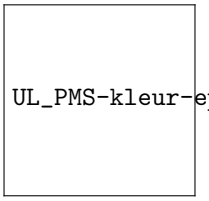


# Master Computer Science

Testing Game Experience Using Deep Reinforcement Learning and Curriculum Learning


| | |
|---|---|
| Name: | Philip Edelaar |
| Student ID: | 3373592 |
| Date: | 20/08/2025 |
| Specialisation: | Computer Science: Data Science |
| 1st supervisor: | Mike Preuss |
| 2nd supervisor: | Giulio Barbero |


Master's Thesis in Computer Science


Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

**Abstract**

This research utilizes two reinforcement learning agents, one of which will employ a curriculum learning strategy. The agents will be used to analyze the complex game experience of Pokémon Gold, by simplifying the game to make it more feasible for the algorithms. It will test whether assumptions for the simplification still make the game representable and show that for most of the game the game can be analyzed using these assumptions.

# Contents

# 1 Introduction

Nowadays, a lot of research in the field of Artificial Intelligence (AI) for video games focuses on optimization (e.g., how to beat the games quickly, efficiently and without damage). Reinforcement learning (RL) is a learning paradigm that is often used in research on this topic. This is also true for Pokémon games, especially in a player versus player (PvP) setting. However, single player modes represent a different challenge, with more complex rules. The goal of this research is to apply RL algorithms to Pokémon Gold, a single player game, in order to analyze game play and explore whether applying simplifications to the game still make it a viable testbed for AI.

# 2 Related Work

Our research involved the exploration of several AI algorithms and research on the use of AI in Pokémon. This chapter provides an overview of these tools in the context of game AI research.

## 2.1 Reinforcement Learning

RL is often used when automating game play. A standard RL algorithm comprises an agent interacting with an environment. This interaction can be denoted as a Markov Decision Process (MDP), defined: $(S, A, P, R, \gamma)$

- $S$ the state space, is defined as the set of all possible states
- $A$ the action space, is defined as the set of all possible actions
- $P(s_{t+1}|s_t, a_t)$ is defined as the transition probability function for state $s_t \in S$ and $a_t \in A$ on timestep $t$
- $R_{t+1}(s_t, a_t)$ is the reward function for the agent performing action $a_t \in A$ in state $s_t \in S$ on timestep $t$
- $\gamma \in [0, 1]$ is defined as the discount factor

Classic reinforcement learning has been around for some time. The book by Sutton and Barto [15] captures a lot of the variants and algorithms in the field. In 1996 there was a survey [6] with the use cases of RL and states that the AI community have gathered an interest in RL. This will later be refined as deep reinforcement learning.

### 2.1.1 Deep Reinforcement Learning

Deep reinforcement learning (DRL) is a more recent approach, combining RL and deep learning. it uses deep neural networks to derive a policy from the state input. This allows an algorithm to have a much higher dimensional space input making the options for RL much greater. One of the major breakthroughs was when researchers started creating agents that could play Atari games [7]. By using the raw pixels as input and creating a variant of Q-learning the agent could learn to play games like Pong and Space Invaders. When these games could be learned via DRL, chess and Go quickly followed as the next challenge. AlphaGo was created to beat Go [13] achieving a $99.8\%$ winning rate against other programs.

### 2.1.2 Proximal Policy Optimization

PPO[11] is a more recent DRL method, which utilizes a clipping mechanism. Whereas other policy gradient methods may suffer from destabilizing policy updates, this clipping mechanism limits the difference in policy updates making this algorithm more robust. PPO has a wide range of uses, one of which is in video games. An example of this is a multi-agent case which uses Starcraft [19].

### 2.1.3 Curriculum Learning

Curriculum learning is a way of learning used in multiple AI fields to tackle hard problems. The idea is that the agent first trains on a simplified version of the problem, which can be in all parts of the MDP, and then as iterations go on increase the complexity until the agent can solve the complex problem. An example which also includes PPO is a research that uses CL for autonomous driving [16].

## 2.2 Video Games and Artificial Intelligence

A lot of research uses AI to play and analyze video games. Our research tries to use AI on part of a game which is what the following research also focuses on. This research focuses on navigation in 3D video games [1].

### 2.2.1 Mimicking Human Behavior with AI

Mimicking human behavior using AI is not something new. A famous example of OpenAI uses RL to let agents play hide and seek [2]. The agents learn to use objects from scratch and explore game mechanics while they play as a human player would.

### 2.2.2 Pokémon and AI

The single player Pokémon games have been subject to research in the field of AI for some time. Research has been done on Pokémon Red in which the agent plays the full game until the first gym [9]. This research, in contradiction to our research applies an agent to all aspects of the game, but only plays until the first gym.

# 3 Research Question

This research will incorporate knowledge from section 2 and we will pose three subquestions which will be part of the main research question.

## 3.1 Assumptions for Simplification

Complex games usually have large state spaces, large action spaces and a multitude of game play mechanics that are closely interconnected. Even though the DRL algorithms of today work really well, this can be a challenge to model. It is for this reason that this research delves deep into how a video game can be stripped of parts of the the game play while still being representable of its core game play:
**RQ1: How to simplify complex games while maintaining their core game play?**

## 3.2 Mimicking Human Game play

Reinforcement learning already has a human side in the fact that it learns from experience to get a better reward. Couple this with curriculum learning which uses knowledge from previous problems to solve new problems and imagining that a curriculum learning agent can mimic human game play is not far fetched.
**RQ2: How can curriculum learning mimic player experience?**

## 3.3 Reinforcement Learning and Game Difficulty

The use of curriculum learning can on one hand be used to mimic human behavior, yet on the other hand it can tell us something about the game's difficulty at a certain stage. Therefore **RQ3: Can reinforcement learning and curriculum learning tell us about a game's learning curve?**

## 3.4 Main Research Question

All these research questions are part of how we can analyze a player's game experience. The combination of learning about the game and learning about how a player will play and experience it, will give us a unique insight in how RL and CL can help us to analyze this.
**How to use deep reinforcement learning and curriculum learning to analyze complex game experience?**

# 4 Defining the Game Space: Pokémon Gold

Pokémon Gold is part of the second generation Pokémon games which was introduced for the Gameboy Color in the year 2001 in Europe. Its publisher Game Freak had already published the first generation Pokémon games a few years before. The premise of the game is to capture and train animal-like creatures called Pokémon and let them battle other Pokémon.

## 4.1 Game play

The core game play of standard Pokémon games can be divided into four components.

- Exploration
- Battling
- Training
- Team building

These four components are all connected to one another. This research will focus on the team building aspect of the game. All the parts are in some way interconnected, so the other three parts will be considered a constant. They will be treated as an automated mechanic in order to experiment with the reinforcement learning agents as team builders.

### 4.1.1 Team Building

Building a team in Pokémon Gold is relatively simple. There are six available spots for six Pokémon. These Pokémon all have a level and with that level certain 'stats' ("The word "statistic" (or "statistics") is not used in any core series game with this meaning") [14]:

- Health Points (HP)
- Attack
- Defense
- Speed
- Special Attack
- Special Defense

Pokémon also have a maximum of two types. These types can multiply damage based on the chart in figure 1. The effectiveness of the types is usually done logically, for example: Water is effective against fire. The steel and dark type are two new additions that were not around in the first Pokémon game.


Figure 1: Type Chart in Pokémon Gold

### 4.1.2 Gym Leaders

Another important aspect of the game are the gym leaders. These gym leaders are the players' opponent and serve as a test of skill which the player has to beat to get access to new parts of the game. There are eight gym leaders in the base part of the game, and after defeating these and finishing the game the player gets access to another 8 gym leaders. For this research, the algorithm will try to beat the initial eight gym leaders, because beating these will show the learning curve of the player best.
Each gym leader has Pokémon of a specific type.

1. Falkner, flying
   - Pidgey
   - Pidgeotto
2. Bugsy, bug
   - Metapod
   - Kakuna
   - Scyther
3. Whitney, normal
   - Clefairy
   - Miltank
4. Morty, ghost
   - Gastly
   - Haunter
   - Haunter
   - Gengar
5. Chuck, fighting
   - Primeape
   - Poliwrath
6. Jasmine, fighting
   - Magnemite
   - Magnemite
   - Steelix
7. Pryce, fighting
   - Seel
   - Dewgong
   - Piloswine


Figure 2: Battle in Pokémon Gold

8. Clair, fighting
    - Dragonair
    - Dragonair
    - Dragonair
    - Kingdra

# 5 Methodology

This section will use the information from section 2 and describe how these methods apply to this specific research.

## 5.1 Reinforcement Learning

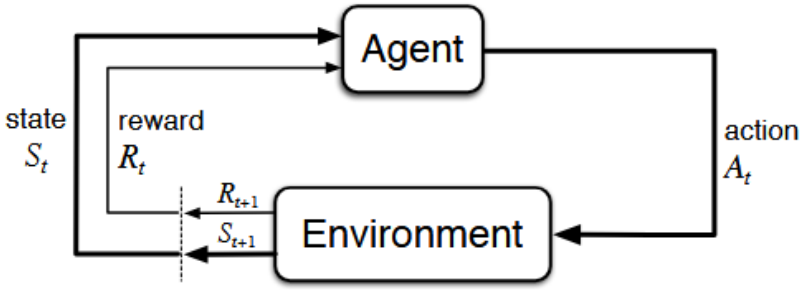


Figure 3: The Agent Interacting with its Environment [15]

Figure 3 shows the relation of each element of the tuple.
In this case, one episode consists of six discrete timesteps, one for each choice of Pokémon to fill the team (section 4.1.1). Hence $t \in \{0, 1, 2, 3, 4, 5\}$. When timestep $t = 5$ is reached, the algorithm will return a reward and the episode will terminate. Due to the fixed length of the episodes and the amount of timesteps having no influence on the reward, setting $\gamma = 1$ is justified.

