An Empirical Analysis of the Validity of Competitive Pokémon Rule Sets

Nicholas Fluty[1], Ryan D. Flores[1], Judy Goldsmith[1] and Brent Harrison[1]

[1]Department of Computer Science, University of Kentucky, Lexington, KY 40506-0633 USA

Abstract

In competitive Pokémon battling, players have adopted a set of extra rules that are meant to encourage fair play. They are used to constrain team formation so that no one team has an overwhelming advantage over all others. These rule sets are often derived based on trial and error, intuition, or post-hoc evaluations of team performance, which means that the rules may not be ideal solutions to the problem they are supposed to address, or the problem may not have been worth addressing.

In this paper, we explore how artificial intelligence and machine learning techniques can be used to potentially evaluate the quality of a rule set. This is meant to be a preliminary study that will ultimately lead to the automatic formulation of such rule sets. Our case study investigates how the inclusion or exclusion of one-hit-knock-out (OHKO) moves affects the outcomes and player behaviors in games between two teams battling under Generation 1 rules.

Keywords

Competitive Rules, Artificial Intelligence, Machine Learning

1. Introduction

Pokémon is a game in which players construct teams of combatants, the titular Pokémon, to battle against other players' teams. A great deal of thought is often put into how these teams are constructed, as one wants to utilize powerful Pokémon while promoting good synergy as a team. In order to ensure that a healthy competitive atmosphere is maintained, there are often rules put in place on how a team can be constructed. This is meant to ensure that strategies that are potentially too strong don't become prevalent as a part of the competitive metagame.

The rulesets Smogon uses to govern their competitive battles are good examples of this. Smogon is a competitive battling community that organizes tournaments, provides competitive battling resources, etc. In service of this, they also define rulesets that are used when these tournaments are held. These rules govern how players construct and use their teams and are meant to guard against overpowered or degenerate strategies. In addition to a set of rules common to all battles, Smogon defines various battle formats that restrict which Pokémon can be used to allow for diverse usage of both strong and weak Pokémon. The most commonly used format is reffered to as *OverUsed (OU)*, which allows all but some of the strongest "legendary" Pokémon that were intentionally given this advantage for purposes outside of competitive play. Again, these rulesets are meant to ensure that no one strategy for constructing teams or battling is strictly dominant over all others.

While these rules are often necessary for healthy competitive play, constructing these rulesets can be quite difficult. Often, these rules are based on speculation, anecdotal evidence, or post-hoc analysis. As such, the formation of effective rulesets is an imperfect science that can be time-consuming and prone to errors (constructing rules where there shouldn't be one or missing a rule that should be present).

In this paper, we investigate how machine learning (ML) techniques can be used to support the evaluation of competitive rule sets for the game of Pokémon. We are using the Pokémon domain to test this concept because of the existence of community tournaments that contain rules that exist outside of the game environment. Specifically, we present a case study in which we examine the Smogon rules associated with the first generation of the game, demonstrate how we can test changes, and present a data-based discussion of the effects of the change. We chose this ruleset because Smogon rules are largely community-driven and not necessarily subject to rigorous empirical analysis. The primary contribution of this work is to explore how AI and machine learning techniques can be used to perform vulnerability tests on these types of rulesets. This case study serves as preliminary evidence of the feasibility of such an approach, and we hope it will encourage further work in the area.

The remainder of the paper is organized as follows. In the next section, we review relevant related work on evaluating rule sets and metagame in Pokémon. We will then introduce the Smogon generation 1 tournament rule set. Finally, we will detail our case study and present the results of said study.

2. Related Works

The primary contribution of this work is in evaluating the rulesets associated with competitive play in games. Specifically, in this paper we evaluate how the rulesets associated with competitive Pokémon affect dominant teams. There has been past work that has examined the Pokémon metagame [1, 2], but that previous work examines what teams of Pokémon are particularly strong in a metagame as defined by the rules associated with competitive play or investigate countering the metagame [3]. In this paper, we examined whether these rules are justified and how one might prove them.

