UNIVERSITÀ
DEGLI STUDI DI TRIESTE

DATA SCIENCE &
SCIENTIFIC COMPUTING

Exploring Reinforcement Learning in Pokemon Battles

Reinforcement Learning Project

Matteo Boi

A.A. 2022/2023

Index

Pokemon Battles as RL environment

Complex problem:

- Current pokemon
- Status conditions
- HP level
- Stat changes
- Opponent's pokemon
- Move informations
- Move usage
- State of the player's team
- Possibility to switch pokemon
- Battlefield condition
- Items

Pokemon Battles as RL environment

Semplification:

- Agent = Pokemon.
- Each pokemon has one type, no stats, same HP.
- Only generic moves (fire, water, …).
- No items.

Goal: to make the agent learn type advantages.

- Fire is supereffective against grass (2x damage).
- Fire is not very effective against water (0.5 damage).

Environment V0

- **3 possible types (fire, water, grass)**

- **Action space:** fire move, water move, grass move.
- **Observation:** type of the opponent pokemon.

- **Episode:** entire battle (until one of the two pokemons faints)

- **Two reward functions:**
  - Full reward ("guided"): +1 if the move used is supereffective
    0 if the move is standard
    -1 if the move is not very effective
    + (remaining HP) of the agent pokemon at the end of the episode

  - HP reward: = (remaining HP) no guidance during the episode

Environment V1

- **4 possible types (normal, fire, water, grass)**

- **Action space:** normal move, fire move, water move, grass move.
- **Observation:** type of the opponent pokemon.

- **Episode:** entire battle (until one of the two pokemons faints)

- **Two reward functions:**
  - <u>Full reward</u> ("guided"): +1 if the move used is superffective
    0 if the move is standard
    -1 if the move is not very effective
    + (remaining HP) of the agent pokemon at the end of the episode

  - <u>HP reward:</u> = (remaining HP) no guidance during the episode

Environment V1

- **4 possible types (normal, fire, water, grass)**

- **Action space:** normal move, fire move, water move, grass move.
- **Observation:** (type of the agent pokemon, type of the opponent pokemon).

- **Episode:** entire battle (until one of the two pokemons faints)

- **Additional damage:** if the move used is of the same type of the pokemon

- **Two reward functions:**
  - <u>Full reward</u> ("guided"):
    - As before.
    - +0.5 if the move used is of the same type of the pokemon.
  - <u>HP reward:</u>
    - As before.

Environment V2

- **6 possible types (normal, fire, water, grass, fighting, flying)**
- **Random moveset:** (move of the same type of the pokemon,
  3 random moves between the remaining types).
- **Action space:** 4 moves in the moveset
- **Observation:** (type of the agent pokemon, — 6 possible values
  type of the opponent pokemon, — 6 possible values
  types of the 3 moves in the moveset) — 6 possbile values for each move

Q-learning

Off-policy temporal difference control with ε-greedy action selection.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right]$$

With Q that directly approximates q_* (optimal action-value function).