In RL, the agent uses a policy $\pi(a_t|s_t)$ to determine which action to take on timestep t. These policies will have both an exploration factor, exploring state space $S$, and an exploitation factor, finding improved solutions in areas of high reward.
The following sections will describe how the MDP is defined for this research.

### 5.1.1 Agent

The agent interacts with the environment and chooses the actions based on a policy. In this case, the agent is the player. At each step of each episode it decides which of the available Pokémon it includes in the team based on the current state and the environment.

### 5.1.2 Environment

The environment dictates the decision probability function. In this case the agent always transition to a state in which it has the previously chosen Pokémon in a team. The environment must be created in such a way that it still represents the game, while also making it simpler in order for the algorithm have a better chance at converging. Therefore, assumptions have to be made that are both practical, but also represent the game well enough. The following assumptions will be tested by comparing the results to data of existing players involving Pokémon selection. The degree the agents deviate from this data will inform us about the validity of our simplified environment.

**Leveling and Evolutions** Leveling is part of the training component of the game (section 4.1) and is standardized by setting all Pokémon to the same level as the level of this gym leader's lowest leveled Pokémon. This represents how a player would level their Pokémon in actual game play while also forcing the algorithm to find good solutions. If the level were higher, the algorithm might get stuck in suboptimal solutions. Additionally, once a Pokémon reaches a certain level it has the option to evolve into a stronger version of itself. The player has an option to decline this evolution which in very niche cases has a positive effect. For simplicity reasons we decided that all Pokémon level the first time they get the chance.

**Moves** Each Pokémon has their own learnset, a list of levels in which it will be able to learn certain moves. We decided that each Pokémon has access to its four most recently learned move. This mimics the players curiosity of wanting to try new moves they just unlocked.
Additionally, there are hidden machines (HM) and technical machines (TM). These are obtainable items which can be given to the player's Pokémon to teach them a move.

**Battle Simulations** Many algorithms, including RL algorithms, for optimizing battle strategies in Pokémon exist [17] (see section 1). When a battle is only used for getting a reward we want the duration to be as quick as possible to ensure the algorithm can run many steps. Therefore simple heuristics were chosen:

- If a move has no PP left, do not use that move
- If the enemy has a status effect, do not use a status afflicting move
- Choose a move that is most effective due to type effect (section 4.1)

### 5.1.3 State Space

Each state $s_t \in S$ is represented by a tuple consisting of:

- Current team
- Gym leader team
- Counter for the step $t$

For each Pokémon in the current team and in the gym leader's team the observation is a list of all the base stats of the Pokémon followed by the stats for the move they use. For the type of the Pokémon and the move multi-hot encoding and one-hot encoding is used respectively.
The initial state is always the state in which the agent has zero Pokémon in their team. The representation of the absence of Pokémon or moves is an array of 0's

### 5.1.4 Action Space

The action $a_t \in A$, where $A$ is a discrete action space, is picking one of the available Pokémon.
In many RL implementations for video games the size of the action space is small (input buttons), however in this research the action space is much larger: picking one of all the Pokémon. Therefore, we have implemented action masking

**Action Masking** Action masking [18] is used to disable certain actions for the agent. For the first gym only a subset of all the Pokémon in the game are available, since the player can only explore part of the open world. This makes actions masking very natural for our research given that for each gym there are more Pokémon available than for the last.

#### 5.1.5 Reward Function

There are multiple ways to define how well a team performs. If the player is only interested in winning or losing, then $r = 1$ for winning and $r = 0$ for losing. This does not discriminate between comfortably winning and barely surviving. Therefore, we decided that $r = -1$ when the agent loses and the ratio of the health points: $r = \frac{\sum \text{HP left}_i}{\sum \text{total HP}_i}$ for $i \in \{\text{team}\}$. Subsequently: $r \in \{-1\} \bigcup (0, 1]$. Making winning a priority because of the low reward and winning comfortably an exploitation factor.

### 5.2 Deep Reinforcement Learning

Using DRL instead of just RL is relevant in our case, because of the complexity of Pokémon Gold. It can handle the increasing action space size much better than traditional RL.

#### 5.2.1 Proximal Policy Optimization

PPO is regarded as a robust and stable DRL algorithm. PPO can deal with large action spaces well, which is helpful for this research. Additionally, PPO balances exploration and exploitation which is useful for mimicking human game play.

### 5.3 Curriculum Learning

Section 5.1.4 indicates the use of action masking in the action space. The expansion of the action space when fighting later gyms is how Pokémon Gold is designed. Curriculum learning [3] was applied here to match game play.
By increasing the size of the actions space: $|A|$, the agent has a more complex problem to solve since there are more options to put in the team. Therefore, the implementation of CL is used in multiple ways: One is to give the agent an easier time solving the last gym by using knowledge obtained in the previous ones and two mimic player behavior of relying on what worked in the past to use in the future.

## 6 Experiment Design

The experiments will consist of two PPO agents: one baseline [8] agent which will learn each gym from scratch and one curriculum learning agent that will train on the model it used for the previous gym. The runs will be $100000$ steps for each gym for each agent. The PPO algorithm is implemented via the Stable Baselines 3 package [10] in Python. The standard hyperparameter values have been used for both agents. During the run the algorithm displays the mean reward after one rollout of $2048$ steps. The results will be averaged over 10 runs [8] and after each $10000$ steps the algorithm will display the 5 Pokémon that have been used the most.

## 7 Results

The results, obtained as described in section 6. Each gym has a figure of the learning curves or the two agents and two separate figures for the Pokémon that it often chose.

## 7.1 Gym 1

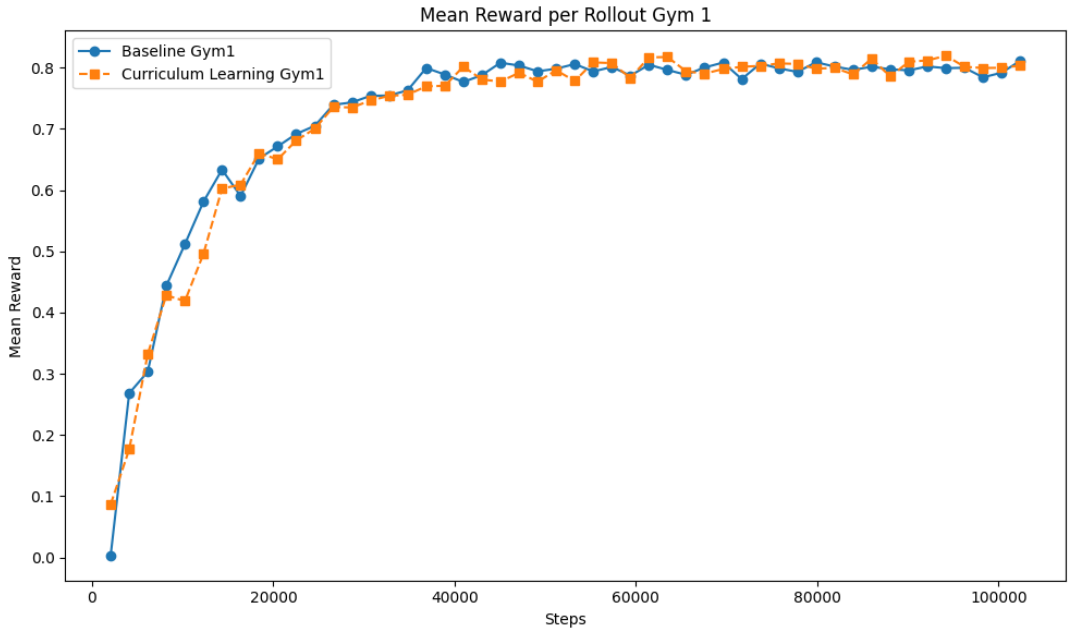


Figure 4: Learning Curve of Gym 1

In figure 4 the learning curves of both agents have a similar trajectory. This makes sense, because this is the curriculum agent's first iteration and it has therefore the same initial knowledge as the baseline agent.