To do this, we take inspiration from automated playtesting literature and propose that machine learning techniques can be used to identify problems with rule sets or rules that are not justified. Typically, video game playtesting is performed by humans to determine whether a game contains errors. This process is time consuming and prone to human error. Thus, there has been an increased interest in

11th Experimental Artificial Intelligence in Games Workshop, November 19, 2024, Lexington, Kentucky, USA.

ndfl222@uky.edu (N. Fluty); rhma226@uky.edu (R. D. Flores); goldsmit@uky.edu (J. Goldsmith); bha286@g.uky.edu (B. Harrison)
0009-0007-3250-4915 (N. Fluty); 0009-0000-2790-5908 (R. D. Flores); 0000-0002-8383-5390 (J. Goldsmith); 0000-0002-1301-5928 (B. Harrison)

automating this process using artificial agents. In the past, researchers have explored several AI and machine learning methods for automating this process [4, 5, 6, 7]. Still, these approaches are typically done to evaluate level design or game mechanics with respect to designer goals. Other work has focused on better creating agents that can mimic playtesters of different personas [8] or skillsets [9], but still the primary focus is on evaluating game mechanics or level design. In this work, we focus on evaluating rules that are not inherent to the game itself, but that are designed after the fact to encourage competitive play.

When determining the effect that rule sets have on a team's win probability, we use machine learning to learn battle strategies for each team. Machine learning has been readily explored in Pokémon [10, 11, 12, 13, 14] and found to be an effective tool for teaching agents how to battle. The main difference between our work and this past work is not in the method, but in the motivation behind the method. These past works primarily focused on developing techniques to be more competent in battle. We are using these techniques in service of evaluating player-made rule sets.

3. Smogon Rule Sets

Smogon uses a modified rule set compared to the official tournaments hosted by Nintendo. The rules vary across the different Pokémon generations. These modified rules effect the formation of the teams as well as the actions a player can take during a battle. Every few years a vote is held on the Smogon forums to see whether or not any rules need to be updated or replaced.

The first set of rules dictates the team formation. The Species Clause prevents a player from having multiple of the same Pokémon on their team. This is to encourage more diversity and to prevent players from running teams of the same Pokémon. Next is the Evasion Clause, which prevents player from using the moves Double Team or Minimize. These moves make a Pokémon harder to hit, and can lead to stalled games where neither player can win. The final clause is the one-hit-knock-out (OHKO) Clause. Pokémon are not allowed to have the moves Horn Drill, Guillotine, Sheer Cold, or Fissure. These moves, referred to as OHKO moves, have a low hit rate but will cause the opponent's Pokémon to faint if they do hit. The general opinion on the forums is that this rule prevents strong players from losing due to randomness

The second set of clauses affects player behavior during the battles. The Sleep Clause prevents a player from directly causing more than one of their opponent's Pokémon to fall asleep at a time. If a move attempts to break this clause, the game will automatically prevent the sleep from occurring. The Freeze Clause is the same as the Sleep Clause but it refers to the freeze status effect. The Endless Battle Clause prevents players from intentionally preventing their opponent from winning without forfeiting. The Timer Clause causes a player to automatically lose the battle if their player timer is exhausted.

4. Experimental Design

To show how ML can be used as a tool for testing rules for Pokémon battles, we decided to designate one rule for experimentation. Due to the seemingly independent reasoning for each of Smogon's rules, it would make sense that rules are tested individually as one sees fit.

We decided the rule of greatest interest to us was the one prohibiting the use of OHKO moves. Not all Pokémon can use the moves, so we would expect this rule to restrict a small set of Pokémon.

In their best use cases, OHKO moves will work $\sim 30\%$ of the time, dealing a fixed amount of damage that can knock out any opponent. Smogon may have set this rule to prevent its users from being too strong, but it might alternatively exist to keep the game more interesting. For example, relying on luck to this extent may limit players' ability to win through better decision-making. As a counterargument, ice moves have a $\sim 10\%$ chance of permanently freezing the opponent, a large reason to use the move, yet there is no rule against it.

