POKÉLLMON: A Human-Parity Agent for Pokémon Battle with Large Language Models

Anonymous Authors[1]

Abstract

We introduce POKÉLLMON, the first LLM-embodied agent that achieves human-parity performance in tactic battle games. It incorporates three key strategies: 1) In-context reinforcement learning that consumes text described feedback instantly derived from battles to iteratively refine its generation policy; 2) Knowledge-augmented generation that employs external knowledge to counteract hallucination and enables the agent to act timely and properly; 3) Action generation with self-consistency to mitigate the *panic switching* phenomenon when the agent faces a powerful opponent and want to avoid the battle. Online battle against human players demonstrate POKÉLLMON's human-level battle performance and strategies, achieving 49% of wining rate in the ladder competitions and 56% of wining rate in the invited battles. We also unveil its vulnerabilities to human players' attrition strategies and deception tricks. Our implementation and replayable battle logs are available at: xxx.

1. Introduction

Recent has witnessed significant success in LLMs on NLP tasks (), yet it has been less studied that the ability of LLMs autonomously acting in the physical world and interacting with humans with extended generation space from text to action, a pivotal paradigm in the pursuit of Artificial General Intelligence. Games are suitable test-beds to develop LLM-embodied agents to interact with the virtual environment in a way resembling human behavior. For example, Generative Agents (Park et al., 2023) conducts a social experiments with LLMs assuming various roles in a "the sims"-like sandbox, where agents exhbit behavior and social interactions mirroring humans. In Mincraft, life-long-learning agents (Wang et al., 2023a;b) are designed to explore the world and develop new skills to solve tasks and make tools.

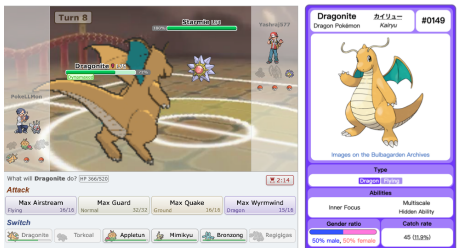

Figure 1. At each turn, the player is requested to decide which action to perform, *i.e.*, whether to let *Dragonite* to take a move or switch to another pokémon off the field.

Compare to existing games, tactic battle game is more suitable for benchmarking the game-playing ability of LLMs because the wining rate can be directly measured and invariant opponents like AI or human players are always available. Pokémon battle, as a mechanism for evaluating the battle ability of pokémon trainers in well-known Pokémon games, has several unique advantages as the first attempt for LLMs to play tactic battle games:

(1) The state and action spaces are discrete and can be translated into text losslessly. Figure 1 is an illustrative example for a Pokémon battle: At each turn, the player is requested to select an action to perform given the current battle state, which is the information of pokémon from each side. The action space consists of four moves and five possible pokémon to switch; (2) Its turn-based format eliminates the demands of intensive gameplay, alleviating the stress on the inference time cost for LLMs. Therefore, performance hinges solely on the reasoning abilities of LLMs; (3) Despite its seemingly simple mechanism, Pokémon battle is strategic and complex: To make a good decision, a player should take various factors into consideration, including species/type/ability/stats/item/moves of all the pokémon on and off the field. In a random battle, each pokémon is randomly selected from a large candidate pool (more than 1,000) with distinct characteristic, demanding the players both the Pokémon knowledge and reasoning ability.

Scope and Contributions: The scope of this paper is to develop an LLM-embodied agent that mimics the way a

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

human player engages in Pokémon battles. The objective is to explore the key factors that make the agent a good player and to examine its strengths and weaknesses in battles against disciplined human players.

To enable LLMs play battles autonomously, we implement a battle environment that can parse and translate logs into text described state, and send generated action to the server for execution. By evaluating existing LLMs, we observe the existence of hallucination: LLMs can mistakenly send out pokémon at a type disadvantage or use ineffective moves against the opposing pokémon. As a result, the most advanced LLM, GPT-4, scores a winning rate of 26% when playing against a heuristic bot, compared to nearly 60% of human players.

We introduce two strategies to combat hallucination: (1) In addition to the state description, we provide LLMs with text described feedback for the previous actions like the change of HP in consecutive two turns, the effectiveness of moves and the priority of taking moves, *etc.*, which are signals that can be easily perceived by human eyes from the animation and plays a role as "reward" to refine the generation policy; (2) We augment the generation of the agent by leveraging external knowledge like type advantage relationship and move/ability descriptions, simulating a human player searching for the information of unfamiliar pokémon and effects of moves/abilities.

We discover the *panic switching* phenomenon: when the agent faces a powerful opposing pokémon, it starts to generate inconsistent actions by switching to different pokémon in consecutive turns to avoid battle, wasting the chance of attacking. We identify that existing approaches like Chain-of-Thought (Wei et al., 2022) (CoT) can even deteriorate this problem by generating thoughts with panic feelings, while Self-Consistency (Wang et al., 2022) (SC) alleviates the problem by voting out the most consistent action without over-thinking. This observation mirrors human behavior, where in stressful situations, over-thinking and exaggerating difficulties can lead to panic and impede acting.