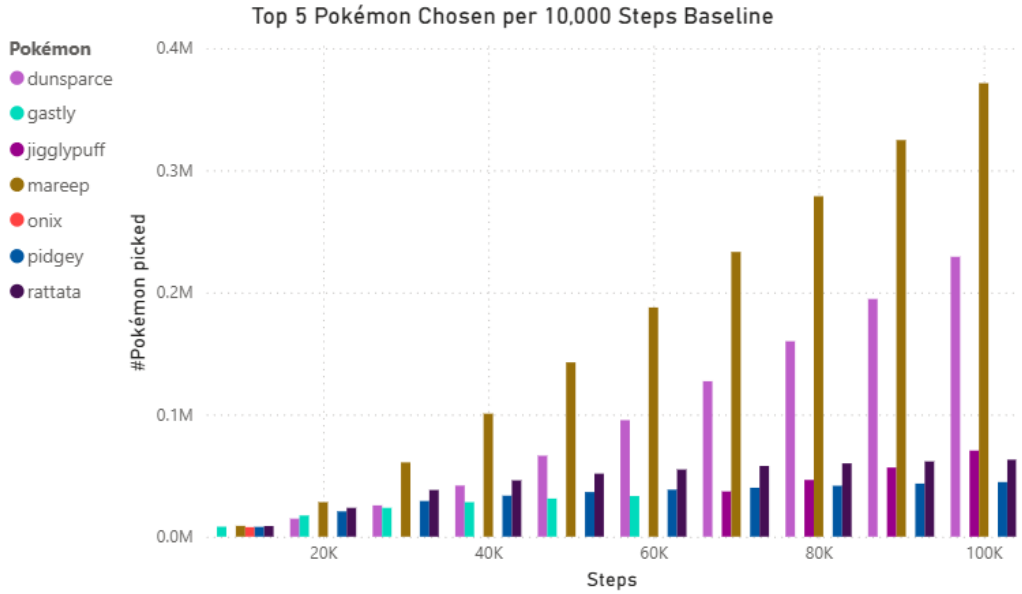


Figure 5: Most Chosen Pokémon Gym 1 Baseline Agent

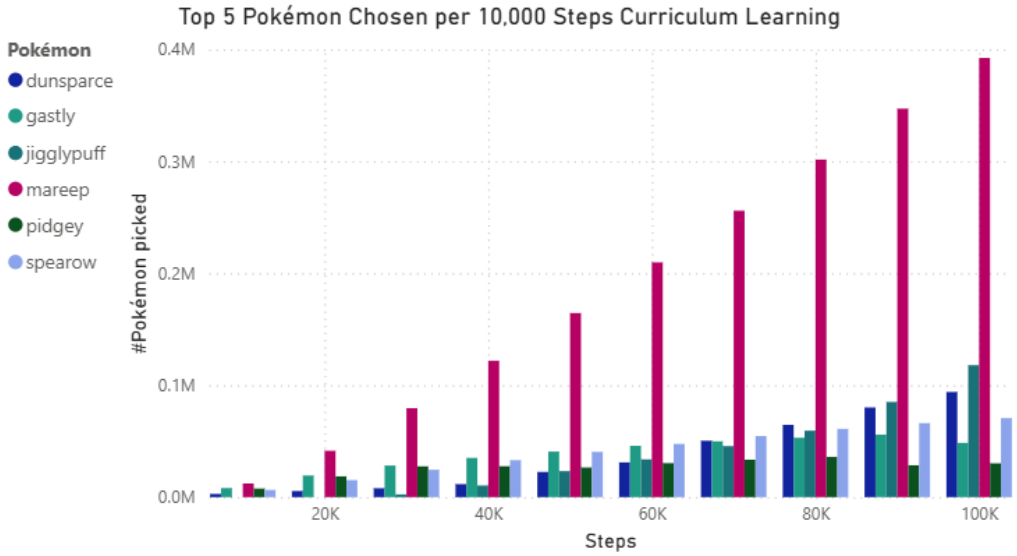


Figure 6: Most Chosen Pokémon Gym 1 Curriculum Agent

The same is true when looking at the most picked Pokémon by the agents. Figure 5 and figure 6 both show the algorithms opting for the Mareep.

## 7.2 Gym 2

Figure 7 shows that the curriculum agent has a higher reward at the start. This suggests that the prior knowledge obtained from gym 1 was useful for decision making in gym 2. However, to reach the optimal team formation, the curriculum agent takes a until 70000 steps before converging. Alternatively, the baseline agent starts with a worse mean reward, but finds the optimal team composition in 40000 steps.

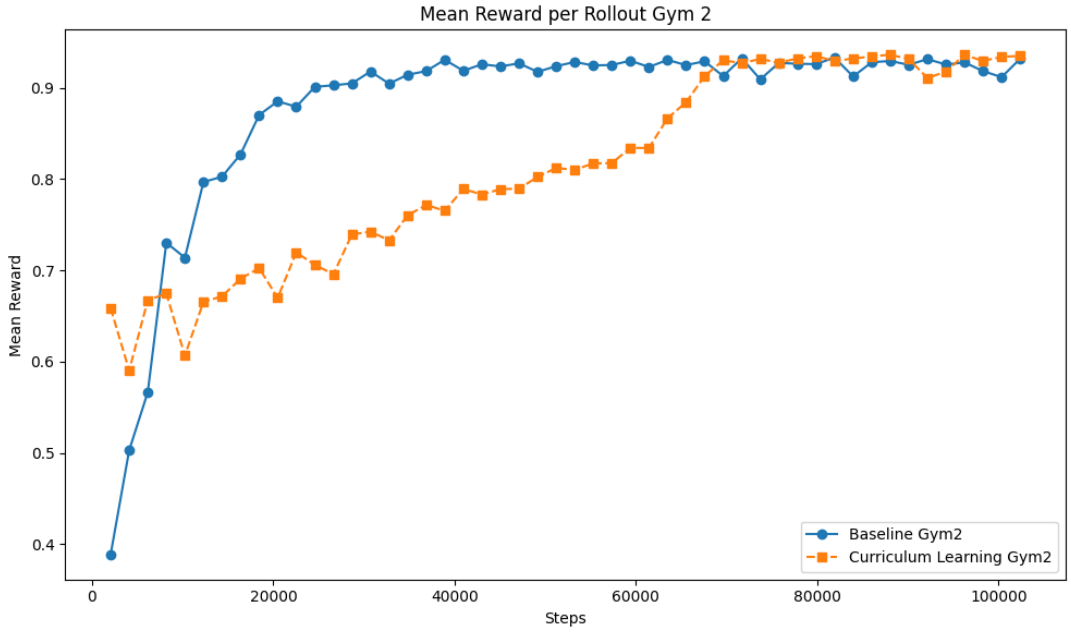


Figure 7: Learning Curve of Gym 2

Looking at the chosen Pokémon describes the phenomenon. The baseline agent in figure 8 shows a preference for both the Jigglypuff and the Onix. On the other hand, the curriculum agent in figure 9 shows that the agent opts for Jigglypuff and Mareep found in section 7.2 again showing bias for decisions that have worked well previously.

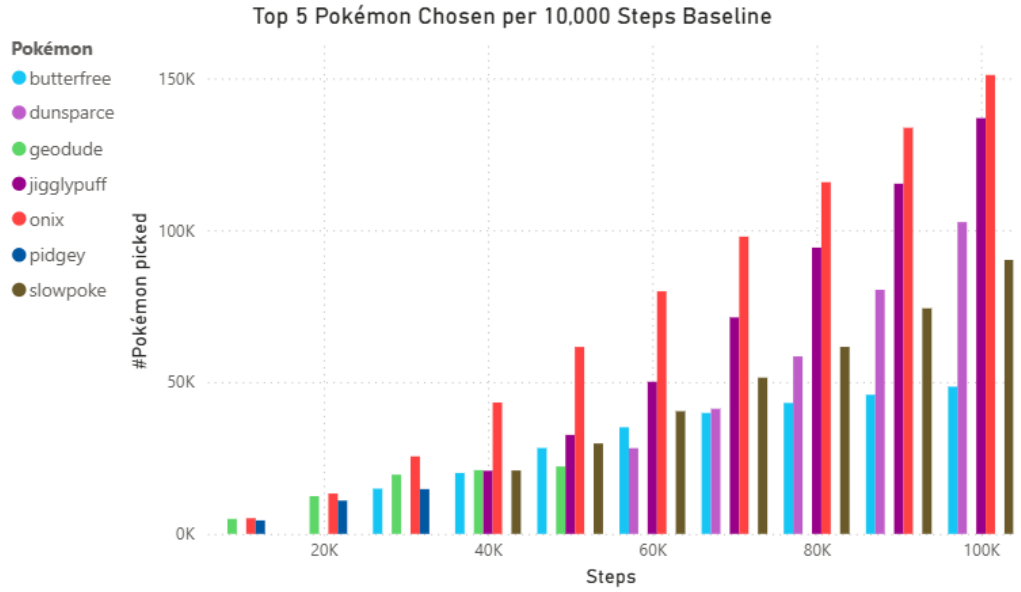


Figure 8: Most Chosen Pokémon Gym 2 Baseline Agent

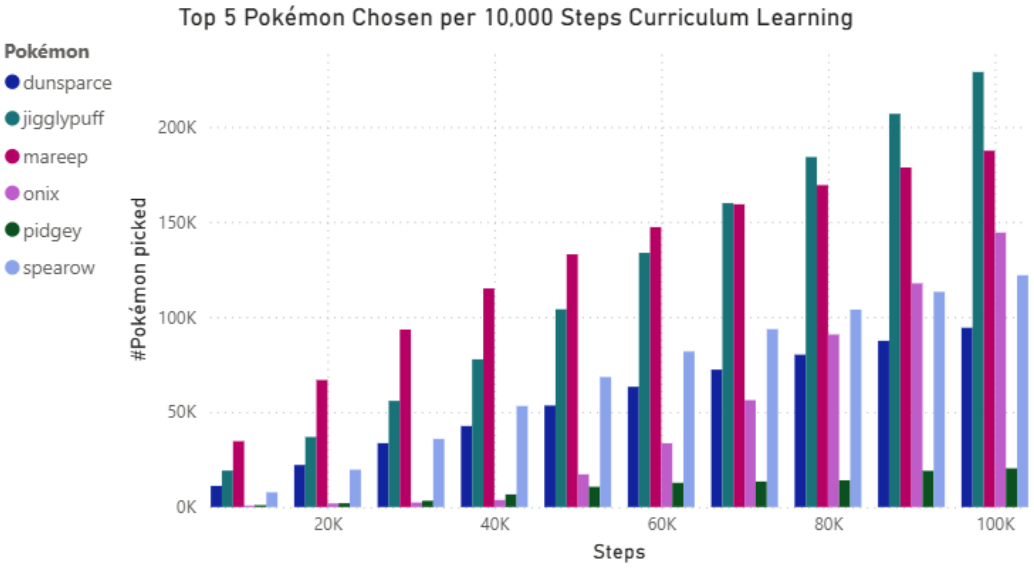


Figure 9: Most Chosen Pokémon Gym 2 Curriculum Agent

## 7.3 Gym 3

The results for gym 3 further follow the theme of the previous gym. In this case, when looking at figure 10, the advantage the curriculum agent has because over the baseline agent is so good that the algorithm starts on its conversion point, meaning the agent does not have to learn anything. The baseline agent on the other hand does very similar to the baseline agent in the previous gyms.

