Modeling 1v1 Pokémon Battles with Zero-Sum Games and Expectimax Trees

Jake Smith
University of Wisconsin-Madison
jasmith55@wisc.edu

Nico Salm
University of Wisconsin-Madison
salm@cs.wisc.edu

Kevin Wang
University of Wisconsin-Madison
kevin.wang@wisc.edu

Abstract

This paper tackles modeling a specific metagame of Pokémon battling, 1v1 Pokémon battles, through the usage of zero-sum games, expectimax trees, and Nash equilibria. A 1v1 Pokémon battle consists of two phases: a choosing phase and a battling phase. Our approach is divided into three main steps: 1) deriving optimal move strategies for the battling phase while accounting for randomness (move accuracy, critical hits, etc.), 2) simulating Pokémon matchups to compute win rates based on these optimal strategies, and 3) using the resulting win rates to model the choosing phase and derive Nash equilibria. The first two steps were accomplished by simulating Pokémon battles in Python and applying reinforcement learning to uncover strategies where necessary. After utilizing this simulation for all possible matchups, we generated a 7x7 payoff matrix of Pokémon matchup win rates. The resulting payoff matrix could then be reduced to 3x3 subsets and solved for Nash equilibria to derive optimal strategies during the selection phase. Our results revealed that despite the inherent complexity of 1v1 Pokémon battles, clear optimal strategies could be derived for move choices, Pokémon matchups, and selection order.

1 Introduction

Pokémon battles are a core element of the franchise, with platforms like Pokémon Showdown fostering competitive play through custom battle simulators and metagames. Among these formats, 1v1 Pokémon battles provide a simplified, yet strategic, setting ideal for game theory analysis. With fewer variables than traditional battles, the 1v1 format is well-suited for utilizing traditional game theory techniques to derive optimal strategies in competitive play.

1.1 What is a 1v1 Pokémon Battle?

First, some context is needed on what a Pokémon battle actually is. Before battling, players must bring a set number of Pokémon, each varying based on moves, items, abilities, and stats. The win condition in a Pokémon battle is getting your opponent's HP to zero before they do, primarily accomplished by using damaging moves. In a battle, players choose from four available moves, each varying greatly in effect.

In the 1v1 format, players bring three Pokémon total and must first select which Pokémon to battle. This is known as the *choosing phase*.

Preprint. Under review.

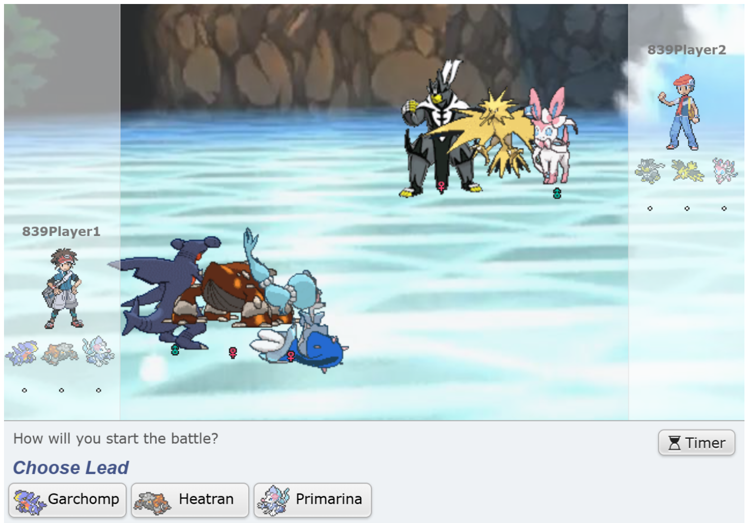

Figure 1: Choosing Phase

Once the Pokemon are chosen, they engage in the battling phase, where the goal is to defeat the opposing Pokemon by selecting the right moves.