PokéLLMon battles against human players online and demonstrates human-like battle performance and strategies: it is good at taking effective moves against different pokémon and exhibits human-like attrition strategy. At the meantime, it is vulnerable to break the attrition strategy of human players, which requires long-term planing ability, and susceptible to tricks of experienced players.

In summary, this paper makes four original contributions:

- We implement a battle environment for LLMs to autonomously play pokémon battles.
- We propose in-context reinforcement learning to iteratively refine the generation policy and knowledge-augmented generation to combat hallucination.
- We discover that the agent can generate inconsistent actions because of panic feelings when facing powerful opponents, and CoT can deteriorate the phenomenon.
- PokéLLMon, the first LLM-embodied agent for tactic battle games, achieves human-level performance in online battle against disciplined human players.

2. LLMs as Game Players

Communicative games: Communicative games revolve around communication, deduction and sometimes deception between players. Existing studies show that LLMs demonstrate strategic behaviors in board games like Werewolf (Xu et al., 2023), Avalane (Light et al., 2023), WorldWar II (Hua et al., 2023) and Diplomacy (Bakhtin et al., 2022).

Open-ended games: Open-ended games allow players to freely explore the game world and interact with others. Generative Agent (Park et al., 2023) provides a social experiment with LLMs assuming various roles in a "the sims"-like sandbox, where agents exhibit behavior and social interactions mirroring human-like patterns. In MineCraft, Voyager (Wang et al., 2023a) employs curriculum mechanism to explore the world and generates and executes code for solving tasks. DEPS (Wang et al., 2023b) proposes an approach of "Describe, Explain, Plan and Select" to accomplish 70+ tasks. Planing-based frameworks like AutoGPT (Significant Gravitas) and MetaGPT (Hong et al., 2023) can be adopted for the exploration task as well.

Tactic battle games: Among existing game types, tactic battle game is the most suitable type to benchmark LLMs' game-playing ability since the wining rate can be directly measured, and invariant opponents like AI or human players are always available. Recently, LLMs are employed to play StarCraft II (Ma et al., 2023) against the built-in AI with a text-based interface and a chain-of-summarization approach. In comparison, Pokémon battle has several unique advantages: (1) Translating pokémon battle state into text is lossless; (2) Pokémon battle is strategic due to numerous pokémon species and move effects. Battling against disciplined human players elevates the difficulty to a new height; (3) The battle is turn-based without real-time stress given the inference time cost of LLMs, making the performance solely relies on the reasoning ability of LLMs.

3. Background

3.1. Pokemon

Species: There are more than 1,000 pokémon species (bul, 2024c), each with its unique ability, type(s), statistics (stats) and battle moves. Figure 2 shows two representative pokémon: *Charizard* and *Venusaur*.

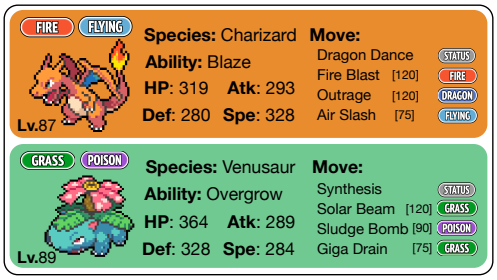

Figure 2. Two representative pokémon: *Charizard* and *Venusaur*. Each pokémon has type(s), ability, stats and four battle moves.

Type: Each pokémon species has up to two elemental types, which determines its advantages and weaknesseses. Figure 3 shows the advantage/weakness relationship between 18 types of attack move and attacked pokémon. For example, fire-type moves like "Fire Blast" of *Charizard* can cause double damage to grass-type pokémon like *Venusaur*, while *Charizard* is vulnerable to water-type moves.

Stats: Stats determine how well a pokémon performs in battles. There are four stats: (1) Hit Points (HP): determines the damage a pokémon can take before fainting; (2) Attack (Atk): affects the strength of attack moves; (3) Defense (Def): dictates resistance against attacks; (4) Speed (Spe): determines the order of moves in battle.

Ability: Abilities are passive effects that can affect battles. For example, *Charizard*'s ability is "Blaze", which enhances the power of its Fire-type moves when its HP is low.

Move: A pokémon can use four moves in battles, which can be categorized as attack moves or status moves. An attack move can cause instant damage with a power value and an accuracy, and associated with a specific type, which often correlates with the pokémon's type but does not necessarily align. In comparison, status moves do not cause instant damage but affect the battle in various ways including altering stats, healing or protect pokémon, changing status or battle conditions, *etc*.

There are 919 moves in total and each has its distinctive effect (bul, 2024b). A move's effectiveness depends on many factors like types, stats, abilities and moves of pokémon both-side and some special conditions like weather, *etc*.