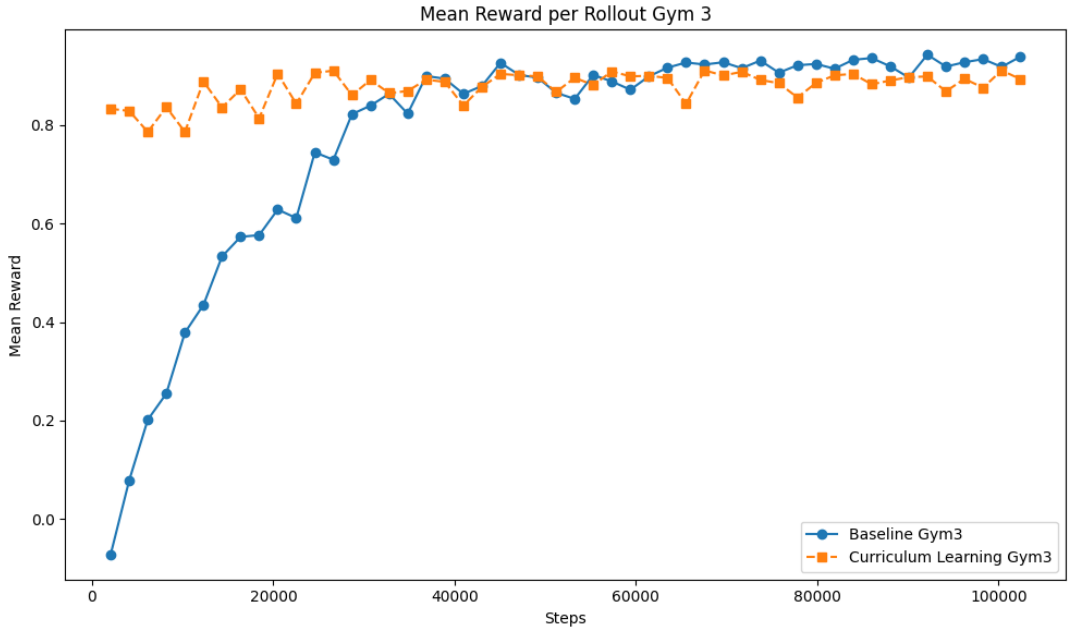


Figure 10: Learning Curve of Gym 3

The figures regarding the chosen Pokémon show a similar behavior. The curriculum agent in figure 12 shows an early preference towards Onix. A logical choice because of that is what the previous gym required. Conversely, the baseline agent in figure 11 takes around 40000 steps to reliably pick Onix for the team resulting in the delayed convergence in figure 10.

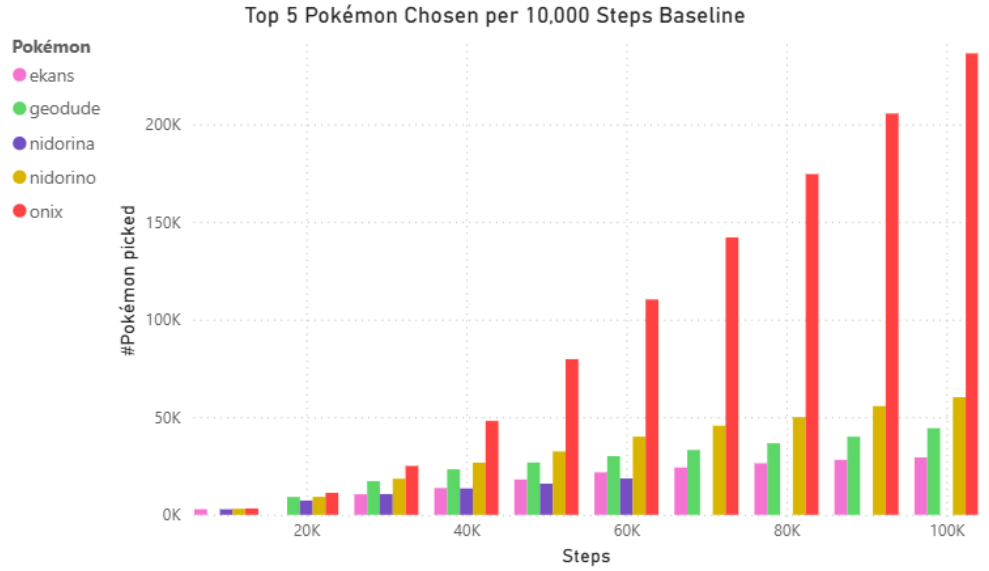


Figure 11: Most Chosen Pokémon Gym 3 Baseline Agent

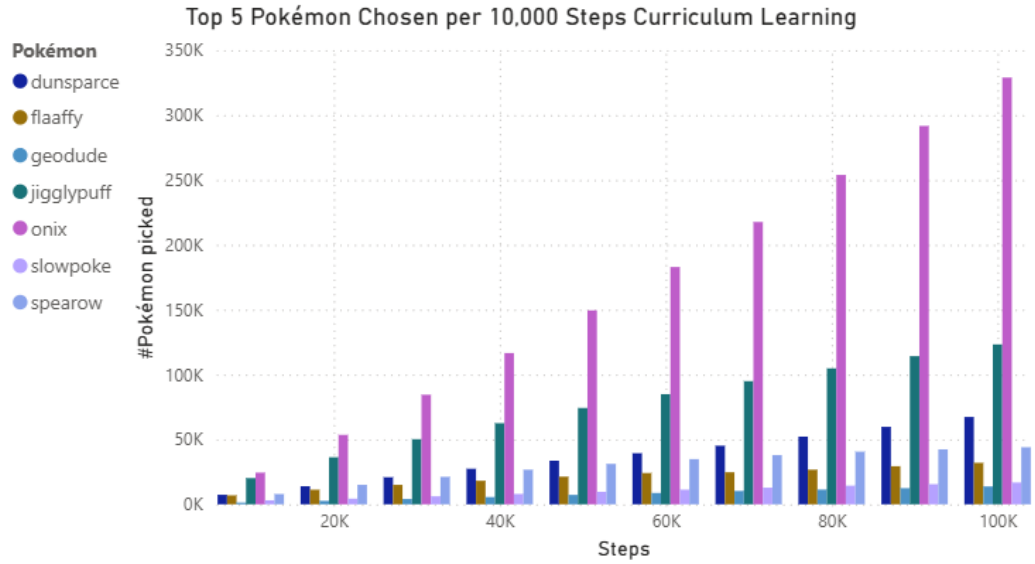


Figure 12: Most Chosen Pokémon Gym 3 Curriculum Agent

## 7.4 Gym 4

The previous three gyms show a similar trend in which the curriculum learning has an advantage because of its previous iterations. The results of gym 4 show that this trend does not continue for the next gyms. Figure 13 shows a nearly identical trajectory for both the baseline agent and the curriculum agent. Starting of with a small mean reward, but both quickly converging to a high mean reward.

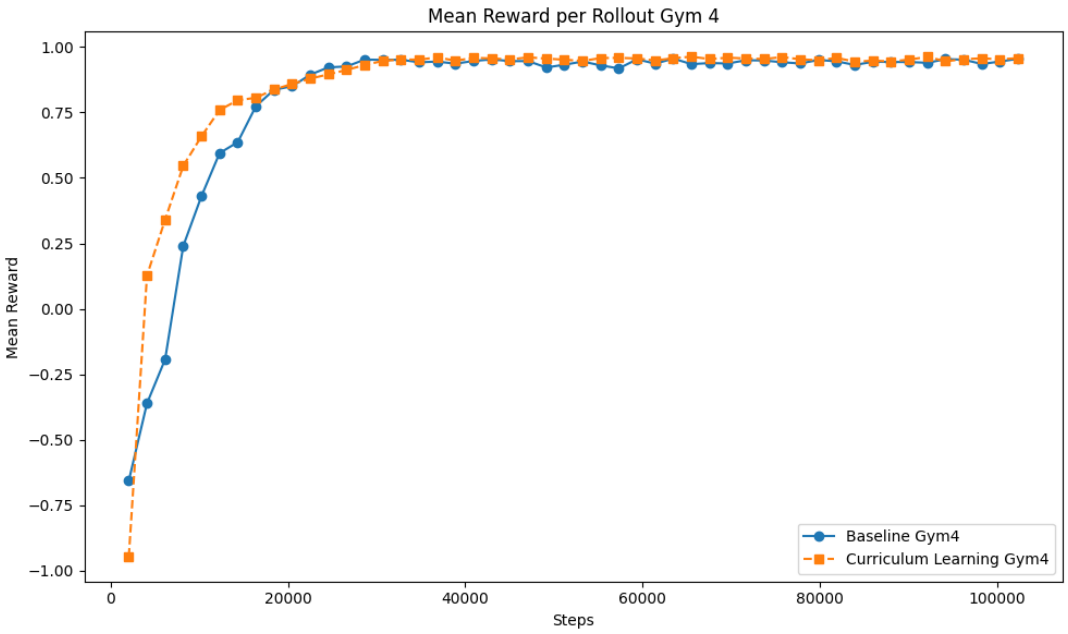


Figure 13: Learning Curve of Gym 4

The preference for Pokémon for both agents is also different from previous gyms. Umbreon, one of the very few dark types, can be seen to be the top pick for both the baseline agent and the curriculum agent in figure 14 and 15 respectively. This gym being dominated by Umbreon suggests that the developers wanted to force the player to use the newly introduced dark type (section 4.1).