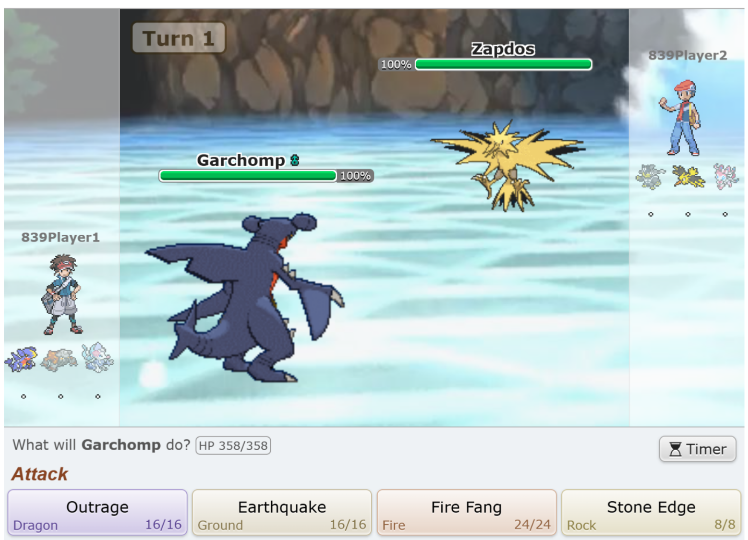

Figure 2: Battling Phase

An important detail to note is Pokemon battles have a lot of non-determinism through mechanics such as accuracy (moves may miss), critical hits (small chance to deal more damage), and damage rolls (same moves against the same Pokemon may not always deal the same exact amount of damage). This can lead to even identical matchups and move choices leading to completely different outcomes.

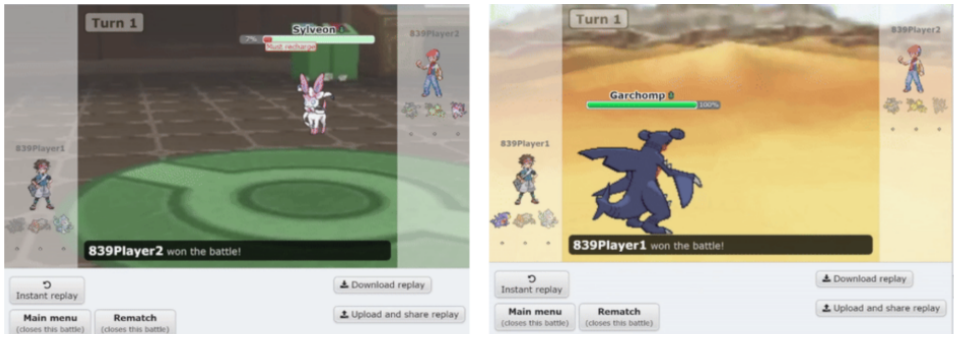

Figure 3: Despite both players using the same Pokémon and selecting the same moves, the left game results in one player winning, while the right game results in the other player winning.

1.2 Project Goals

Our main goals are to model the battling phase by deriving optimal strategies for each Pokémon and then use these strategies to calculate win rates. We then aim to apply these win rates to drive Nash equilibria for the choosing phase. In doing so, we aim to answer three main questions:

1. Which move choices are optimal throughout a battle?
2. How do Pokémon fare against one another when using optimal move choices?
3. Which Pokémon should be selected for the best chance of victory?

2 Problem Formulation

We model the 1v1 Pokémon battles as a zero-sum game between two players, each selecting a three-Pokémon team out of a pool of seven of the most used Pokémon from the January 2021 Pokémon Showdown 1v1 leaderboard. The objective of each player is to reduce their opponent's Pokémon's HP to zero before their own is defeated.

Figure 4: The 7 Pokémon we modeled

2.1 Key Assumptions

- **No Moveset Variations:** Each Pokémon uses the same moveset, ability, item, and stat spread. In our case, they are using sets from Smogon's Strategy Dex [3].
- **Optimal Strategy:** Both players are assumed to always use optimal strategies during the battle when calculating win rates.

For each Pokémon matchup, we calculate win rates assuming both players are using optimal strategies throughout the battle. These win rates are then used to model the choosing phase and identify Nash equilibria for Pokémon selection.

3 Method and Approach

3.1 Modeling Battles and Deriving Win Rates

To model battles, we construct a 4x4 payoff matrix for each turn, representing all possible move choices. For damaging moves, this corresponds to the percentage of HP dealt to the opponent. Setup and status moves impact future turn payoff matrices. These values were all calculated using Pokémon Showdown's damage calculator [1].