3.2. Battle Rule

In one-to-one random battles, two battlers face off, each with six randomly selected pokémon. Initially, each battler sends out one pokémon on the field, while keeping the others off the field for future switches. The objective is to make all of the opponent's pokémon faint (by reducing their HP to zero) while preserving at least one of own pokémon remains unfainted. The battle is turn-based: at the beginning of each turn, both players simultaneously choose an action to perform. Actions fall into two categories: (1) taking moves

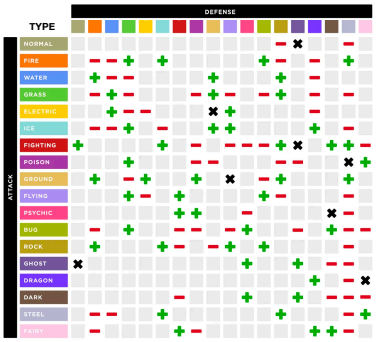

Figure 3. Type advantage/weakness relationship. "+" denotes super-effective (2x damage); "−" denotes ineffective (0.5x damage); "×" denotes no effect (0x damage). Unmarked is standard (1x) damage.

or (2) switching to another pokémon. The battle engine then executes these actions and update the battle state for the next turn. If a pokémon is fainted after a turn and the battler has other unfainted pokémon, the battle engine forces the battler to switch (a forced switch), which does not consume the player's action within the next turn. After a forced switch, the player still can choose a move or making another switch.

4. Battle Environment

Battle Engine: The environment interact with a battle engine server called Pokémon showdown (pok, 2024), which provides a web GUI for human to interact, and also web APIs to receive state message or send action message in defined formats.

Battle Environment: We implement a battle environment based on (Sahovic, 2023a) to support LLMs battle against players. Figure 4 is an illustrative example of how the entire framework works. At the beginning of a turn, the environment get an action-request message from the server, including the execution result from the last turn. We first parse the message and update local state variables, and then translate the state variables into text. The text description primarily consists of four parts: (1) own team information, including the attributes of pokémon on-the-field and its move set for selection, and the unfainted pokémon off-the-field for switching; (2) Opponent team information including the attributes of opposing pokémon on-the-field and off-the-field; (3) Battle field information like the weather, entry hazard and terrain; (4) Historical turn log information, including actions of both side pokémon from the previous turns, which is stored in a log queue.

LLMs take the translated state description as input and output an action for the next turn. The action is sent to the server and executed alongside the action chosen by the human player.

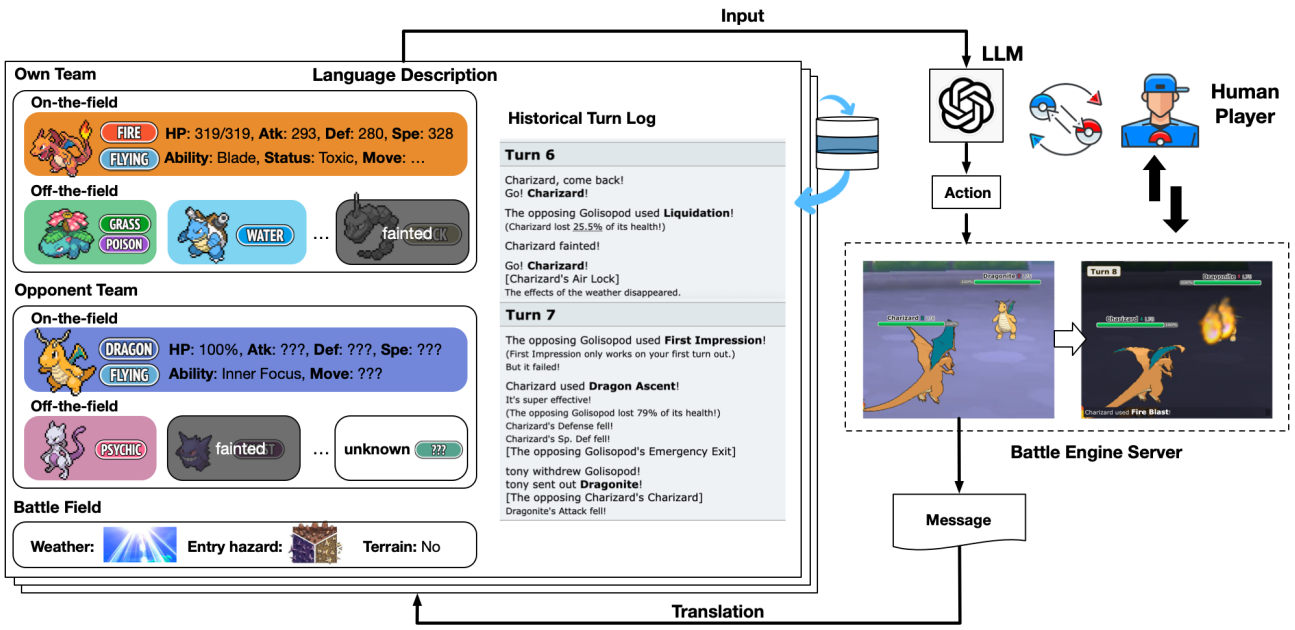

Figure 4. The battle environment that enables LLMs to battle with human players: It parses the messages received from the battle server and translates state logs into text descriptions. LLMs take these state descriptions and historical turn logs as input and generates an action for the next step. The action is then sent to the battle server and executed alongside the action chosen by the opponent player.