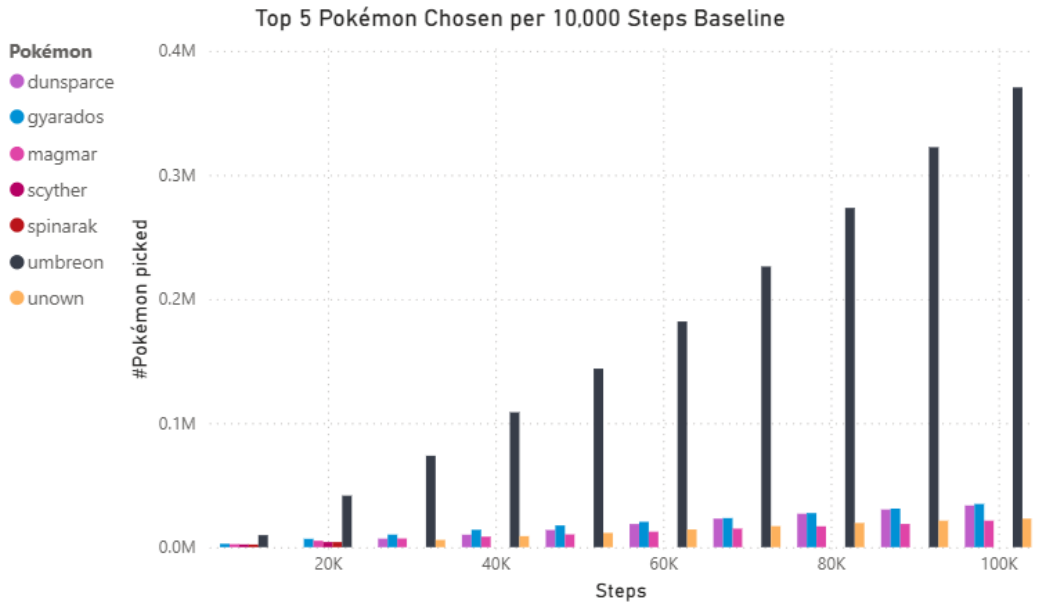


Figure 14: Most Chosen Pokémon Gym 4 Baseline Agent

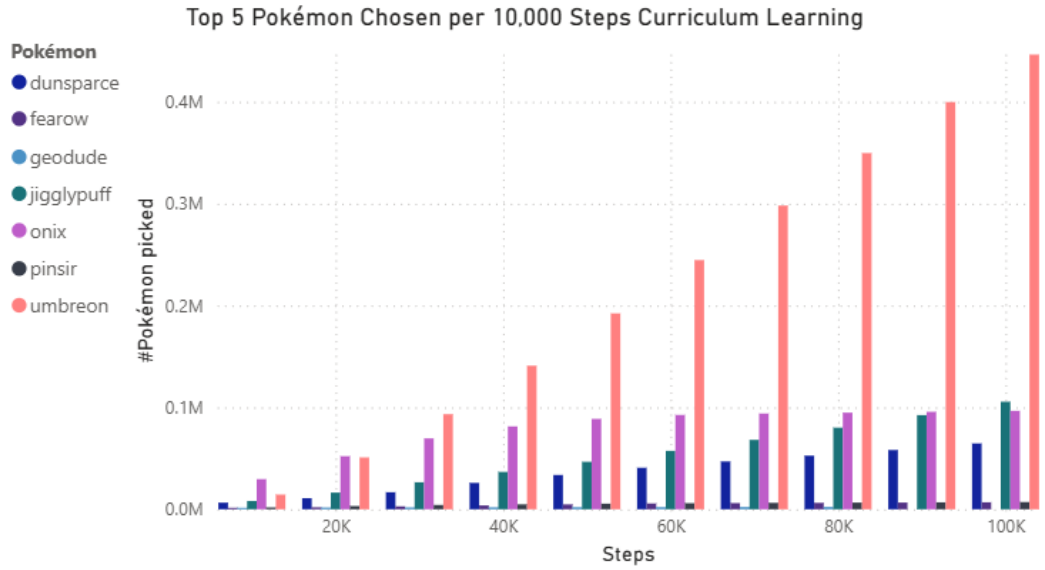


Figure 15: Most Chosen Pokémon Gym 4 Curriculum Agent

## 7.5 Gym 5

Gym 5, again, has two similar learning curves for both agents as seen in figure 16. A thing to note is that both agents achieve a perfect mean reward after 25000 steps. This indicates that the gym might have been too easy, and when looking at the gym leader (section 4.1) one can see that this gym leader only has two Pokémon.

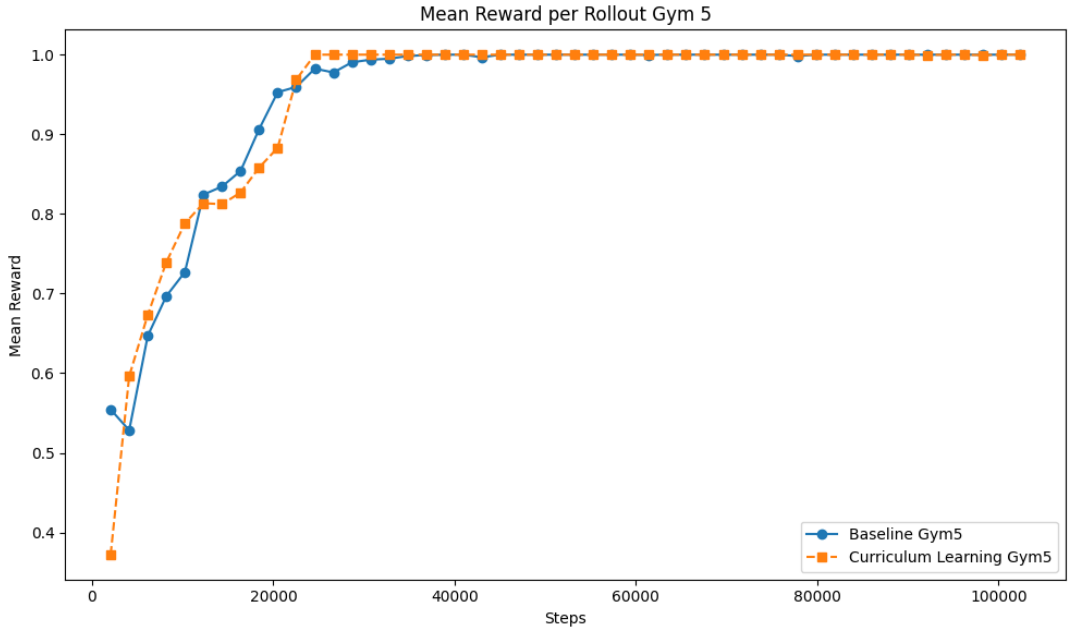


Figure 16: Learning Curve of Gym 5

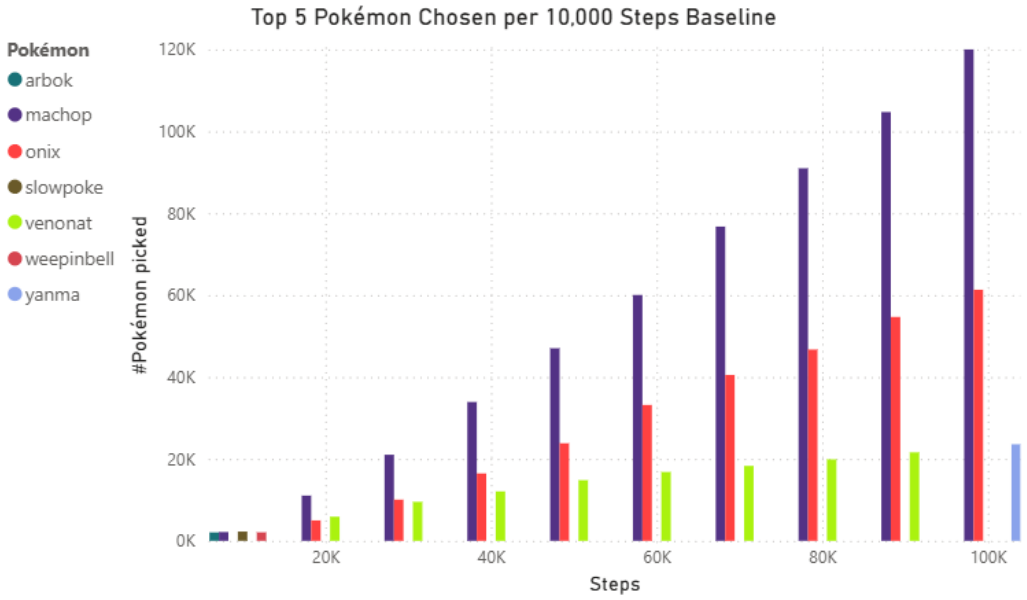


Figure 17: Most Chosen Pokémon Gym 5 Baseline Agent

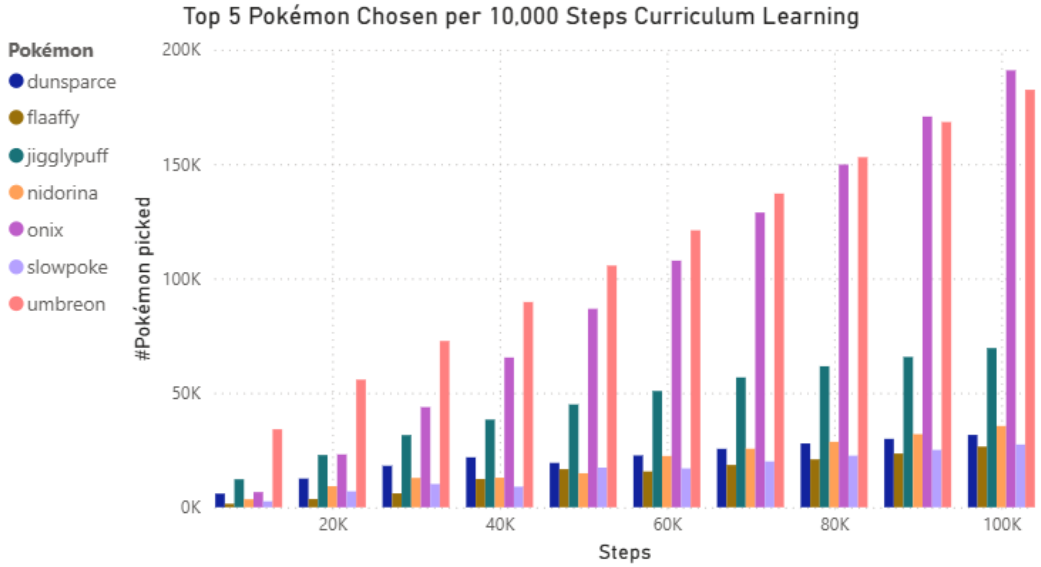


Figure 18: Most Chosen Pokémon Gym 5 Curriculum Agent

## 7.6 Gym 6

The learning curves of the two agents for gym 6 are again very similar to each other. Figure 19 shows that both agents had a harder time to converge than in the previous gym