While it would be very interesting to know exactly how the utility of all Pokémon change by the removal of the rule and production of a new ranking of some sort, that would be either too difficult to compute or require many assumptions that may not be agreeable. We instead perform a single experiment where we hope to see some notable effects of the rule.

To perform our experiment, we use ML to control the actions of players in four repeated team battle scenarios. We configure two teams with three Pokémon each, determine control movesets for each that make sense for competitive play, and create alternate movesets that make use of OHKO moves. In the four scenarios, we test the likelihood of Team 1 winning when one, both, or neither of the teams use the alternate movesets. All other factors are assumed to be constant.

We conduct these battles using the most popular of Smogon's battle formats, OU, which only prohibits using Mew and Mewtwo. There are 14 Pokémon in the OU tier. Pokémon in lower tiers are allowed but not recommended in the format because they are seen as not worth using over the other 14.

By observing both the outcome of games and the specific learned behavior of the player agents, we should have insight on how competitive play may be affected by the removal of this rule set. If a team's change in moveset appears to increase its likelihood of winning, implying that players would want to use the moves, then we can be confident that the rule set impacts play. If not, then our results are not conclusive but suggest that the rule may not have much impact.

4.1. The Teams

Reasonably designing the two teams and their movesets is an important part of testing the rule, as we want the results of the experiments to have implications on how skilled players would optimally play. For example, if the introduction of OHKO moves to Team 1 yields large benefits against Team 2, this would be irrelevant if the control movesets of Team 1 were already ineffective, such that we could see the same benefits by using other currently legal moves. In another case, if Team 2 is not a good representation of a normal competitive team, then one's ability to beat it is not meaningful.

First, we acknowledge that 3v3 battles are uncommon for competitive games. Community rule sets, tier lists, and recommended movesets are all generally made under the assumption that battles are 6v6. Our primary reasons for this

change are to limit the number of variables affecting battle and improve the speed and performance of our applied ML. Not every Pokémon is capable of using OHKO moves in the games, so a reduced team size helps raise the concentration of OHKO move usage while allowing us focus on just a few of its users.

Table 1 details all relevant information about the two teams and their movesets. Following is a justification of this configuration.

Table 1
Configurations of Pokémon on both teams

	Team 1	**Team 2**
Slot 1	Slowbro	Snorlax
Slot 1 Moves	Amnesia Surf Thunder Wave Rest / **Fissure**	Body Slam Reflect Rest Self-Destruct / **Fissure**
Slot 2	Rhydon	Tauros
Slot 2 Moves	Earthquake RockSlide/**HornDrill** Body Slam Substitute	Body Slam Hyper Beam Blizzard Earthquake/**HornDrill**
Slot 3	Alakazam	Chansey
Slot 3 Moves	Psychic Seismic Toss Thunder Wave Recover	Thunderbolt Ice Beam Soft-Boiled Thunder Wave

In designing the teams, we noted that the three Pokémon, Snorlax, Tauros, and Chansey, can be see on almost every 6v6 team in OU. The Type or Types of a Pokémon usually cause them to have particular advantages or disadvantages against different Pokémon, but these three being of the Normal Type means that there are fewer moves that have type advantage against them. Similarly, their moves are less likely to be highly effective or ineffective against other types. It happens that these three also have some of the highest stat totals, making them some of the strongest Pokémon in isolation.

Another reason for their high usage is their synergy in performing different roles for a team. Snorlax is one of the bulkiest Pokémon in the game and has the damage to threaten any opponent. Tauros has both damage and speed, making it able to quickly finish off weakened opponents. Used effectively, it can force enemies to switch out, allowing for a free hit on the new Pokémon. Chansey specializes in inflicting negative status conditions and can recover health faster than a lot of Pokémon can deal damage. Snorlax and Chansey can also be seen using a variety of movesets depending on what best compliments the rest of the team.