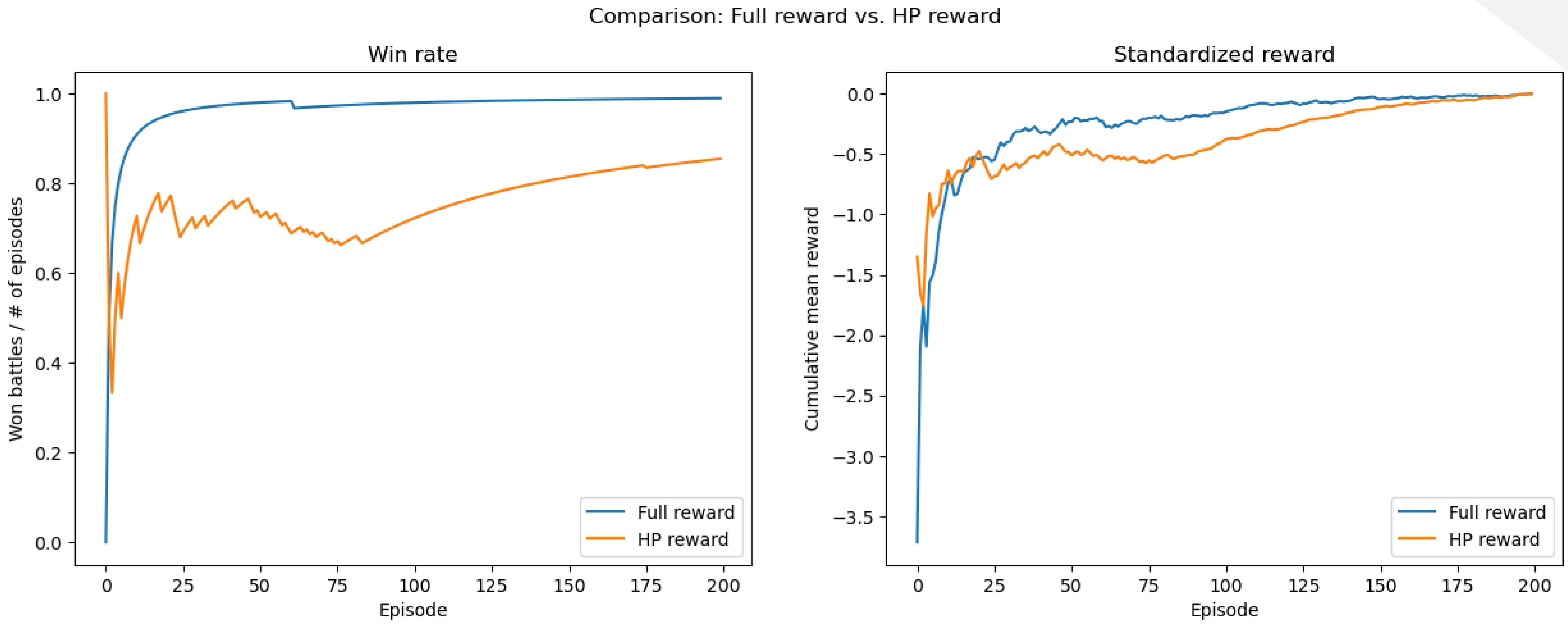
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

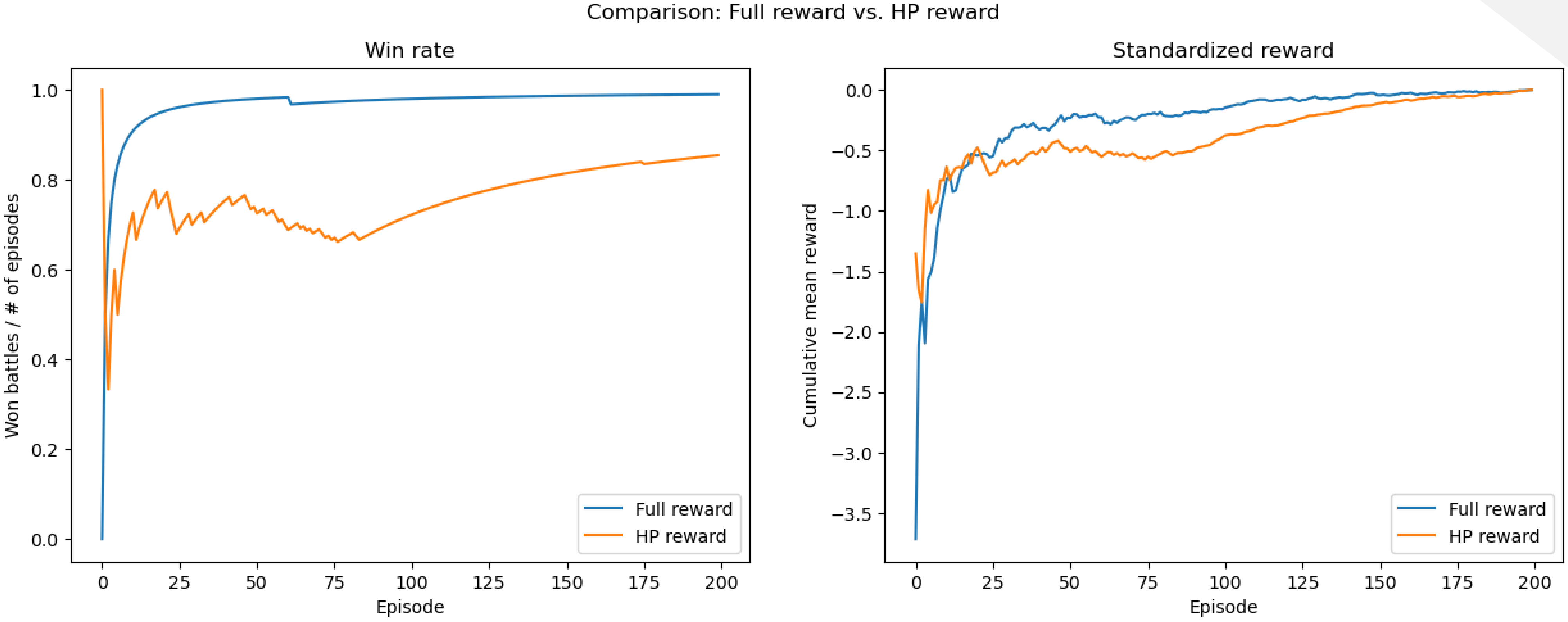
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$
    $S \leftarrow S'$
  until S is terminal