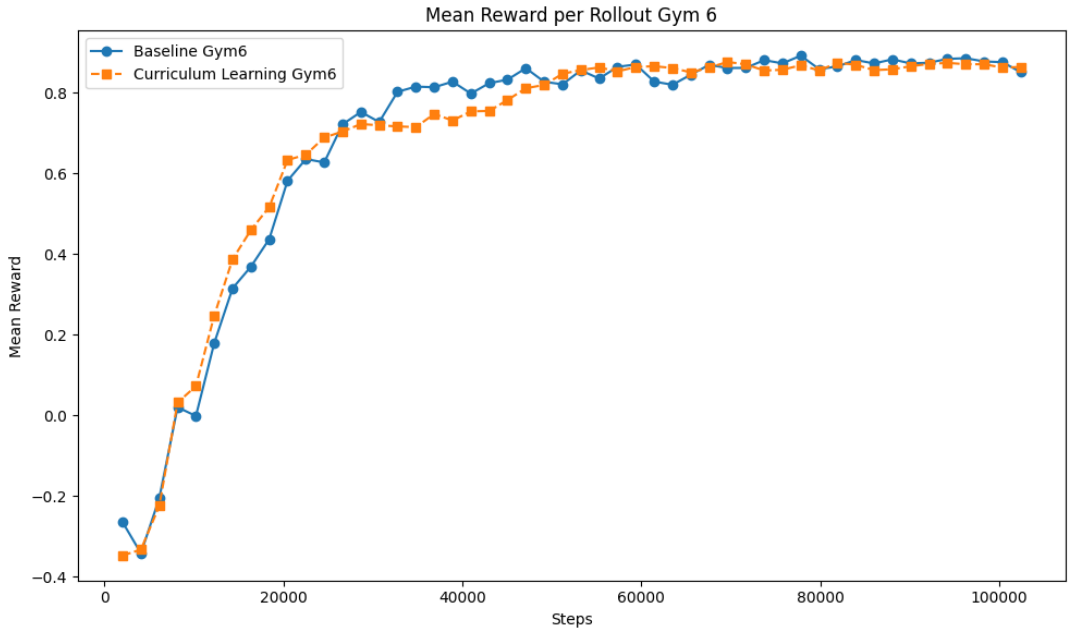


Figure 19: Learning Curve of Gym 6

Eventhough both agents show a preference for Magmar, figure 21 shows the bias of the curriculum agent in choosing the Onix.

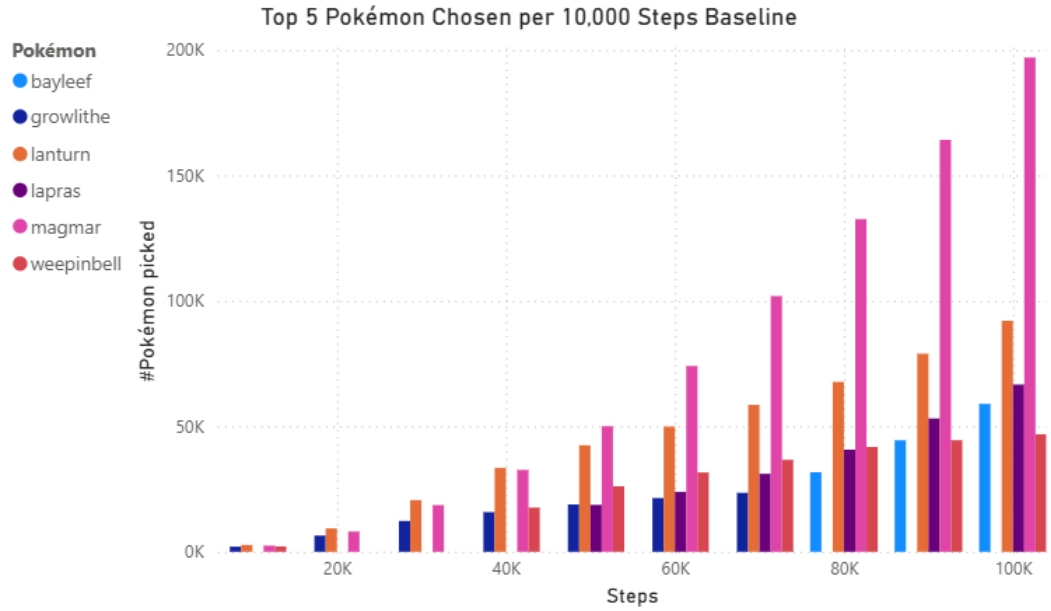


Figure 20: Most Chosen Pokémon Gym 6 Baseline Agent

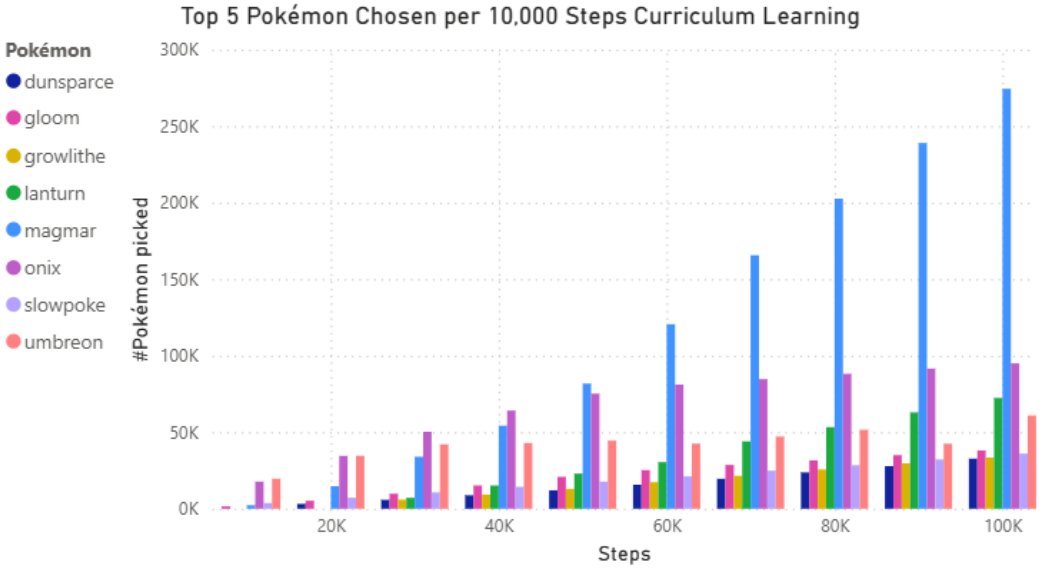


Figure 21: Most Chosen Pokémon Gym 6 Curriculum Agent

## 7.7 Gym 7

After three gyms of nearly identical learning curves for both agents, the figure 22 shows similar learning curves to gym 2 7. It shows that even though the curriculum agent has a promising initial reward. Its bias for Magmar, as seen in figure 24, from the previous gym prevents the curriculum algorithm from converging. Only after 80000 steps, the curriculum agent starts to use Corsola more leading to the same convergence as the baseline agent. This agent quickly learns about the use of Corsola and needs half the amount of steps of the curriculum agent to reach convergence, as seen in figure 23.