5. Preliminary Evaluation

In this section, we evaluate the battle abilities of existing LLMs, including GPT-3.5 (Ouyang et al., 2022), GPT-4 (Achiam et al., 2023), and LLaMA-2 (Touvron et al., 2023), to gain insights into the challenges associated with this task.

5.1. Pokemon Battle

Directly letting LLMs battle against human players is time-consuming since human needs time to think (4 minutes for 1 battle in average). To save time, we adopt a heuristic bot (Sahovic, 2023b) to initially battle with human players for ladder competitions, and then use the bot to benchmark existing LLMs. The bot is programmed to use status boosting moves, set entry hazards, selecting the most effective actions by considering the stats of pokémon, the power of moves, and type advantages/weaknesses.

The statistic results are presented in Table 1, where the battle score is defined as the sum of the numbers of the opponent's fainted pokémon and the player's unfainted pokémon at the end of a battle. Consequently, the opponent player's battle score is equal to 12 minus the player's battle score. Random is a simple strategy that randomly generates an action at every time and MaxPower chooses the move with the largest power value. Obviously, GPT-3.5 and LLaMA-2 are just slightly better than Random and even GPT-4 cannot beat the bot, let along well-disciplined human players from ladder competitions.

Table 1. Performance comparison against a heuristic bot.

Player	Win. rate ↑	Score ↑	Turn #	Battle #
Human	59.84%	6.75	18.74	254
Random	1.2%	2.34	22.37	200
MaxPower	10.40%	3.79	18.11	200
LLaMA-2	8.00%	3.47	20.98	200
GPT-3.5	4.00%	2.61	20.09	100
GPT-4	26.00%	4.65	19.46	100

By observing LLMs play battles and requesting explanations for their actions, we noticed the occurrence of hallucination: LLMs can mistakenly claim non-existent type-advantage relationships or, even worse, reverse the advantage relationships between types. A clear understanding of type advantage/weakness plays is crucial in pokémon battles, as choosing a pokémon with a type advantage can result in dealing more damage and sustaining less.

5.2. Test of Hallucination

Table 2. Confusion matrices for type advantage prediction.

Model	LLaMA-2				GPT-3.5				GPT-4			
Class	A	B	C	D	A	B	C	D	A	B	C	D
A	5	46	0	0	0	0	49	2	37	8	5	1
B	25	179	0	0	2	6	185	11	0	185	17	2
C	15	46	0	0	0	2	57	2	3	24	32	2
D	1	7	0	0	0	0	7	1	0	0	0	8

To test the hallucination of LLMs, we construct a task named type advantage/weakness prediction by asking LLMs to output whether it is A. super-effective (2x damage), B. standard (1x damage), C. ineffective (0.5x damage) or D. no

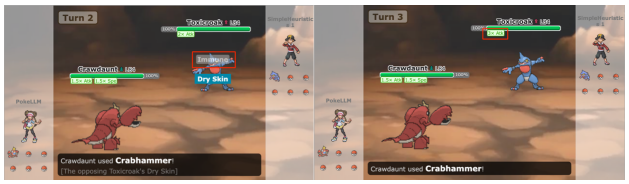

Figure 5. An example that the agent repeatedly uses the attack move "Crabhammer" in turn 1 and turn 2, which was zero effect to the opposing pokémon due to its ability "Dry Skin." Consequently, this error provides the opponent with two free turns to boost it's attack to threefold.

effect (0x damage), given a pair of attack move type and defense pokémon type. The 324 (18x18) testing pairs are constructed based on Figure 3.

Table 2 shows the three confusion matrices of LLMs, where their performance is highly related to their wining rates in Table 1. LLaMA-2 and GPT-3.5 suffer from severe hallucination problems, while GPT-4 achieves the best performance with an accuracy of 84.0%, we still observe it frequently making ineffective actions, which is because in a single battle, LLMs need to compare the types of all the opponent's pokémon with types of all their pokémon, as well as types of moves.

6. PokéLLMon

We introduce three strategies that boosts the battle abilities of GPT-4 to match the competitive level of human players.

6.1. In-Context Reinforcement Learning (ICRL)

Human players make decisions based not only on the current state but also on the (implicit) feedback of previous actions, such as the change in a pokémon's HP over two consecutive turns after it is attacked by a move. Without these feedback provided, the agent might stick to the same erroneous action for many turns. Figure 5 provides an illustrative example: In turn 2, the agent chooses to use "Crabhammer", a water-type attack move against the opposing *Toxicroak*, a pokémon with the ability "Dry Skin", which nullifies damage from water-type moves. The "Immune" message displayed in the battle animation can prompt a human player to change actions, even without knowledge of "Dry Skin", however, is not included in the state description and historical actions. As a result, in turn 3, the agent still generates the same action, giving the opponent two free turns to triple the attack stats of *Toxicroak*, leading to defeat ultimately.

