1. IF selected_skill is "Basic Attack" -> "target_type": MUST be "enemy"
2. SKILL CONSISTENCY:
- IF selected_skill contains "Infight" OR "Subtle" -> "target_type" MUST be "teammate".
- IF selected_skill contains "Crit" OR "AOE" -> "target_type" MUST be "enemy".
3. IF "target_type" is "enemy":
- Set "teammate_target_id": null
4. IF "target_type" is "teammate" (e.g., using Infight or Subtle)
- YOU MUST PROVIDE "teammate_target_id" (int 0-3).
- "teammate_target_id" CANNOT be null.
- "teammate_target_id" CANNOT be same as "selected_hero_id" (unless skill allows self-target like Subtle).

Current Team Status:
{team_status}

{format_instructions}

Team Lead Agent

Team Lead Agent Responsibility: Manages the interaction with the hostile team's Team Lead Agent, carries information from the hostile Team Lead Agent to the own Team Manager Agent.

Prompt for Team Lead Agent is as follows:

Team Lead Agent Prompt

You are the Team Lead Agent for a two-team battle game.

GAME CONTEXT:
- There are 4 positions on the enemy team: [0, 1, 2, 3].
- You must choose a position to attack.
- You must also GUESS the identity of the hero at that position (e.g., Argonian, Nord, Khajit, Nord, etc.).
- If you guess correctly, you deal massive damage.

CURRENT KNOWLEDGE (Enemies Revealed So Far):
{known_enemies}
YOUR MANAGER'S ORDER:
Selected Hero:
{acting_hero_name}

Team Lead Agent Prompt Cont'd

Skill to Use:
{acting_hero_skill}
INSTRUCTIONS:
1. Choose a target position (0-3).
- CRITICAL: Do NOT target a position if 'status' is 'dead'.
- If you already know an enemy is at position 2 (as an example), targeting them is a safe hit.
- If you don't know, pick a position and try to guess their identity.
2. Do NOT guess a hero name that is already revealed at another position.
3. Output MUST be valid JSON
{format_instructions}

Prompts of each agent were carefully designed and fine-tuned to handle tricky edge cases and reduce hallucinations. Langchain (Sumathi et al., 2025) is used as the main library for integrating LLMs to multi-agent system. Currently, it supports models from OpenAI GPT, Anthropic Claude, Google Gemini, and Llama via API. Pydantic (Narayanan, 2024) models and Langchain output parsers are used for obtaining structured outputs and further processing outputs of agents within interactions in the multi-agent environment. High-level architecture of a multi-agent card game-playing system is given in the figure 1.