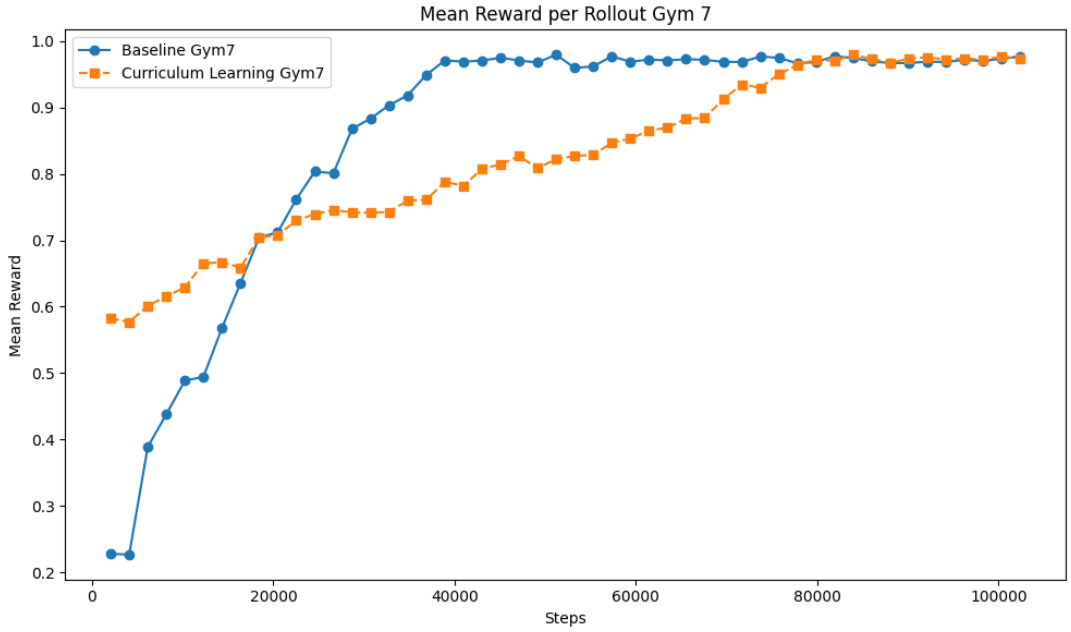


Figure 22: Learning Curve of Gym 7

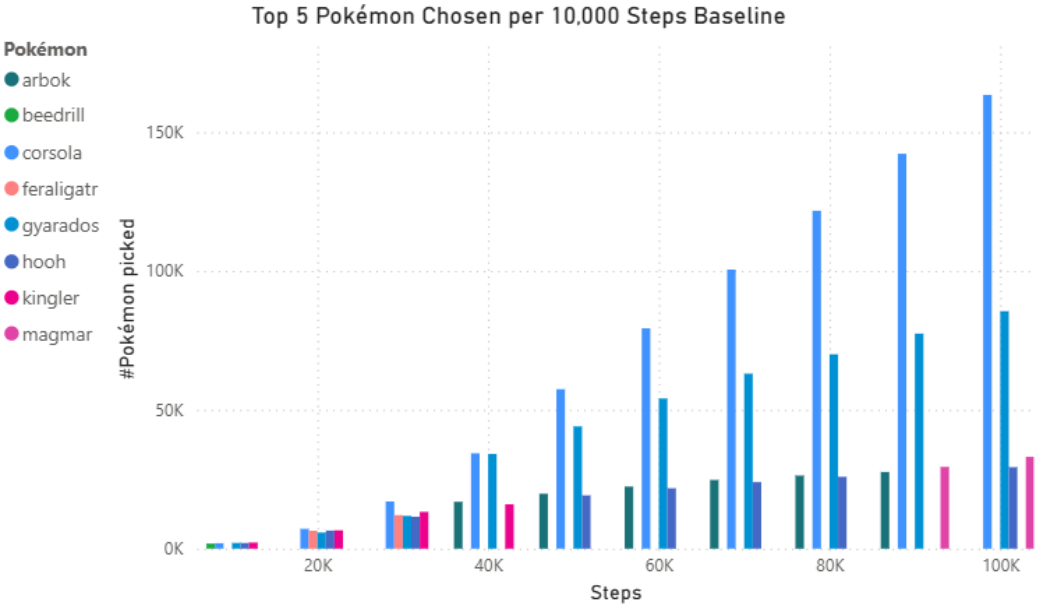


Figure 23: Most Chosen Pokémon Gym 7 Baseline Agent

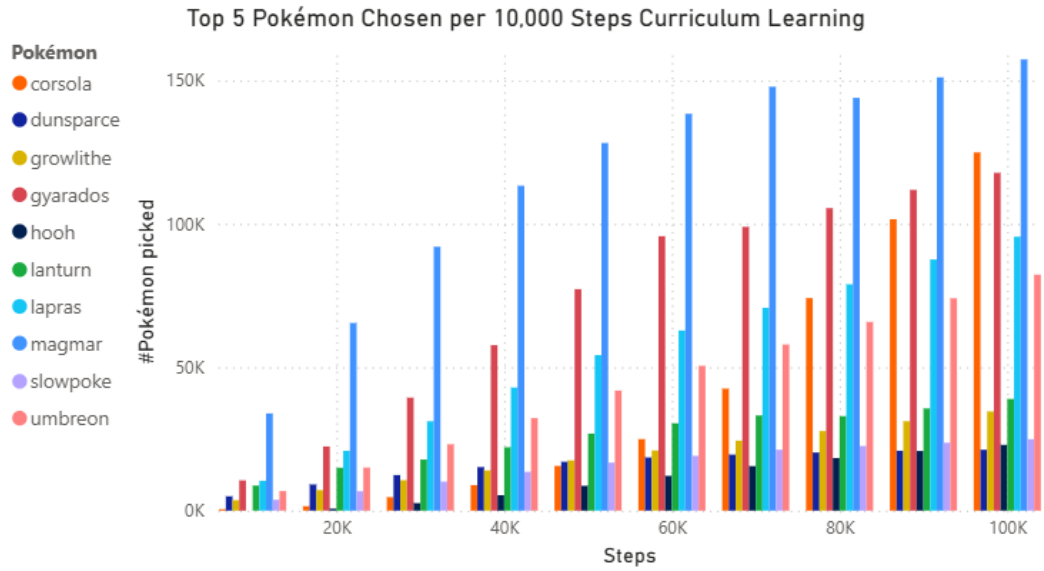


Figure 24: Most Chosen Pokémon Gym 7 Curriculum Agent

## 7.8 Gym 8

Gym 8 is the last gym of the base game and it shows in the results. Looking at figure 25, both agents have a hard time finding teams to even consistently win, especially the curriculum agent.

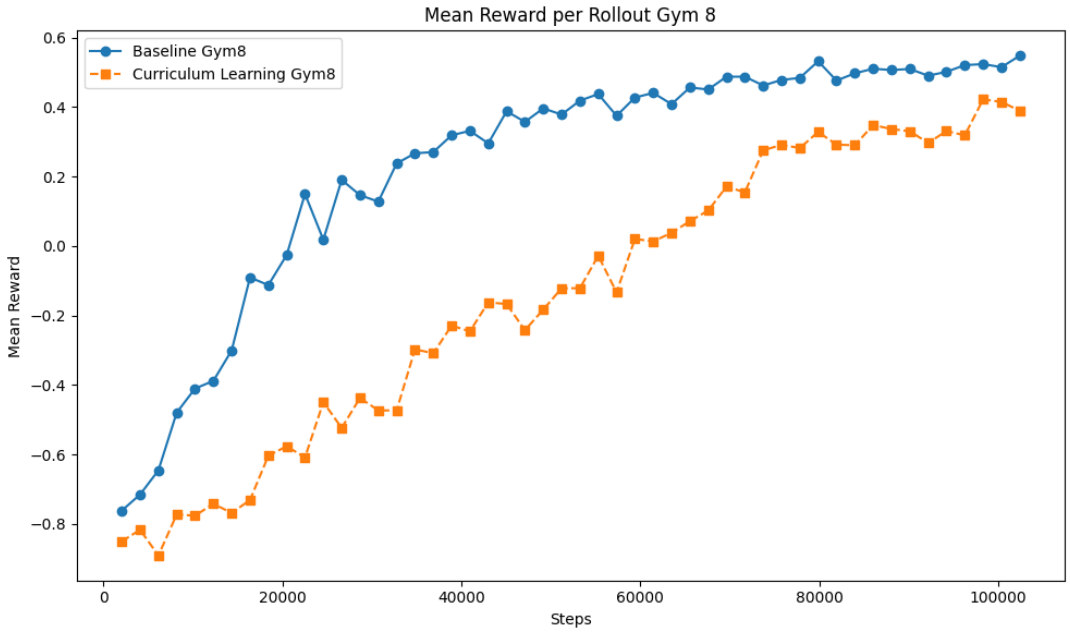


Figure 25: Learning Curve of Gym 8

The Pokémon the agents also have different preferences. Whereas the baseline agent solely chooses Pidgeot in figure 26, the curriculum agent tries a lot of the options from the previous gyms, including Corsola and Umbreon.