From Sutton & Barto 2020

Results: Environment V0



Results: Environment V0



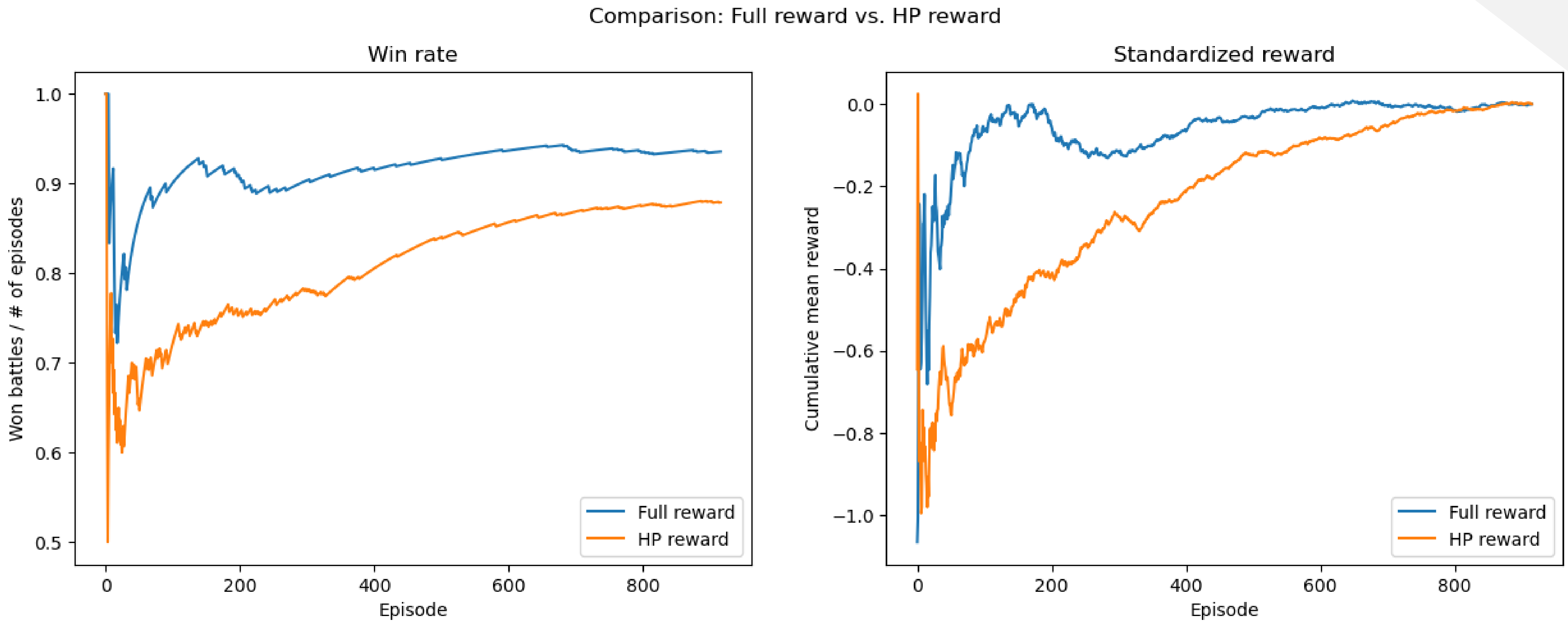
Convergence: $\sim$5 episodes for Full reward
$\sim$30 episodes for HP reward