For these reasons, we believed it would be interesting to see these three together as a complete team, where Snorlax and Tauros can both be given OHKO moves. The only other OHKO move users in OU are Rhydon and Slowbro, so we put them on the other team. Rhydon is advantageous for its ability to absorb Normal moves, while Slowbro can inflict paralysis with Thunder Wave. Paralysis is critical to the team because OHKO moves will not work if the user has less speed than the target. It also seemed interesting to make use of Slowbro, since he is one of the least used and lowest rated Pokémon categorized into OU.

Because these two are particularly slow, their team would benefit from having a fast Pokémon that can use also use Thunder Wave. There were several nominees for this role, but we selected Alakazam for its slightly better offense, which would help compensate for the absence of Tauros from that team.

4.2. The Moves

When determining the moves for the control teams, we wanted to balance using the most common moves with making all Pokémon reasonably useful for the battle. All three Pokémon on the Normal team are usually seen with Ice moves, which is redundant and counters Rhydon too well. Snorlax is the one most commonly seen without its Ice move, so we gave it Self-Destruct to account for other difficult situations.

A Pokémon can only know four moves, so a move would have to be replaced from each Pokémon that would be given an OHKO move. The fixed damage nature of these moves makes it seem reasonable to replace a damaging move that is usually only used situationally. For Snorlax, we replaced Self-Destruct. For Tauros, we replaced Earthquake, which is mostly just used against Gengar in OU.

We anticipated that Rhydon would have no reason to use Rock Slide or Body Slam when it has Earthquake, so we just replaced Rock Slide. Slowbro usually only has one damaging move, so ideally we would replace Amnesia or Rest. Each of these moves in the absence of the other is less useful, as they are effective in combination. In the end, we decided to sacrifice Rest because it leaves the user vulnerable and more predictable.

It may be important to note that neither team was given moves to inflict the sleep condition on enemies. Smogon has always recognized sleep to be one of the most powerful mechanics given to some Pokémon, which is why they have a rule preventing a team from making two of their opponents be asleep at the same time. Our primary reason for not using these moves was that the Pokémon we were considering selecting could not use them. We avoided intentionally giving the teams access to these moves because we feared that they simply could not be balanced; there are not enough Pokémon on each team for players to not be devastated by the reduction in options. We could also argue that both teams have enough ability to inflict other status conditions, helping fill the void made by sleep's exclusion.

4.3. Software Used

To simulate Pokémon battles, we used the pkmn engine, a Pokémon battle simulation engine optimized for performance in larger scale projects [15]. This open source tool accurately implements battles as done in the original game code and the popular simulator Pokémon Showdown. Pokémon Showdown is sponsored and endorsed by Smogon for competitive battles, as it provides practical implementations of battles for all Pokémon generations and a practical interface for playing online. The pkmn engine currently only fully supports the first generation of Pokémon, but it is able to run faster than Showdown while having less unneeded overhead.

We also used and credit another open source project, Wrapsire, which provides a C++ interface for the compiled library produced by the pkmn engine [16]. We opted to use C++ for this project because of its advantages for memory management, multi-threading, and compiler optimizations.

4.4. Monte Carlo Tree Search

Pokémon is a stochastic turn-based game, where a turn is initiated by both players selecting an action independently but concurrently. In order to observe battles that are consistent with the skill expected of competitive players, action selection must be mindful of what gives the highest chances of winning. To accomplish this, we decided to use Monte Carlo Tree Search (MCTS).

We chose to use MCTS primarily because of its ability to handle uncertainty and large state spaces. MCTS has also been shown to be effective at simulating Pokémon battles [13]. Pokémon uses a random seed as a factor in many different calculations, most notably in standard damage calculation and secondary move effects. Because of this, a turn initiated from a given game state by given actions may have hundreds of possible unique resulting game states depending on the seed. This is not problematic for MCTS because it naturally weighs the value of an action based on the frequency that different states result from it, and these states similarly develop better heuristics for move selection as the search continues.