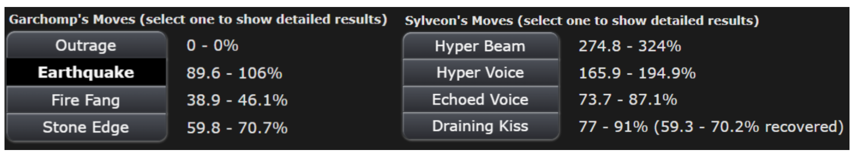

Figure 5: Example Turn Payoff Matrix

To account for randomness we construct a tree of possible game outcomes. Players will then choose moves that maximize their expected win rates across all possible scenarios.

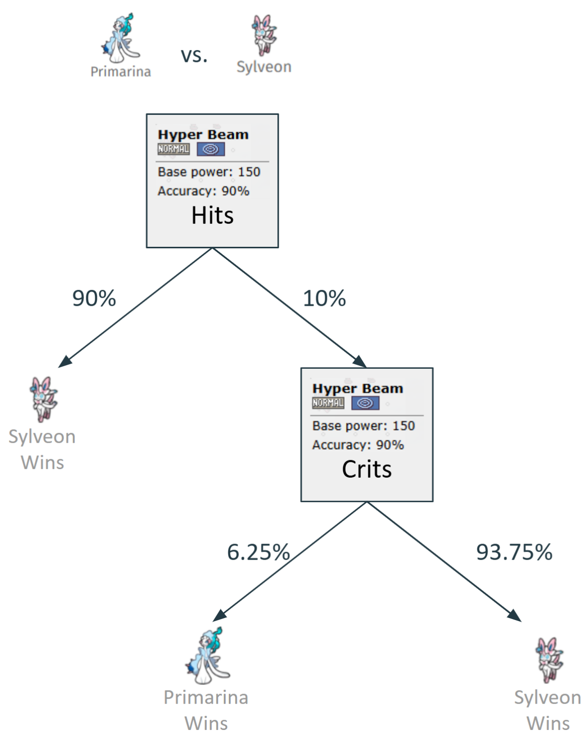

Figure 6: Example Game Tree

Additionally, we need to account for mind games inherent in certain moves like Sucker Punch. Sucker Punch is a move that usually goes first but only works if your opponent uses an attacking move, and this can create complex scenarios where the optimal play fully depends on what move your opponent chooses.

To account for all of these factors, we developed Python simulations with 1 million simulations to simulate randomness in battle and are used to derive optimal move choices. In cases where the optimal move choices are unclear, we deploy reinforcement learning to find the best strategies for each Pokemon. To determine win rates, we rerun the simulation with both Pokemon's optimal strategies.

3.1.1 Example Game 1: Heatran vs. Garchomp

In this example, we examine a battle between Heatran and Garchomp. Garchomp is using the item *Choice Band*, which locks it into the first move it uses in exchange for a massive attack boost. Heatran, on the other hand, benefits significantly from its item *Air Balloon*, which grants immunity to Garchomp's otherwise 4x super-effective Earthquake.

Garchomp's Move Choice: Since Garchomp's Earthquake is shut down by Heatran's Air Balloon and Fire Fang is neutralized by Heatran's ability, Garchomp can choose between using *Stone Edge* and *Outrage*. *Outrage* deals less damage to Heatran but has perfect accuracy, while *Stone Edge* does more damage but has an 80% chance to land, along with a higher critical hit ratio. After running the simulation, it is determined that Garchomp should use *Stone Edge*, as it has a slightly higher win rate than *Outrage*.

Heatran's Move Choice: For Heatran, it can choose between using *Will-O-Wisp* on turn 1, which burns Garchomp and halves its attack for the rest of the game, or continuously using *Dragon Pulse*. Heatran's other two attacking moves are not considered, as they deal less damage and offer no added advantages over *Dragon Pulse*. Another factor Heatran must consider is the 85% accuracy of *Will-O-Wisp*, and what to do if this move misses. The simulation concludes that Heatran's best chance of winning is using *Will-O-Wisp* on turn 1, followed by *Dragon Pulse* for the remainder of the battle.