Existing reinforcement learning (Schulman et al., 2017; Mnih et al., 2016) uses numeric rewards to evaluate actions and refine the policy. Since LLMs can understand languages and distinguish what is good and bad, feedback described in text provides a new form of "reward": by inserting text feedback for the previous actions, the agent is able to refine

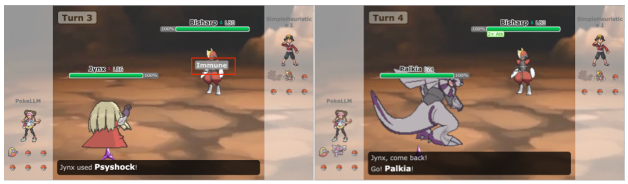

Figure 6. In turn 2, the agent uses "Psyshock", which cause zero damage to the opposing pokémon. With feedback provided, the agent switch to another pokémon.

Table 3. Improvement of ICRL against the heuristic bot.

Player	Win. rate ↑	Score ↑	Turn #	Battle #
Human	59.84%	6.75	18.74	254
Origin	26.00%	4.65	19.46	100
ICRL	36.00%	5.25	20.64	100

its "policy" without explicit training, namely In-Context Reinforcement Learning (ICRL).

In practice, we generate four types of feedbacks: (1) The change in HP over two consecutive turns, which reflects the actual damage caused by an attack move; (2) The effectiveness of the attack move, i.e., whether it is super-effective, ineffective, or has zero effect (immunity) because of type advantages and ability/move effects; (3) The priority of executing moves, providing a coarse estimation of speed, as we lack precise stats for the opposing pokémon; (4) The effects of moves: moves can deal damage and also cause additional effects like stat boosts or debuffs, inflict abnormal statuses such as being poisoned, burned, frozen, *etc.*, or alter the battlefield conditions.

Table 3 shows the improvement brought by ICRL. Compared to the original performance of GPT-4, the winning rate is boosted by 10%, and the battle score increases by 12.9%. During the battles, we observe that the agent begins to change its action if the moves in previous turns do not meet the expectation, as shown in Figure 6: After observing that the opposing pokémon is immune to the attack, it switches to another pokémon.

6.2. Knowledge-Augmented Generation (KAG)

Although ICRL is able to alleviate the hallucination, it can still cause fatal consequences before the feedback arrives. For example, if the agent sends out a grass-type pokémon to battle a fire-type pokémon, the former is likely be taken down in one single turn before the agent realize it is a bad decision. To alleviate hallucination, existing studies (Lewis et al., 2020; Patil et al., 2023) leverages external knowledge to augment generation. In this section, we introduce two types of external knowledge to further reduce hallucination.

Type advantage/weakness relationship: In the original state description in Figure 4, we annotate all the type information of pokémon and moves to let the agent infer the type advantage relationship by itself. To reduce the hallucination

contained in the reasoning, we explicitly annotate the type advantage and weakness of the opposing pokémon and our pokémon with descriptions like "*Charizard* is strong against grass-type pokémon yet weak to the fire-type moves".

Move/ability effect: Given the numerous moves and abilities with distinct effects, it is challenging for human players to memorize all these effects, let alone for LLMs that have never played Pokémon battles. For instance, it's difficult to infer the effect of a status move based solely on its name: "Dragon Dance" is a move that boosts the user's attack and speed by one stage, whereas "Haze" can reset the boosted stats of both Pokémon and remove abnormal statuses like being burnt. Even attack moves have various additional effects besides dealing damage. Therefore, we crawled all effect descriptions of moves, abilities from Bulbapedia (bul, 2024b;a), attaching the corresponding effect descriptions to the moves and abilities in the state description.

These external knowledge, serves as Pokédex, an electronic device functioned as an encyclopedia in Pokémon games, which provides comprehensive information for different pokémon species.

Table 4. Improvement of KAG against the heuristic bot.

Player	Win. rate ↑	Score ↑	Turn #	Battle #
Human	59.84%	6.75	18.74	254
Origin	36.00%	5.25	20.64	100
KAG[Type]	55.00%	6.09	19.28	100
KAG[Effect]	40.00%	5.64	20.73	100
KAG	58.00%	6.53	18.84	100

Table 4 shows the results of two knowledge-augmented generations, where knowledge-augmented generation with type advantage relationship (KAG[Type]) significantly boosts the winning rate from 36% to 55%. Move/ability effect descriptions also enhance the winning rate by 4 AP. By combining two types of knowledge, KAG achieves a winning rate of 58% against the heuristic bot, approaching a level competitive with human players.