Results: Environment V0

- **Optimal Policy:**

	Opponent type		
	Fire type	**Water type**	**Grass type**
Agent's move type	Water move	Grass move	Fire move

Results: Environment V1



Convergence: ~1100 episodes for Full reward
~1300 episodes for HP reward

Results: Environment V1

- **Optimal Policy:**

Agent type \ Opponent type	Normal type	Fire type	Water type	Grass type
Normal type	**Normal move**	Water move	Grass move	Fire move
Fire type	**Fire move**	Water move	Grass move	Fire move
Water type	**Water move**	Water move	Grass move	Fire move
Grass type	**Grass move**	Water move	Grass move	Fire move

Multi-Armed Bandit

Observation: the state never changes during the episode.

Multi-Armed bandit algorithm: ε-greedy action selection.

- **Arms:** actions available to the pokemon

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

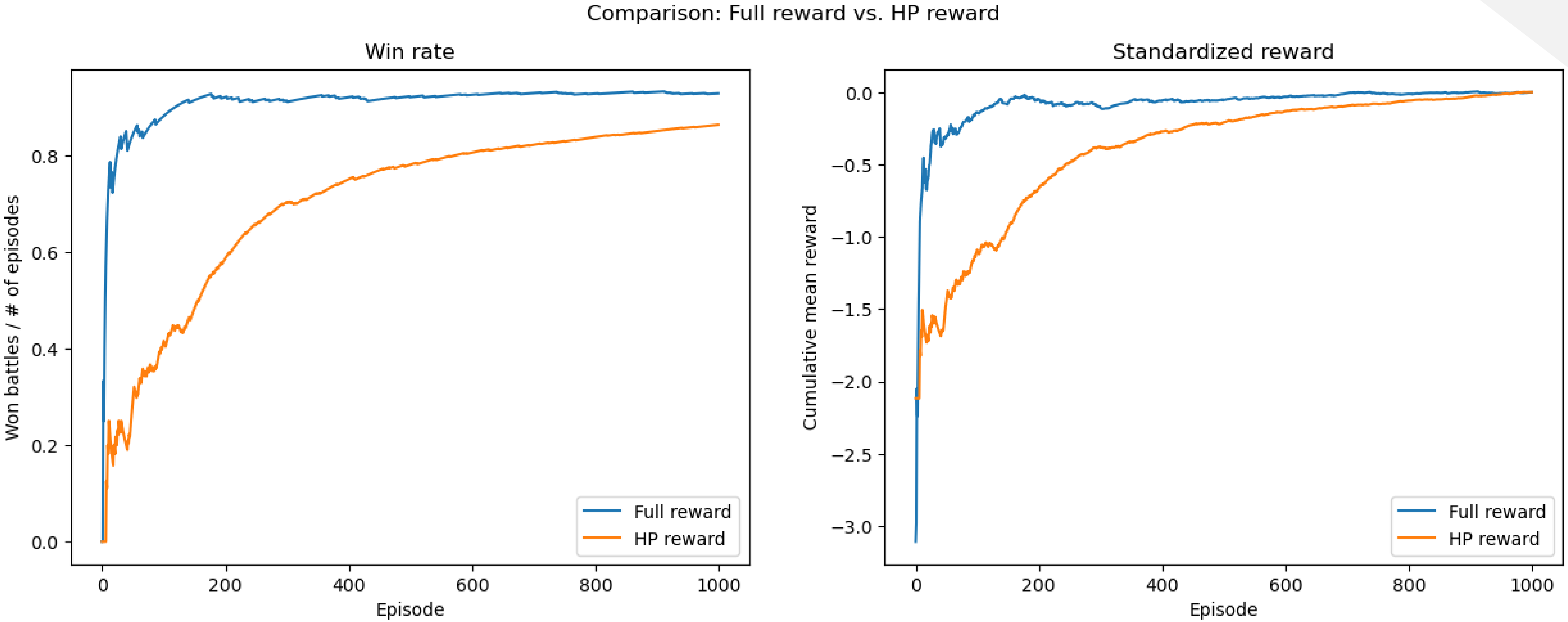
$R \leftarrow bandit(A)$

$N(A) \leftarrow N(A) + 1$

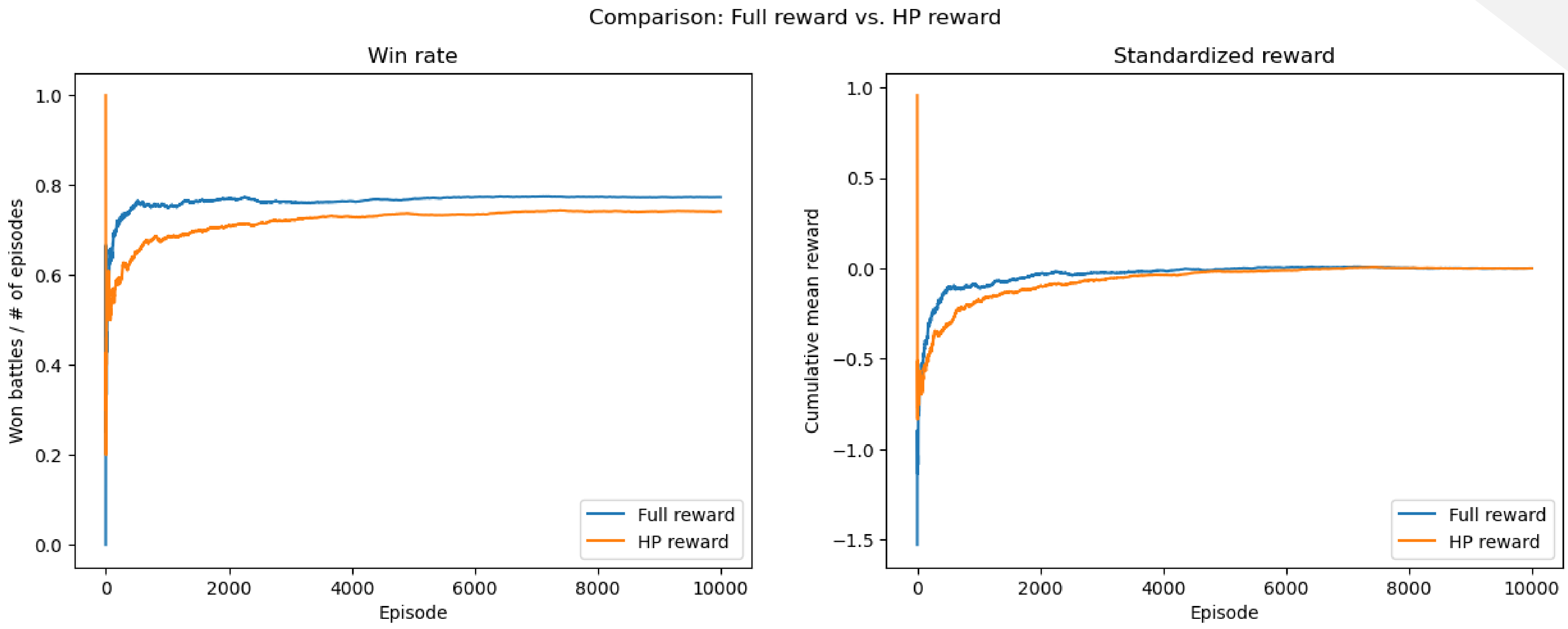
$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\left[R - Q(A)\right]$