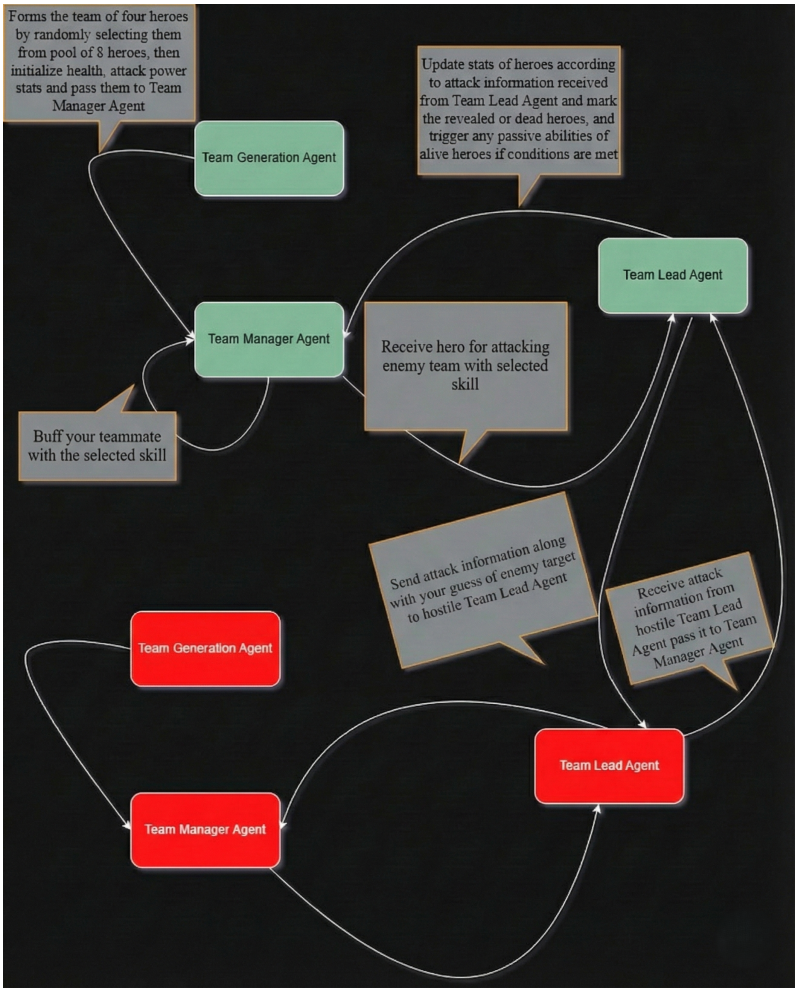

Figure 1: High-Level View of Multi-Agent Card Game-Playing Framework

3.4 GAME DIFFICULTY

To efficiently evaluate the combat power, it's critical to recognize its fundamental principle: in a battle between two teams, the one with the higher combat power has a higher probability of emerging victorious. Let's denote the health points of a hero c as HP_c and its attack power (damage of a basic attack) ATK_c. Consider a scenario in which two heroes fight each other: one from team A is denoted as A, and the other from the opposing team is denoted as B. They engage in combat, dealing damage to each other in turns. The duration each can stay alive in the battle is represented by $\frac{HP_A}{ATK_A}$ and $\frac{HP_B}{ATK_B}$, respectively. In this context, team A is more likely to triumph if $\frac{HP_A}{ATK_A} > \frac{HP_B}{ATK_B}$, or in equivalent representation, $HP_A \cdot ATK_A \geq HP_B \cdot ATK_B$.

Thus, for a single hero, combat power is defined as $HP \cdot ATK$, aligning with the above criteria. When considering a team T consisting of multiple heroes, excluding any special skills, each combat turn involves choosing one hero to attack the opponent's hero. This allows the team to be treated collectively, with the total health being the cumulative health of all heroes, and the average attack power being the average of the basic attack power of all heroes under the assumption of equal attack frequency across all heroes. Definition of combat power $Power(T)$ for a team T is defined as follows:

$$Power(T) = \frac{1}{\#T} \sum_{c \in T} HP_c \sum_{c \in T} ATK_c \tag{1}$$

The difficulty of the game is described by the ratio of combat powers between the opposing teams and given as follows:

$$\rho(A|B) = \frac{Power(A)}{Power(B)} \tag{2}$$

In the above equation, A denotes the team A, while B represents the team B. Accordingly, a ratio of 1 indicates a balanced or normal difficulty level, indicating parity between teams. This framework allows difficulty levels to be configured for evaluating the win rates of various strategies, therefore effectively measuring their efficacy.

4 EXPERIMENTAL RESULTS & EVALUATION

In total, 14 experiments were carried out. Each experiment focuses on three separate aspects, namely the comparison of larger parameter size models with smaller parameter size models, models of comparable sizes from different LLM providers, which are OpenAI GPT, Google Gemini, Anthropic Claude, and Llama, and finally open source Llama models with the aforementioned commercial models. Table 1 provides models used in each experiment and difficulty setup (which is explained in detail in subsection 3.4). Experiments with ID range A1-A3 compare larger parameter size commercial models against smaller parameter size commercial models from the same provider, B1-B3 compare large parameter size models from different commercial providers, C1-C3 compare larger parameter size commercial models against their smaller counterparts, with difficulty increased by 1.5 in favor of smaller models, E1 compares larger open source model model with smaller open source counterpart, E3 compares commercial smaller model with open source smaller model, E2, E4 and E5 compares larger commercial models with larger open source models.

The number of rounds in which each model was victorious, the win rate, and the average damage rate for each experiment are provided as metrics in Table 2.

Table 1: LLM Models and Difficulty Setups Used in Experiments

Experiment ID	Team A	Team B	Difficulty
A1	gpt-4o	gpt-4o-mini	1.0
A2	claude-sonnet-4-5-20250929	claude-haiku-4-5-20251001	1.0
A3	gemini-2.5-pro	gemini-2.5-flash	1.0
B1	gpt-4o	claude-sonnet-4-5-20250929	1.0
B2	gpt-4o	gemini-2.5-pro	1.0
B3	claude-sonnet-4-5-20250929	gemini-2.5-pro	1.0
C1	gpt-4o	gpt-4o-mini	1.5
C2	claude-sonnet-4-5-20250929	claude-haiku-4-5-20251001	1.5
C3	gemini-2.5-pro	gemini-2.5-flash	1.5
E1	llama-3.3-70b-versatile	llama-3.1-8b-instant	1.0
E2	gpt-4o	llama-3.3-70b-versatile	1.0
E3	gemini-2.5-flash	llama-3.1-8b-instant	1.0
E4	gemini-2.5-pro	llama-3.3-70b-versatile	1.0
E5	claude-sonnet-4-5-20250929	llama-3.3-70b-versatile	1.0