In practice, we observe that the agent starts to use some very special status moves at proper time. As an example shown in Figure 7, a steel-type *Klefki* is vulnerable to the ground-type attack of the opposing *Rhydon*, a ground-type pokémon. Usually in such a disadvantage, the agent will choose to switch to another pokémon. However, it chooses to use the move "Magnet Rise", which has the effect that can makes the user immune to Ground moves for five turns. The bot uses a ground-type attack move "Earthquake" yet is invalidated by the effect of "Magnet Rise".

6.3. Reasoning

Existing studies (Wei et al., 2022; Yao et al., 2023; Wang et al., 2022) show that reasoning can improve the ability of

Figure 7. The agent understands the move effect and uses it properly: Klefki is steel-type and vulnerable to the ground-type attack of the opposing Rhydon. Instead of switching, the agent chooses to use the "Magnet Rise" move, which levitates Klefki to prevent it from the ground-type attack for five turns. In the same turn, the opposing Rhydon chooses the ground-type attack "earthquake", which is invalid due to the effect of "Magnet Rise".

LLMs on solving complex tasks. Instead of generating a one-shot action in each step, we evaluate existing reasoning approaches including Chain-of-Thought (Wei et al., 2022) (CoT), Self-Consistency (Wang et al., 2022) (SC) and Tree-of-Thought (Yao et al., 2023) (ToT). For CoT, the agent initially generates a thought that analyzes the current battle situation and output the action conditioned on the thought. For SC (k=3), the agent generates three times of actions and select the most voted answer as the output. For ToT (k=3), the agent generates three action options and picks out the best one evaluated by itself.

Table 5. Comparison of reasoning approaches against the bot.

Player	Win rate ↑	Score ↑	Turn #	Battle #
Human	59.84%	6.75	18.74	254
Origin	58.00%	6.53	18.84	100
CoT	54.00%	5.78	19.60	100
SC (k=3)	**64.00%**	6.63	18.86	100
ToT (k=3)	60.00%	6.42	20.24	100

Table 5 presents the comparison results of the original IO prompt generation and three algorithms. Notably, CoT results in a performance degradation by a 6 AP drop in the winning rate. In comparison, SC brings a performance improvement, with its winning rate surpassing that of human players. Beyond the comparative results, our greater interest lies in understanding the underlying reasons for these observations.

As introduced in Section 3.2, for each turn there is single action can be taken, which means if the agent chooses to switch yet the opponent choose to attack, the switch-in pokémon will sustain the damage. Usually switching happens when the agent decides to leverage the type advantage of an off-the-battle pokémon, and thus the damage taken is sustainable since the switch-in pokémon is typically type-resistant to the opposing pokémon's moves. However, when the agent with CoT reasoning faces a powerful opposing pokémon, its actions become inconsistent by switching to different pokémon in consecutive turns, which we call *panic switching*, wasting chances of taking moves and leading to the defeat. An illustrative example is shown in Figure 8: at

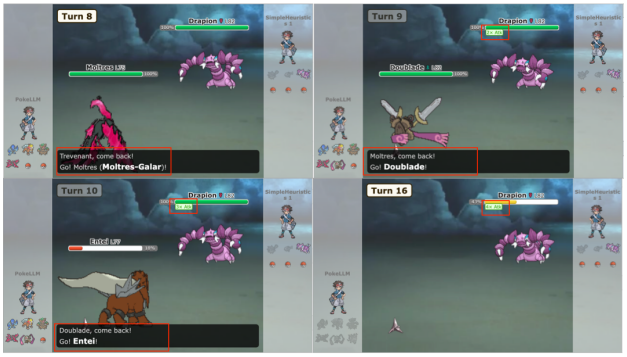

Figure 8. Panic switching: When facing a powerful pokémon, the agent with CoT switches pokémon in three consecutive turns because the generated thoughts contain feelings of panic, leading to inconsistent actions. This gives the opponent three free turns to quadruple its attack stats and quickly defeat the agent's entire team.

the beginning of turn 8, the agent has 6 unfainted pokémon whereas the opponent has 4. The agent chooses to continuously switch to different pokémon in three consecutive turns, giving the opposing pokémon three free turns to boost its attack stats to four times and take down the agent's entire team quickly.

Table 6. Statistic analysis for panic switching

Player	Win. rate ↑	Switch rate	CS1 rate	CS2 rate
Origin	58.00%	17.05%	6.21%	22.98%
CoT	54.00%	26.15%	10.77%	34.23%
SC (k=3)	64.00%	16.00%	1.99%	19.86%
ToT (k=3)	60.00%	19.70%	5.88%	23.08%

In Table 6 we provide statistical evidence, where CS1 represents the proportion of active switches where the last-turn action is a switch and CS2 rates represent the proportion of active switches here at least one action from the last two turns is a switch, among all active switches, respectively. The higher the CS1 rate, the greater the inconsistency of generation. Obviously, CoT largely increases the continuous switch rate, whereas, SC decreases the continuous switch rate.