From Sutton & Barto 2020

MAB Results: Environment V0

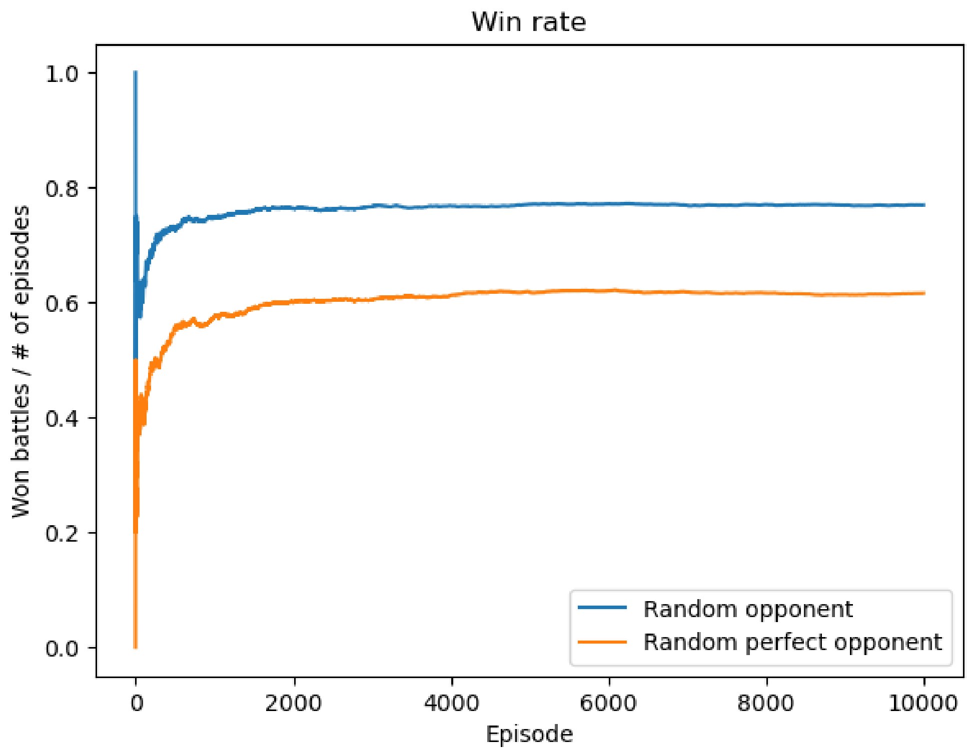


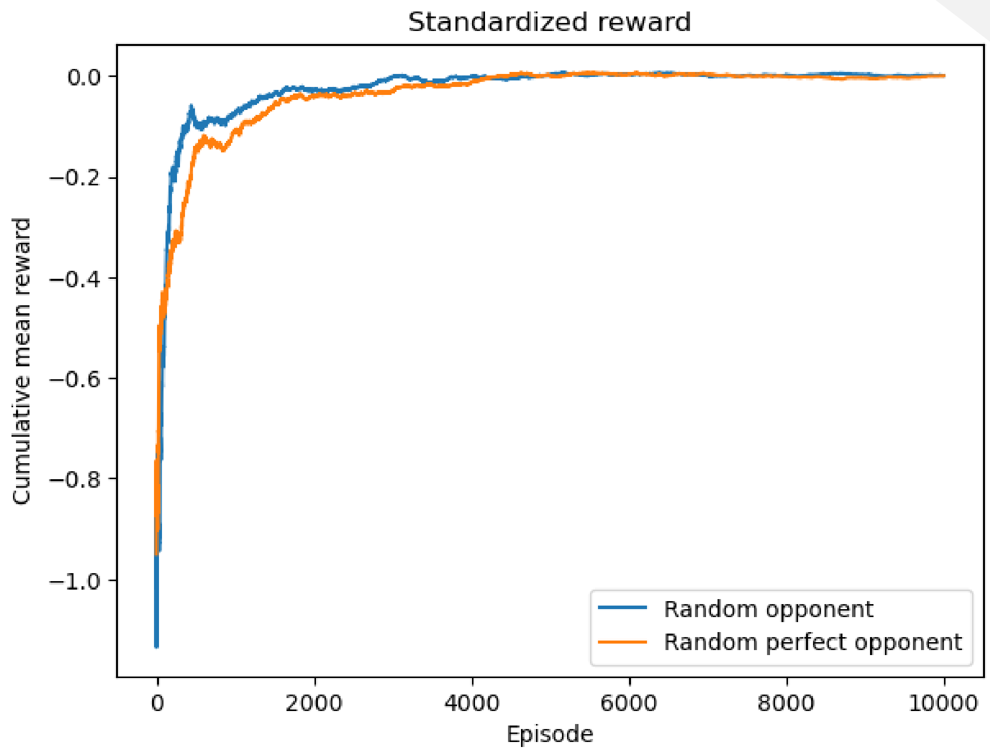
MAB Results: Environment V1



MAB Results: Environment V1

- **Random-perfect opponent:** chooses a random move with $p = 0.5$
  chooses the supereffective move otherwise





Observation: a good portion of the lost battles is due to bad matchups

Improvements

- **Learning opponent** instead of a dummy one.

- **More complex scenarios:**
  - HP as observation.
  - Integration of more pokemon types.
  - Strategic moves for more complex interactions.

- **Dynamic turn order system:** based on the pokemon speed.

- **Partial observability of the opponent:** hidden type and moves.

Thank You.