The primary problem we faced with implementing MCTS for Pokémon battles was with determining how to handle concurrent action selection. Minimax trees are commonly used for turn-based games like chess, but these assume that players alternate turns and know what the enemy previously selected. This assumption contradicts the nature of competitive Pokémon battles, as it almost always involves players intentionally being unpredictable. This is due to a natural rock-paper-scissors-like relationship among strategies. For example, recovery may counter gradual damage, while applying buffs or statuses may counter recovery, and the right damaging move may yield an enemy's buffs ineffective or make them lose before they gain the longer-term benefits.

To address this issue, we decided to randomly determine at every simulation of a turn who will select their action first, and who will pick based on that selected action. While this does not perfectly represent the distribution of move combinations used in competitive games, it effectively balances the scenarios where a player either picks their safest action or correctly predicts that their opponent is picking their safest action. This gives an equal advantage to each player, while also producing more consistency in game outcomes, which is helpful for assuring the integrity of our results. It is also significantly less computationally expensive than developing stochastic policies, which makes it easily applicable at all frequently visited states during a search.

There is an additional concern caused by simulating forward in a game, and that is that it causes you to know everything about the opponent's team. Normally Pokémon is a partial information game, where one does not know anything about the opponent's team until they switch to each Pokémon and use their moves. If the battles in this experiment were done by people instead of ML agents, Team 2 not knowing that Alakazam could switch to Rhydon could end up disastrous for them if they told Snorlax to use Self-Destruct.

Representing partial observability not only makes the problem too complicated to get reasonable results but also may not change the results very much. Most Pokémon have a single set of moves that is used identically in over half of games. Additionally, teams in 6v6 battles tend to intentionally cover all of their weaknesses by having a Pokémon resistant to each commonly used move Type, basically following a recipe for team building. Essentially, players can expect their opponents to have some kind of counter until they have already defeated the counter, and it is not as important to know what exactly the counter is. All this leads to teams being rather predictable.

Other design notes had a lesser effect on the results. Primarily, since total health points and amounts of damage dealt are rather inflated, we bucketed states together when counting visits and related information, ignoring the bottom five bits of both Pokémon's health points when identifying a state during a search. This allows for much quicker convergence and is easily modifiable in our code.

4.5. Simulating Each Scenario

To observe the likelihood of Team 1 winning with reasonable precision, we decided to simulate 500 battles for each scenario, using MCTS at each turn. Each search consisted of 100,000 iterations, simulating from the active game state.

Because of the somewhat large depth and stochastic nature of Pokémon battles, we chose to limit the search depth to 25 turns from the active game state. Simulating past 25 turns of depth would likely be unnecessary since states would not get enough visits to do anything other than random rollouts, which only slowly progress to the end state and may favor certain movesets. After 25 turns, we terminated the simulated game and considered the team with the larger sum of percentage healths the winner. When this happened, we gave the winner a modified reward as if it was an average of 9 wins and 1 loss, which was to help encourage actual wins over this method. Implementing this caused it to take less time to complete each search.

We also encountered a problem with battles not ending when all Pokémon on a team were frozen. This was because the algorithm recognized that it could win on any future turn for equal reward. Rather than reducing the reward for winning after more turns, something that could theoretically negatively influence the decisions made, we decided that it was safe to assume that a team with all their Pokémon frozen will not win and let games be terminated at that point. None of these Pokémon had moves to deal damage on turns they do not attack, such as Toxic and Leech Seed.

When running the battles, we collected a detailed log of the actions selected at every turn and the values of relevant variables, such as health and status conditions. We used these to extract interesting information that may inform our discussion of how gameplay behavior changes when the external rules for team compositions change.