Upon examining the thoughts leading to the panic-switch decision, we observe that the thoughts contain feelings of panic. The agent emphasizes the power of the opposing pokémon and describes the weaknesses of the current pokémon, ultimately deciding to switch to another pokémon, as in "*Drapion* has boosted its attack to two times, posing a significant threat that could potentially knock out *Doublade* with a single hit. Since *Doublade* is slower and likely to be knocked out, I need to switch to *Entei* because...". Action generation conditioned on panic thoughts leads it to continuously switch pokémon instead of attacking. In comparison, SC provides more consistent results by generating actions multiple times without over-thinking and voting out

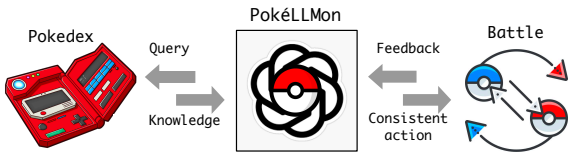

Figure 9. To summary, PokéLLMon is equipped with three strategies: (1) In-Context Reinforcement Learning (ICRL) leveraging feedbacks from the battle to refine the next action generation; (2) Knowledge-Augmented Generation (KAG) with type-advantage relationship and move/ability descriptions to alleviate hallucination; (3) Action generation with self-consistency to prevent the panic switching problem.

the most consistent action, decreasing the continuous switch rates.

The observation is reflecting: when humans face stressful situations, overthinking and exaggerating difficulties lead to panic feelings and paralyze their ability to take actions, leading to even worse situations.

Summary: As shown in Figure 9, PokéLLMon is equipped with three strategies: (1) ICRL leverages feedbacks derived from the battle states to refine the generation policy; (2) KAG augments the generation with external knowledge to combat hallucination and let the agent use moves properly; (3) Self-consistent generation solves the panic switching problem.

7. Online Battle

To evaluate the battle ability of PokéLLMon against human, we set up battles on Pokémon Showdown (pok, 2024), where the agent battled against randomly paired human players for 105 ladder competitions from Jan. 25 to Jan. 26, 2024. Besides, we invited an human player who has over 15 years of experience with Pokémon games, representing the average battle ability of human players to battle against PokéLLMon. All the replayable battle logs can be found at: xxx.

7.1. Battle Against Human Players

Table 7. Performance of PokéLLMon against human players.

v.s. Player	Win. rate ↑	Score ↑	Turn #	Battle #
Ladder Player	48.57%	5.76	18.68	105
Invited Player	56.00%	6.52	22.42	50

Table 7 presents the performance of the agent against human players. PokéLLMon demonstrates comparable performance to disciplined ladder players who have extensive battle experience, and achieves a higher winning rate than the invited player. The average number of turns in ladder competitions is lower because ladder players sometimes forfeit when they believe they will lose to save time.

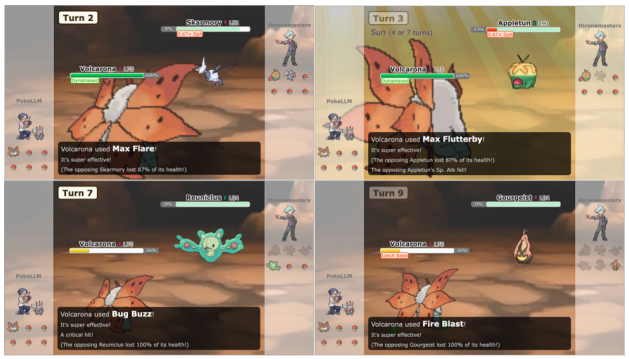

Figure 10. PokéLLMon selects effective moves in every turn, causing the opponent's entire team to faint using only one pokémon.

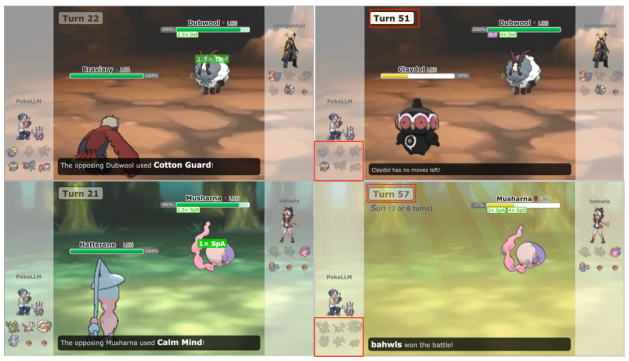

Figure 11. PokéLLMon suffers from the attrition strategy in two battles: the opponent players frequently recover high-defense pokémons. Breaking the dilemma requires a long-term plan and joint effects across many turns.

7.2. Battle Skill Analysis

Strength: PokéLLMon seldom make mistakes at choosing the effective move and switching to another suitable pokémon due to the KAG strategy. As shown in Figure 10, in one battle, the agent uses only one pokémon to cause the entire opponent team fainted by choosing different attack moves toward different pokémon.