After running this battle in the simulation with the derived optimal strategies, the following win rates are calculated:

- **Heatran:** 81.7%
- **Garchomp:** 18.3%

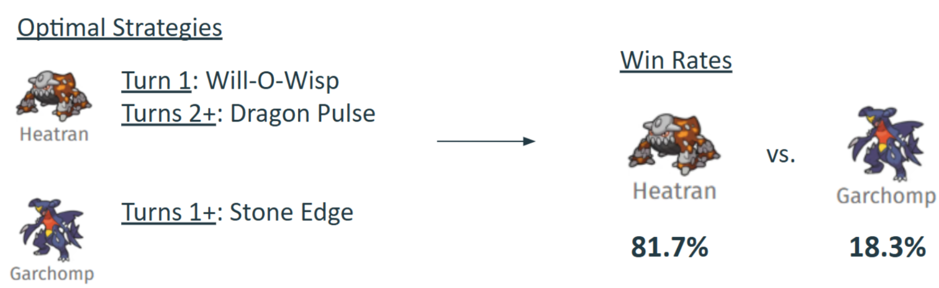

Figure 7: Battle setup and outcome for Heatran vs. Garchomp

3.1.2 Example Game 2: Rillaboom vs. Urshifu

One of the more complicated games we had to model was between *Rillaboom* and *Urshifu*, due to the move *Sucker Punch*. Since *Sucker Punch* only works when an attacking move is used, it creates a complex dynamic where each player's optimal strategy depends on predicting the other player's move. This leads to the following complicated payoff matrix between both players:

Additionally, the chance that *Sucker Punch* deals enough damage varies significantly based on both its current damage roll as well as the damage rolls of previous turns. This introduces a high level of variability, meaning that this matrix may not always hold true. This is where reinforcement learning

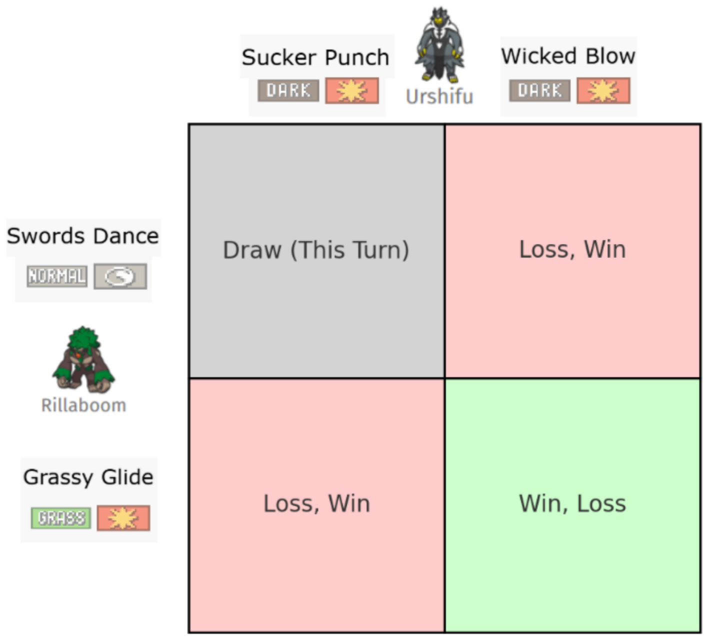

Figure 8: Payoff matrix for Rillaboom vs. Urshifu involving the move Sucker Punch.

was especially useful, as the mixed strategies involved made determining optimal strategies incredibly difficult without it.

After running the simulation, we found the optimal strategies for both Pokémon (see Figure 9), resulting in the following win rates:

- **Rillaboom**: 69.5%
- **Urshifu**: 30.5%

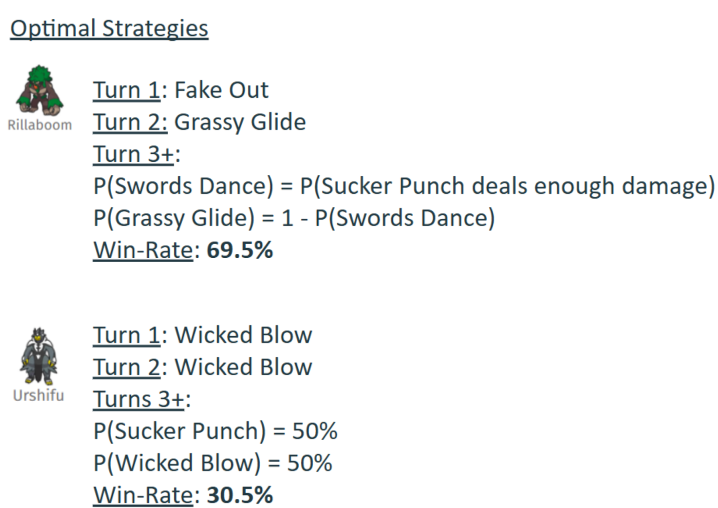

Figure 9: Battle setup and outcome for Rillaboom vs. Urshifu

3.1.3 Final Payoff Matrix

After repeating this process for every possible matchup, we ended up with the following payoff matrix:

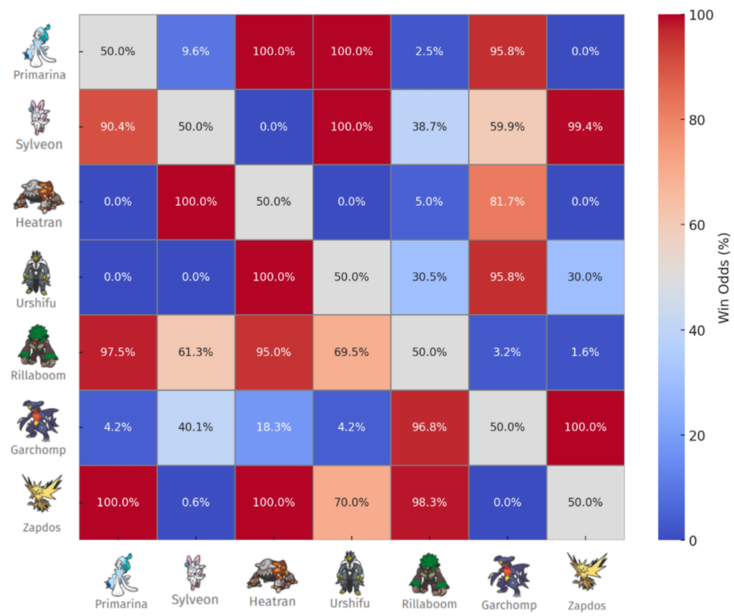

Figure 10: Payoff matrix showing the results of all possible matchups

3.2 Modeling the Choosing Pokemon Phase

To model the choosing Pokemon phase, we construct 3x3 subsets of our payoff matrix to represent battles, where players want to choose Pokemon that give them the best possible win rate.

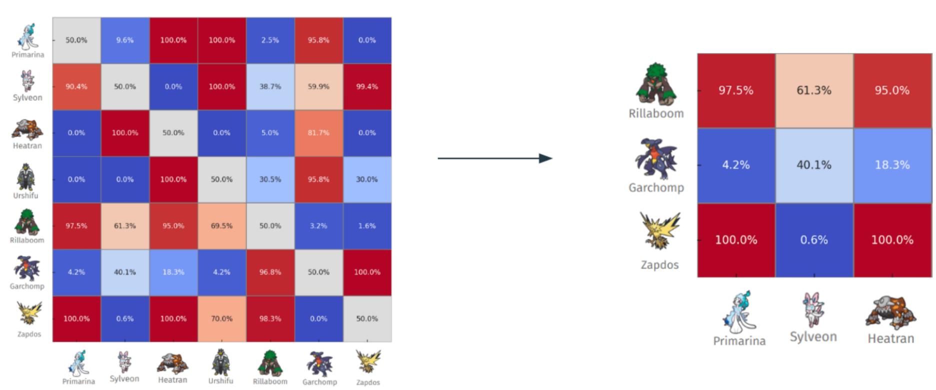

Figure 11: For example, the following subset represents a matchup between Rillaboom, Garchomp, and Zapdos on one team and Primarina, Sylveon, and Heatran on the other

After we construct one of these subset matrices, we then compute the Nash equilibrium to derive each player's optimal strategy and their win rate. Given the nature of these matrices, there were some games with Pure Nash equilibria and very dominant strategies, while others may resemble games like Rock Paper Scissors.

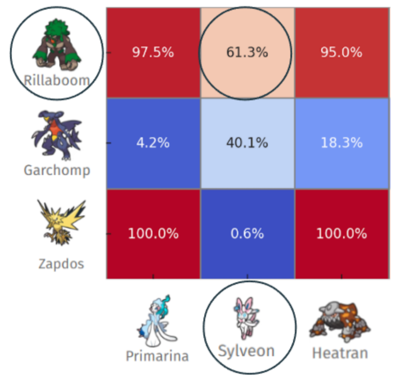

Figure 12: In this game, the row player has a dominant strategy of Rillaboom and the column player has a dominant strategy of Sylveon, leading to a Pure Nash equilibria. The win rate can be derived by this equilibria, which is 61.3% in favor of the row player.