5. Results and Discussion

Table 2
Summary of battles in each scenario

Who Has OHKO Moves	Team 1 Win %	Avg. Turns	OHKO Sel/Bat	KO's /Bat
Neither Team	37.5%	52.754	0	0
Team 1	41.9%	42.448	2.666	0.588
Team 2	49.6%	49.426	1.15	0.100
Both Teams	60.6%	41.002	3.67	0.652

Observing the summary of the battles in Table 2, particularly the Team 1 Win % column, we can see notable

Table 3
Pokémon summary when **NEITHER** has OHKO moves

Pokémon	Move Given	Rate Used	KO's/Battle
T1-Slowbro	Rest	2.29%	N/A
T1-Rhydon	Rock Slide	9.10%	N/A
T2-Snorlax	Self-Destruct	12.36%	N/A
T2-Tauros	Earthquake	10.90%	N/A

Table 4
Pokémon summary when **Team 1** has OHKO moves

Pokémon	Move Given	Rate Used	KO's/Battle
T1-Slowbro	Fissure	21.61%	0.482
T1-Rhydon	Horn Drill	13.56%	0.106
T2-Snorlax	Self-Destruct	16.11%	N/A
T2-Tauros	Earthquake	10.34%	N/A

Table 5
Pokémon summary when **Team 2** has OHKO moves

Pokémon	Move Given	Rate Used	KO's/Battle
T1-Slowbro	Rest	1.86%	N/A
T1-Rhydon	Rock Slide	6.89%	N/A
T2-Snorlax	Fissure	4.00%	0.056
T2-Tauros	Horn Drill	15.22%	0.044

Table 6
Pokémon summary when **BOTH** have OHKO moves

Pokémon	Move Given	Rate Used	KO's/Battle
T1-Slowbro	Fissure	11.98%	0.424
T1-Rhydon	Horn Drill	8.43%	0.096
T2-Snorlax	Fissure	4.98%	0.100
T2-Tauros	Horn Drill	14.14%	0.032

changes in the outcome of games based on the change of movesets. In the control battles where neither team was given OHKO moves, we saw that the three popular Normal types, Team 2, won 5/8 of their games. We expected that this team would have an advantage, since they have high stats, few weaknesses, and seemingly good synergy.

When we tried replacing moves from the already advantaged Team 2 with OHKO moves, they lost some of their advantage. This suggests that the moves we replaced were more useful in this battle than the OHKO moves were. In fact, they collectively managed only 1 successful OHKO per 10 battles. Taking away Tauros's Earthquake likely was not a big deal since it still had Body Slam, but we suspect that the main reason for the change in performance is Self-Destruct, which knocks out the user to deal massive damage.

As a quick side note, Self-Destruct caused ties in 4 of the 2,000 battles by knocking out the final two Pokémon. Each of these was treated as half of a win when calculating Team 1's win ratio.

Snorlax's Self-Destruct can knock out Alakazam in one hit or take out 3/4 of Slowbro's health. 220 of the 482 times Snorlax fainted in the control scenario were from using this move. This indicates that the move was worth having and using against this team. On the other hand, Self-Destruct is more risky in a 6v6 battle because the opponent is more likely to be able to switch to a Pokémon like Gengar that can absorb the blow and leave Snorlax knocked out. Fissure did not help Snorlax here, but it might find more use with the right moveset and team.

When we tried putting OHKO moves into the movesets of just Team 1, we saw their win ratio go up from 37.5% to 41.9%. This suggests that we successfully found a use case for OHKO moves in competitive battles. Slowbro averaged about 1 successfully hit Fissure per 2 battles, while Rhydon hit about 1 Horn Drill per 10 battles. Rhydon's success rate was not very impressive, but this Pokémon's popularity within OU would probably be enough to cause Horn Drill to show up as an alternative to catch opponents off guard.

Slowbro had the most interesting results with OHKO moves. We replaced Rest, which Smogon quotes as being important for use with Amnesia. Slowbro probably would have used Amnesia and Rest a lot more if it had been starting against a special attacker instead of the physical attacker Snorlax. Against a different team with more special attackers, Slowbro might have done better with the recommended moveset. On the other hand, Slowbro's increase in ability to use alternate movesets would make it more useful in different scenarios, which could raise its usage rate. Opponents would not know whether to expect the standard Slowbro or one with Fissure, causing them to have trouble switching appropriately.