Moreover, PokéLLMon exhibits human-like attrition strategy: With some special pokémon have the "Toxic" move that can inflict regular additional damage every turn and the "Recover" move that can recover 50% of HP, the agent starts to first poisoned the opposing pokémon and frequently uses the "Recover" to prevent itself from fainting. By prolonging the battle, the opposing pokémon's HP is gradually depleted by the poisoning damage. Employing this attrition strategy requires an understanding of moves like "Toxic", "Recover" and "Protect", as well as the right timing for their use (such as when there's no type-weakness or when having high defense). This nuanced tactic, relying heavily on specific conditions, is never played by the heuristic bot.

Weakness: PokéLLMon tends to take actions that can achieve short-term benefits, therefore, making it vulnerable to the human players' attrition strategy that requires long-term effort to break. As shown in the two battles in Figure 11, after many turns, the agent's entire team is defeated by the human players' pokémon, which have significantly boosted defense and engage in frequent recovery. Table 8 reports the performance of PokéLLMon in battles where human players either use the attrition strategy or not. Obviously, in battles without the attrition strategy, it outperforms ladder players, while losing the majority of battles when human play the attrition strategy.

Figure 12. An experienced human player misdirects the agent to use a dragon-type attack by firstly sending out a dragon-type pokémon and immediately switch to another pokémon immuned to the dragon-type attack.

Table 8. Battle performance impacted by the attrition strategy

Ladder	Win. rate ↑	Score ↑	Turn #	Battle #
w. Attrition	18.75%	4.29	33.88	16
w/o Attrition	53.93%	6.02	15.95	89

The "Recover" move recovers 50% HP in one turn, which means if an attack cannot cause the opposing pokémon more than 50% HP damage in one turn, it will never faint. The key to breaking the dilemma is to firstly boost a pokémon's attack to a very high stage and then attack to cause unrecoverable damage, which is a long-term goal that requires joint efforts across many turns. PokéLLMon is weak to the long-term planing because current design does not keep a long-term plan in mind across many timesteps, which will be included in the future work.

Finally, we observe that experienced human players can misdirect the agent to bad actions. As shown in Figure 12, our *Zygarde* has one chance to use an enhanced attack move. At the end of turn 2, the opposing *Mawile* is fainted, leading to a forced switch and the opponent choose to switch in *Kyurem*. This switch is a trick that lures the agent uses a dragon-type move in turn 3 because *Kyurem* is vulnerable to dragon-type attacks. In turn 3, the opponent switches in *Tapu Bulu* at the beginning, a pokémon immuned to dragon-type attacks, making our enhanced attack chance wasted.

The agent is fooled because it makes decision only based on the current state information, while experienced players condition on not only the state information, but also the opponent's next action prediction.

Seeing through tricks and predicting the opponent's next action require the agent being disciplined in the real battle environment, which is the future step in our work.

8. Conclusion

In this paper, we enable LLMs to autonomously play Pokémon battles with a dedicated battle environment. We introduce PokéLLMon, the first human-parity agent for Pokémon battles, featuring three strategies: In-Context Reinforcement Learning, which includes instant feedback to refine the generation of the next action; Knowledge-Augmented Generation that leverages external knowledge to alleviate the hallucination problem and ensures proper use of moves; Self-consistent action generation to prevent the issue of panic switching. PokéLLMon demonstrates human-level battle ability and strategies, yet it exhibits some weaknesses when facing experienced human players' tactics.

References

List of abilities, 2024a. URL https://bulbapedia.bulbagarden.net/wiki/Ability#List_of_Abilities.

List of moves, 2024b. URL https://bulbapedia.bulbagarden.net/wiki/List_of_moves.

List of pokémon by national pokédex number, 2024c. URL https://bulbapedia.bulbagarden.net/wiki/List_of_Pokmon_by_National_Pokdex_number.

Pokémon showdown, 2024. URL https://play.pokemonshowdown.com.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022.

Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Hua, W., Fan, L., Li, L., Mei, K., Ji, J., Ge, Y., Hemphill, L., and Zhang, Y. War and peace (waragent): Large language model-based multi-agent simulation of world wars. *arXiv preprint arXiv:2311.17227*, 2023.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Light, J., Cai, M., Shen, S., and Hu, Z. From text to tactic: Evaluating llms playing the game of avalon. *arXiv preprint arXiv:2310.05036*, 2023.

Ma, W., Mi, Q., Yan, X., Wu, Y., Lin, R., Zhang, H., and Wang, J. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *arXiv preprint arXiv:2312.11865*, 2023.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

Sahovic, H. Poke-env: pokemon ai in python, 2023a. URL https://github.com/hsahovic/poke-env.

Sahovic, H. poke-env: Heuristicbot, 2023b. URL https://github.com/hsahovic/poke-env/blob/master/src/poke_env/player/baselines.py.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Significant Gravitas. AutoGPT. URL https://github.com/Significant-Gravitas/AutoGPT.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Wang, Z., Cai, S., Liu, A., Ma, X., and Liang, Y. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Xu, Y., Wang, S., Li, P., Luo, F., Wang, X., Liu, W., and Liu, Y. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.