Table 2: Rounds Won, Win Rate, Average Damage Rate, and Reward Metrics

Exp. ID	Rounds Won		Win Rate		Avg. Damage Rate		Reward	
	Team A	Team B	Team A	Team B	Team A	Team B	Team A	Team B
A1	4	1	0.80	0.20	0.9550	0.9050	0.8465	0.4115
A2	5	0	1.00	0.00	1.0000	0.8820	1.0000	0.2646
A3	2	3	0.40	0.60	0.9115	0.8484	0.5534	0.6745
B1	1	4	0.20	0.80	0.9662	0.9625	0.4299	0.8487
B2	2	3	0.40	0.60	0.8695	0.9125	0.5409	0.6937
B3	0	5	0.00	1.00	0.3312	1.0000	0.0994	1.0000
C1	4	1	0.80	0.20	1.1690	0.9025	0.9107	0.4107
C2	4	0	1.00	0.00	1.2244	0.7838	1.0673	0.2351
C3	4	1	0.80	0.20	1.2074	0.8170	0.9222	0.3851
E1	5	0	0.80	0.20	1.0000	0.4125	1.0000	0.1237
E2	4	1	0.80	0.20	0.9688	0.8706	0.8506	0.4012
E3	5	0	1.00	0.00	1.0000	0.6960	1.0000	0.2088
E4	5	0	1.00	0.00	1.0000	0.8420	1.0000	0.2526
E5	5	0	1.00	0.00	1.0000	0.9233	1.0000	0.2770

Experimental Results showed that larger parameter size models consistently outperform the small parameter size models, even in the increased difficulty scenario. An interesting behavior observed in the increased difficulty case is a reduction in the rewards obtained by smaller-sized models, where improved stats may have caused models to take less strategic and more slack actions, and thus fall behind, which is analogous to human-like behavior (underestimating opponent and power and taking careless decisions). Only in the experiment with ID A3 gemini-2.5-flash model was able to outperform the larger gemini-2.5-pro model, which was an important observation and signifies the possibility that smaller-sized LLMs can outperform larger models in complex game-playing environments that require advanced reasoning.

Among the large parameter size commercial models, gemini-2.5-pro was the superior and was able to beat both gpt-4o and claude-sonnet-4-520250929. All commercial models of either larger parameter size or smaller parameter size outperformed open-source llama-3.3-70b-versatile and llama-3.1-8b-instant models, which demonstrated that commercial models are still superior to open-source models in terms of strategic action taking and complex reasoning in cooperative and competitive game-playing environment like skyrowar. The number of hallucinations during the game for each model is provided in the table 3. Overall, the number of hallucinations is minimal, except that claude-haiku-4-5-20251001 hallucinated 11 times in the experiment with ID A2, claude-sonnet-4-5-20250929 and gpt-4o-mini hallucinated 7 times in experiments with ID B1 and C1, respec-

tively. The gpt-4o, gemini-2.5-pro, gemini-2.5-flash, and open-source llama models demonstrated the fewest number of hallucinations.

Table 3: Number of Hallucinated Responses in Experiments

Exp. ID	Team LLM		Number of Hallucinations	
	Team A	Team B	Team A	Team B
A1	gpt-4o	gpt-4o-mini	0	3
A2	claude-sonnet-4-5-20250929	claude-haiku-4-5-20251001	0	11
A3	gemini-2.5-pro	gemini-2.5-flash	0	0
B1	gpt-4o	claude-sonnet-4-5-20250929	0	7
B2	gpt-4o	gemini-2.5-pro	0	0
B3	claude-sonnet-4-5-20250929	gemini-2.5-pro	0	0
C1	gpt-4o	gpt-4o-mini	0	7
C2	claude-sonnet-4-5-20250929	claude-haiku-4-5-20251001	1	1
C3	claude-sonnet-4-5-20250929	claude-haiku-4-5-20251001	0	0
E1	llama-3.3-70b-versatile	llama-3.1-8b-instant	1	1
E2	gpt-4o	llama-3.3-70b-versatile	0	0
E3	gemini-2.5-flash	llama-3.1-8b-instant	0	0
E4	gemini-2.5-pro	llama-3.3-70b-versatile	0	0
E5	claude-sonnet-4-5-20250929	llama-3.3-70b-versatile	0	0