The scenario where both teams had OHKO moves is not as interesting because we already showed a disadvantage with the movesets for Team 2. However, it offers a little insight to how some things change with different circumstances. Most notably, the change in win rate was significantly greater when both changes were present together. Without going into too much detail, a strong hypothesis could be that the absence of Self-Destruct was more devastating for Team 2 when it had a Fissure-using Slowbro to deal with. Slowbro used over twice as many total moves in battles where Snorlax did not have Self-Destruct.

Aside from the four Pokémon already in OU that can use OHKO moves, this rule effects some Pokémon that are just below the cutoff for this tier. Dragonite may be the most notable of these. In OU, Dragonite is the holder of the highest base stat total but lacks the moves to back it up. Dragonite is relatively fast and can use Thunder Wave. Perhaps the only thing stopping it from becoming one of the strongest Pokémon with Horn Drill is its weakness to Ice. Ice moves are very useful and can be known by most Pokémon in OU.

Lapras is also a strong candidate for OU viability with Horn Drill, as it can put Pokémon to sleep with Sing, is very bulky, and outspeeds other bulky Pokémon. The rule limiting a team to putting just one opponent to sleep at once currently stops Lapras from being viable in OU, as usually having one sleeper is seen as ideal. With the addition of Horn Drill, Lapras might be a better replacement for another sleeper, especially with Blizzard to take down Rhydon pretty consistently. Smogon recommends using 3 damaging moves for Lapras, even though it is not very offensive; one of these is likely easily replaceable by Horn Drill.

Win rates aside, the battles can be seen changing in terms of the experience for players. Earlier we mentioned that the OHKO rule may exist to keep games interesting. One thing that can be noted from Table 2 is that, as the number of teams with OHKO moves increased, the turn counts decreased. When Team 1 had the moves and performed better, the games were about 4/5 as long, even though the win ratio reflected that it was a more even match.

Game length does not necessarily translate to more or less fun for the players, but this helps show how the game had more of a twist toward randomly rewarding a team with a large advantage that would cause it to end more decisively.

A 30-70 gamble with a high reward likely does this. It makes sense that skilled players would appreciate their win ratio having higher correlation to that skill they put a lot of time into mastering, so this seems like a strong potential motive for the rule.

We also show in Table 2 the average number of OHKO moves used by any Pokémon per battle in the column OHKO Sel/Bat. This helps us gain a sense of whether Pokémon are overutilizing OHKO moves with respect to other moves. If a Pokémon is disproportionately using OHKO moves, this could lead to the opposing player becoming annoyed or frustrated with the battle since OHKO moves generally involve luck to succeed. Three OHKO moves being selected in a battle lasting 41 turns is not likely to be seen as overusage or annoying. It may be important to acknowledge that this battle used all four Pokémon in the current OU tier that are able to use OHKO moves, so this can be seen as an estimated upper bound on OHKO move usage. Given this, the luck-based nature of these moves seems less concerning.

Under these almost ideal conditions for OHKO move usage, we observed this strategy not posing a notable problem to player experience. OHKO moves can be observed usually requiring setup by using Thunder Wave or Body Slam, which shows that the strategy still requires a degree of skill and team-building decisions to make effective. Furthermore, we can see that allowing these moves may give more benefit to less commonly used Pokémon, and we can anticipate that some Pokémon that are not considered strong enough for this battle format may become more relevant.

The results of this experiment do not show a strong justification for the rule banning the use of OHKO moves, but that is not to say that an alternate problematic case does not exist. Of course, it must be acknowledged that this is a limited case study designed to show how we can collect and analyze data to test the concerns associated with a rule. Should the rule be seriously put into question by the Smogon community, it would be wise to observe more cases than this and in a less anecdotal way.

6. Conclusion

Additional rule sets are often used in competitive games to create a healthy competitive metagame. This ensures that all participants are playing on a level field. These rule sets, however, are often designed using intuition and post-hoc usage statistics/analysis. In this paper, we explore how artificial intelligence techniques can be used to empirically evaluate competitive rule sets in the game Pokémon.