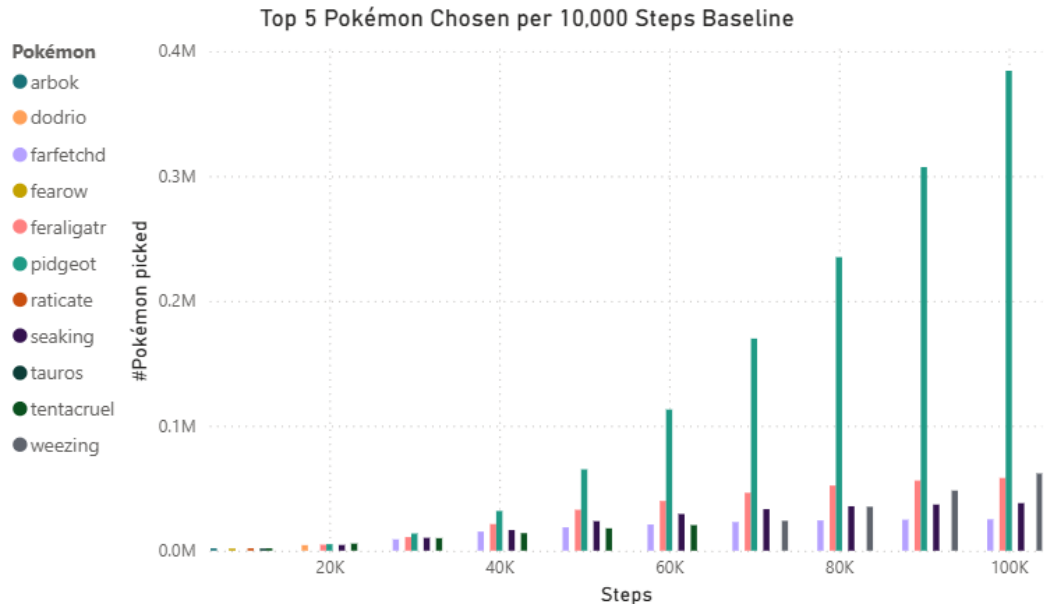


Figure 26: Most Chosen Pokémon Gym 8 Baseline Agent

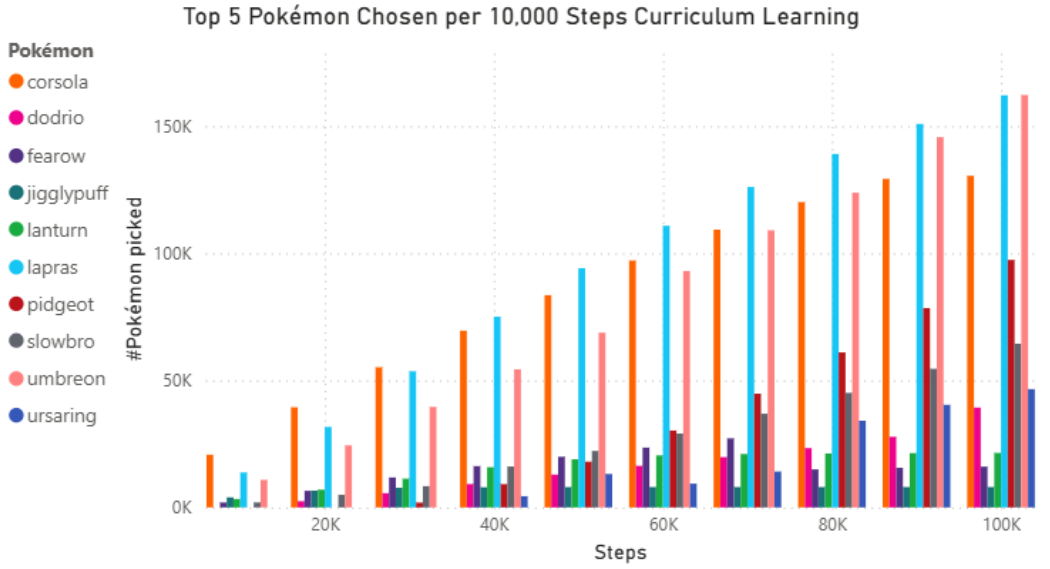


Figure 27: Most Chosen Pokémon Gym 8 Curriculum Agent

# 8 Discussion

The results in section 7 can naturally be separated into three groups: the early game, the mid game and the late game. The main focus will be on how the curriculum learning agent mimics player experience and how it compares to the baseline agent. We will reference this guide [12] (which will now be referred to as the meta) to see how both agents compare to these existing strategies.

## 8.1 Early Game

**Gym 1** In gym 1 graph 4 both agents follow a very similar trajectory and Mareep ends up being the dominant choice. The meta also states that Mareep is one of the best choices for this gym, indicating that the core gameplay of RQ1 (section 3.1) was not altered by the assumptions.

**Gym 2** In gym 2 the effects of using curriculum learning can be seen in the. The initial information contained in the curriculum learning model from the previous iteration show a clear preference to the Mareep found in the last iteration. However, the meta states that the rock type Pokémon works best when dealing with gym 2. This can also be interpreted from the graph. The learning process of the curriculum agent can be described as a human process of trying what worked before and then learning what works better (RQ2). Also, both agents finding the right Pokémon type in the end contributes positively to RQ1.

**Gym 3** The results for this gym suggest that the curriculum agent already has perfect knowledge in comparison to the baseline agent. It starts at the optimum and stays there. When looking at the meta this makes sense, the guide suggests multiple strategies, one of which is using rock type Pokémon . The Onix the curriculum agent used in gym 2 is of rock type, therefore overfitting on solutions including said Onix result in an optimal strategy from the start. Therefore both agents again follow the meta (RQ1).

The results of the first three gyms suggest that this part of the game is relatively easy. The curriculum agent using the information for its benefit mimics what a player would do. This would make sense for the learning curve of a game (RQ3) since games tend to start easy and get harder as the player progresses.

## 8.2 Mid Game

Whereas in the early game the curriculum agent benefits from the information of previous iterations, the results for the mid game suggest a focus on exploration. Looking at the figures for gym 4 and gym 6 and referencing the availability of each Pokémon . Both agents have the same heavy favorite in Umbreon and Magmar for these gyms respectively. This shift focus, from getting one or two Pokémon and training them in the early game, to catching Pokémon of a specific type requires the player to learn different things. Therefore increasing the learning curve. Both of these gyms also follow the meta (RQ1), so the agents still follow the suggested game play.
Gym 5 is odd, both agents opt for different strategies but still manage to get a perfect reward. This suggests that the gym might be to easy for the player at that point. It could mean that the simplification of leveling is not working for this gym (RQ1) or that the game's learning curve is not on the same level as gym 4 and gym 6 (RQ3).

## 8.3 End Game

In the last two gyms both agents use different strategies. The meta for gym 7 states that a fighting Pokémon works best in this case, however the Corsola that both agents in the end found is not of fighting type (RQ1). The meta for gym 8 suggests a Piloswine, which is of ice and ground type. The Pidgeot is of fly type. Even though this result suggests that the assumptions made do not represent the game in the late game. The fact that the agents both have trouble finding the right Pokémon does suggest that the learning curve has increased yet again (RQ3). Also, the curriculum learning agent trying all Pokémon that have worked in gyms before which ties in with RQ2.

## 8.4 Notable Finding

The most notable finding from the results was that even though the agents mostly followed the meta, the starters Pokémon were never picked. All guides talk about how important the right starter is and that they can be very useful in most gym battles.

## 8.5 Regarding the Research Questions

The agents in the early and the mid game of Pokémon gold follow the meta (apart from gym 5), therefore it is reasonable to assume that the core game play was not altered (RQ1). In the late game however the agents find other strategies that are much different than the meta. The curriculum agent does on the other hand represent the player's thought process by first trying out things that work before trying new Pokémon. Thirdly, the curriculum agent also shows the increase of the difficulty curve of the game (RQ3).
By answering these subquestions, we can conclude that this research does help us analyze complex game experience (main research question), even though when games get more complex our approach may be too simple.

# 9 Limitations and Future Work

This chapter focuses on the limiting factors in this research, as well as how these limitations can be overcome in future work.

## 9.1 Limitations

Whereas assumptions can help shape research (section 5.1.2) they also cause limitations. Setting other core elements of the game as a constant limits the scope of which the game can be tested in. One the one hand this research gives us insight on whether a certain aspect of a game is working. On the other hand it does not cover its influence on the other aspects of the game. So to test how well the elements co-exist, one could require a combination of AI algorithms.

## 9.2 Future Work

We believe that future work would begin with expanding the action space further. The results suggest that the type is very important and therefore choosing the right moves and using HM's and TM's for each Pokémon can mimic the game better. This would come very close to battle simulation, since for each battle the best moves are different. This could even incorporate CL, expanding the action space to include move decisions after the agent learns how to build a team. Additionally, future work could include exploring the open world. This approach could trend towards curiosity driven learning [4]. In this case the agent would use intrinsic rewards to explore new states which aligns with exploring the open world of Pokémon. This approach has been used, using a multi-agent variant, to test 3D games [5] with promising results.
On the other hand, the opposite, counter-intuitive approach of anti-curriculum learning, could also be a way to tackle this team building problem.

# 10 Conclusion

To conclude, the early game and the mid game are represented well despite the assumptions to simplify the game. Curriculum learning as a method did not improve the agent's ability to converge in later gyms, however it did give us insight in how the player would play and in the learning curve of the game. Therefore, we can analyze Pokémon Gold reasonably well while keeping the game simple. For future work, there could be less assumptions and therefore less limitations, but this would mean a more complex problem for the AI agent to deal with.

# References


[1] Eloi Alonso et al. "Deep reinforcement learning for navigation in AAA video games". In: *arXiv preprint arXiv:2011.04764* (2020).

[2] Bowen Baker et al. "Emergent tool use from multi-agent autocurricula". In: *International conference on learning representations*. 2019.

[3] Yoshua Bengio et al. "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 41–48.

[4] Yuri Burda et al. "Large-scale study of curiosity-driven learning". In: *arXiv preprint arXiv:1808.04355* (2018).

[5] Raihana Ferdous et al. "Curiosity Driven Multi-agent Reinforcement Learning for 3D Game Testing". In: *2025 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE. 2025, pp. 121–129.

[6] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[7] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[8] Andrew Patterson et al. *Empirical Design in Reinforcement Learning*. 2024. arXiv: 2304.01315 [cs.LG]. URL: https://arxiv.org/abs/2304.01315.

[9] Marco Pleines et al. "Pokemon red via reinforcement learning". In: *arXiv preprint arXiv:2502.19920* (2025).

[10] Antonin Raffin et al. "Stable-baselines3: Reliable reinforcement learning implementations". In: *Journal of machine learning research* 22.268 (2021), pp. 1–8.

[11] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[12] Spirit Shackle. *Pokémon Gold and Silver Walkthrough and Capture Guide - PokéCommunity Daily — daily.pokecommunity.com*. https://daily.pokecommunity.com/2017/09/21/pokemon-gold-and-silver-walkthrough-and-capture-guide/. [Accessed 19-08-2025].

[13] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[14] *Stat — bulbapedia.bulbagarden.net*. https://bulbapedia.bulbagarden.net/wiki/Stat. [Accessed 12-08-2025].

[15] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. Vol. 1. 1. MIT press Cambridge, 1998.

[16] Bhargava Uppuluri et al. "CuRLA: Curriculum Learning Based Deep Reinforcement Learning for Autonomous Driving". In: *arXiv preprint arXiv:2501.04982* (2025).

[17] Jett Wang. "Winning at pokémon random battles using reinforcement learning". PhD thesis. Massachusetts Institute of Technology, 2024.

[18] Ziyi Wang et al. "Learning State-Specific Action Masks for Reinforcement Learning". In: *Algorithms* 17.2 (2024). ISSN: 1999-4893. DOI: 10.3390/a17020060. URL: https://www.mdpi.com/1999-4893/17/2/60.

[19] Chao Yu et al. "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24611–24624. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9c1535a02f0ce079433344e14d910597-Paper-Datasets_and_Benchmarks.pdf.