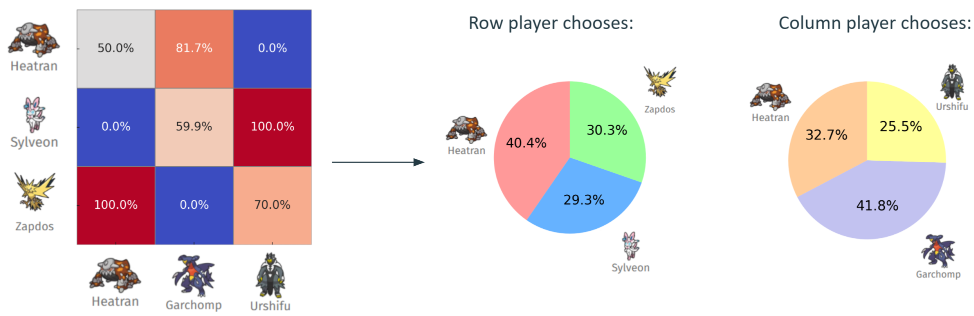

Figure 13: In contrast, this game shows no clear dominant strategy for either player, leading both to employ complex mixed strategies. This results in a game closer to Rock, Paper, Scissors than a traditional "solved game," with the win rate for the row player at 50.5%.

4 Results

After computing the Nash equilibrium for every possible matchup, we discovered the best and worst performing teams for our subset of Pokémon.

Rank	Team	Avg Win Rate
1	Heatran, Sylveon, Zapdos	62.4%
2	Primarina, Sylveon, Zapdos	59.3%
3	Rillaboom, Sylveon, Zapdos	58.6%
4	Sylveon, Urshifu, Zapdos	58.5%
5	Garchomp, Rillaboom, Sylveon	56.1%
6	Garchomp, Sylveon, Zapdos	55.7%
7	Heatran, Rillaboom, Sylveon	55.6%
8	Primarina, Rillaboom, Sylveon	55.6%
9	Garchomp, Primarina, Rillaboom	55.6%
10	Garchomp, Primarina, Sylveon	55.4%
11	Rillaboom, Sylveon, Urshifu	54.9%
12	Garchomp, Heatran, Zapdos	53.8%
13	Garchomp, Sylveon, Urshifu	53.7%
14	Garchomp, Primarina, Zapdos	52.5%
15	Garchomp, Rillaboom, Urshifu	52.5%
16	Garchomp, Heatran, Rillaboom	52.1%
17	Heatran, Primarina, Sylveon	52.1%
18	Heatran, Primarina, Zapdos	51.9%
19	Garchomp, Heatran, Sylveon	51.5%
20	Heatran, Sylveon, Urshifu	51.2%
21	Garchomp, Rillaboom, Zapdos	51.0%
22	Primarina, Rillaboom, Zapdos	50.9%
23	Heatran, Urshifu, Zapdos	49.6%
24	Rillaboom, Urshifu, Zapdos	49.5%
25	Garchomp, Urshifu, Zapdos	49.0%
26	Primarina, Sylveon, Urshifu	48.9%
27	Heatran, Rillaboom, Zapdos	48.7%
28	Garchomp, Heatran, Primarina	47.8%
29	Garchomp, Primarina, Urshifu	41.9%
30	Primarina, Rillaboom, Urshifu	41.3%
31	Heatran, Rillaboom, Urshifu	41.1%
32	Primarina, Urshifu, Zapdos	37.6%
33	Heatran, Primarina, Rillaboom	32.4%
34	Heatran, Primarina, Urshifu	31.7%
35	Garchomp, Heatran, Urshifu	29.7%

Figure 14: Best and worst performing teams based on Nash equilibrium calculations

As can be seen, the team that performed best was Heatran, Sylveon, and Zapdos which had an average of 62.4% chance to win against all teams.

On the contrary, the team with the worst performance was Heatran, Garchomp, and Urshifu which had an average of 29.4% chance to win against all teams.