To do this, we use Monte-Carlo tree search to simulate 3v3 battles using a relaxed version of the Smogon generation 1 rule set, allowing Pokémon to use OHKO moves. Results show that OHKO moves can have a visible effect on how the battles play out, but that effect is nuanced in many ways. Our results indicate that while OHKO are used in battle when they're available, they likely are not used enough to cause annoyance. This, however, is not definitive, and more work must be done before any conclusions can be drawn. Overall, this approach gives designers critical context on how rules affect the potential metagame surrounding competitive play.

Overall, we feel that these preliminary results are promising and provide evidence that further work on utilizing automated playtesting techniques to evaluate competitive rulesets has merit. In addition, the approach we use in this paper has applications to games outside of Pokémon. Any competitive game that can be simulated could very easily make use of the approaches that we use in this paper. We hope this encourages further research into how artificial intelligence and machine learning can be applied in competitive games and metagames.

References

[1] S. Reis, R. Novais, L. P. Reis, N. Lau, An adversarial approach for automated pokémon team building and meta-game balance, IEEE Transactions on Games (2023).

[2] S. P. R. A. Reis, Artificial intelligence methods for automated difficulty and power balance in games (2024).

[3] D. Crane, Z. Holmes, T. T. Kosiara, M. Nickels, M. Spradling, Team counter-selection games, in: 2021 IEEE Conference on Games (CoG), IEEE, 2021, pp. 1–8.

[4] R. Ferdous, F. Kifetew, D. Prandi, I. Prasetya, S. Shirzadehhajimahmood, A. Susi, Search-based automated play testing of computer games: A model-based approach, in: International Symposium on Search Based Software Engineering, Springer, 2021, pp. 56–71.

[5] P. L. P. de Woillemont, R. Labory, V. Corruble, Automated play-testing through rl based human-like playstyles generation, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 18, 2022, pp. 146–154.

[6] S. Stahlke, A. Nova, P. Mirza-Babaei, Artificial players in the design process: Developing an automated testing tool for game level and world design, in: Proceedings of the Annual Symposium on Computer-Human Interaction in Play, 2020, pp. 267–280.

[7] S. Stahlke, A. Nova, P. Mirza-Babaei, Artificial playfulness: A tool for automated agent-based playtesting, in: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–6.

[8] C. Holmgård, M. C. Green, A. Liapis, J. Togelius, Automated playtesting with procedural personas through mcts with evolved heuristics, IEEE Transactions on Games 11 (2018) 352–362.

[9] B. Horn, J. Miller, G. Smith, S. Cooper, A monte carlo approach to skill-based automated playtesting, in: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, volume 14, 2018, pp. 166–172.

[10] D. Simoes, S. Reis, N. Lau, L. P. Reis, Competitive deep reinforcement learning over a Pokémon battling simulator, in: 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), IEEE, 2020, pp. 40–45.

[11] G. Rodriguez, E. Villanueva, J. Baldeón, Enhancing pokémon vgc player performance: Intelligent agents through deep reinforcement learning and neuroevolution, in: International Conference on Human-Computer Interaction, Springer, 2024, pp. 275–294.

[12] D. Huang, S. Lee, A self-play policy optimization approach to battling pokémon, in: 2019 IEEE conference on games (CoG), IEEE, 2019, pp. 1–4.

[13] H. Ihara, S. Imai, S. Oyama, M. Kurihara, Implementation and evaluation of information set monte carlo tree search for pokémon, in: 2018 IEEE international

conference on systems, man, and cybernetics (SMC), IEEE, 2018, pp. 2182–2187.
[14] J. Wang, Winning at Pokémon Random Battles Using Reinforcement Learning, Ph.D. thesis, Massachusetts Institute of Technology, 2024.
[15] K. Scheibelhut, libpkmn, 2021-24. URL: https://github.com/pkmn/engine.
[16] pasyg, wrapsire, 2023-24. URL: https://github.com/pasyg/wrapsire.