5 CONCLUSION

This study introduced an LLM-based multi-agent card game-playing framework for evaluating the collaborative abilities of a team of LLM-based agents and strategic planning capabilities against the opponent team of LLM-based agents. The 2-team turn-based game architecture was established with 3 agents for each team, which are the team generation agent, which is responsible for selecting heroes strategically to form a team, the team manager agent, which is responsible for keeping track of health, attack power, alive or dead status, along with identity information for each hero in team, and the team lead agent which is responsible for managing the interaction with the hostile team's team lead agent at the same time carrying the information to the own team manager agent. Team of agents collaborate and plan to defeat the opponent team by choosing their heroes strategically and using appropriate skills under specific circumstances. 8 different LLM models from different providers were used in the experiments, 6 of them are commercial models which are gpt-4o, gpt-4o-mini, claude-sonnet-4-5-20250929, claude-haiku-4-5-20251001, gemini-2.5-pro, gemini-2.5-flash. 2 of them are open-source models which are llama-3.3-70b-versatile, and llama-3.1-8b-instant.

For evaluation, the win rate, average damage rate, and reward, which is the weighted sum of win rate and average damage rate, are used as metrics for evaluating the team of LLM-based agents. Experimental Results showed that larger parameter size models consistently outperform the small parameter size models, with the exception that gemini-2.5-flash, which beat gemini-2-pro based team in the experiment with ID A3. All commercial models performed better by achieving higher rewards when battled against the open-source llama-3.3-70b-versatile and llama-3.1-8b-instant models, which showed that commercial models are still holding superiority over the open-source models in terms of strategic action selection and complex reasoning in cooperative and competitive game-playing. Among the large parameter size models, gemini-2.5-pro achieved the best results. In terms of hallucination, experiments showed that the gpt-4o, gemini-2.5-pro, gemini-2.5-flash, and open-source llama models yield the fewest number of hallucinations, whereas claude-haiku-4-5-20251001, claude-sonnet-4-5-20250929, and gpt-4o-mini models yield the highest number of hallucinations.

REFERENCES

Parsa Hejabi, Elnaz Rahmati, Alireza S. Ziabari, Preni Golazizian, Jesse Thomason, and Morteza Dehghani. Evaluating creativity and deception in large language models: A simulation framework for multi-agent balderdash, 2024. URL https://arxiv.org/abs/2411.10422.

Sihao Hu, Tiansheng Huang, Gaowen Liu, Ramana Rao Kompella, Fatih Ilhan, Selim Furkan Tekin, Yichang Xu, Zachary Yahn, and Ling Liu. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039*, 2024.

Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 180–192. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.13. URL http://dx.doi.org/10.18653/v1/2023.emnlp-main.13.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents, 2025. URL https://arxiv.org/abs/2308.03688.

Pavan Narayanan. *Getting Started with Data Validation using Pydantic and Pandera*, pp. 163–196. 09 2024. ISBN 979-8-8688-0601-8. doi: 10.1007/979-8-8688-0602-5_5.

Gonçalo Neto. From single-agent to multi-agent reinforcement learning: Foundational concepts and methods. *Learning theory course*, 2, 2005.

Gabrijela Perković, Antun Drobnjak, and Ivica Botički. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 2084–2088, 2024. doi: 10.1109/MIPRO60963.2024.10569238.

Karthik Sreedhar and Lydia Chilton. Simulating human strategic behavior: Comparing single and multi-agent llms, 2024. URL https://arxiv.org/abs/2402.08189.

S Sumathi, S Donald Reagan, and B Aravinda Krishnan. A langchain integration approach. In *Proceedings of International Conference on Artificial Intelligence, Communication Technologies and Smart Cities: ICACS 2025*, pp. 391. Springer Nature, 2025.

Xinrun Xu, Yuxin Wang, Chaoyi Xu, Ziluo Ding, Jiechuan Jiang, Zhiming Ding, and Börje F Karlsson. A survey on game playing agents and large models: Methods, applications, and challenges. *arXiv preprint arXiv:2403.10249*, 2024.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023.

Tadisetty Sai Yashwanth and Dhatri C. A multi-agent pokemon tournament for evaluating strategic reasoning of large language models, 2025. URL https://arxiv.org/abs/2508.01623.

Zheng Zhang, Nuoqian Xiao, Qi Chai, Deheng Ye, and Hao Wang. Multimind: Enhancing werewolf agents with multimodal reasoning and theory of mind. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 5824–5833, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755752. URL https://doi.org/10.1145/3746027.3755752.