Additionally, we were able to determine the average win rate for teams containing each Pokemon, which are the following:

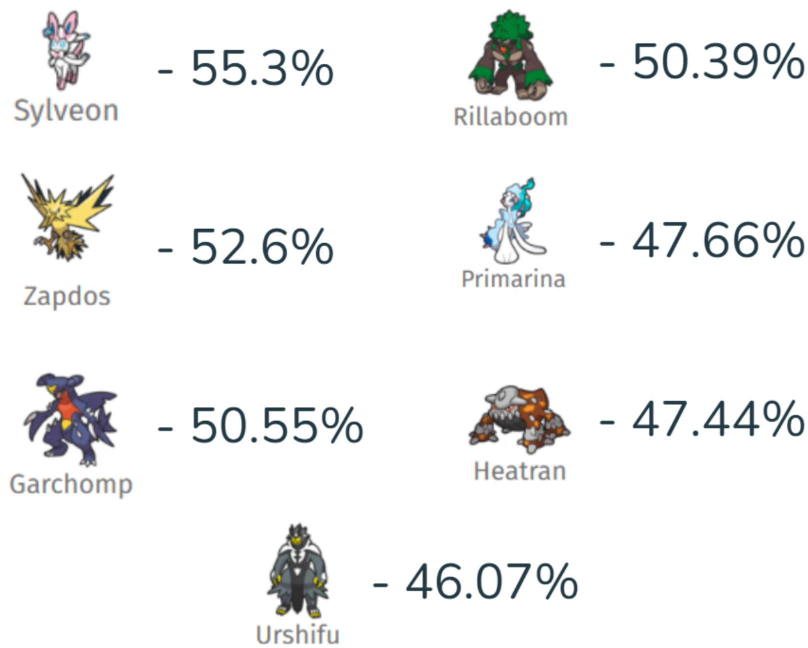

Figure 15: Best and worst performing Pokémon based on Nash equilibrium calculations

5 Challenges

One of the main challenges we encountered was managing the extensive randomness in Pokemon battles, particularly due to damage rolls. Each move has 16 possible damage rolls, and with battles lasting up to 10 turns, the sheer volume of randomness made it impossible to rely solely on expectimax trees for win rate calculations. To address this, we implemented a high-iteration Python simulation to properly account for all the variability.

Another challenge was dominant Pokemon skewing the balance of the choosing phase. Initially, when we were modeling 5 Pokemon, Rillaboom ended up being consistently the best choice for every player that had it with its worst matchup being a 50% chance to win against itself. This led to boring games with repetitive outcomes, so we added Zapdos and Garchomp to make games more even and interesting to model.

6 Conclusion and Future Work

Overall, we are pleased with our ability to successfully model optimal move choices, win rates, and optimal Pokemon selection in 1v1 Pokemon battles. Our analysis provided valuable insights into which teams and Pokémon perform best in these settings while properly accounting for the frequent randomness and mind games that come with every Pokemon battle.

One limitation of our approach was the assumption that all Pokemon have the exact same movies. Since there are tons of possible moveset variations that can change Pokemon matchups drastically, we wanted to keep it simple and look at a few variations. In reality, this is one of the more important aspects of 1v1 Pokemon battles with Pokemon like Iron Valiant, a Pokemon known for having huge moveset variability, currently being the #1 most used Pokemon in 1v1 [2]. Additionally, we'd like to expand the amount of Pokemon we look at to more than 7, and potentially do a full payoff matrix for 50 of the most used Pokemon in the current 1v1 ladder.

In the future, we'd also like to try modeling traditional Pokemon battles with switching included. Instead of choosing 4 moves every turn, we would need to model 4 moves and 5 possible switches (in a 6v6 game). This would be extraordinarily complex as some battles can go up to 1000 turns, but would be very interesting to try to solve regardless.

References

[1] Pokémon Showdown. *Pokémon Showdown Damage Calculator*. Accessed: 2025-04-30. 2025. URL: `https://calc.pokemonshowdown.com/`.

[2] Smogon. *Pokémon Showdown 1v1 Ladder Stats*. Accessed: 2025-04-30. 2025. URL: `https://www.smogon.com/stats/`.

[3] Smogon. *Smogon Strategy Dex*. Accessed: 2025-04-30. 2025. URL: `https://www.smogon.com/dex/ss/pokemon/`.