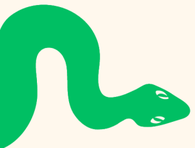
UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

A. Y. 2024 / 2025

Bredariol F., Savorgnan E., Tic R.



UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

our
WORKFLOW

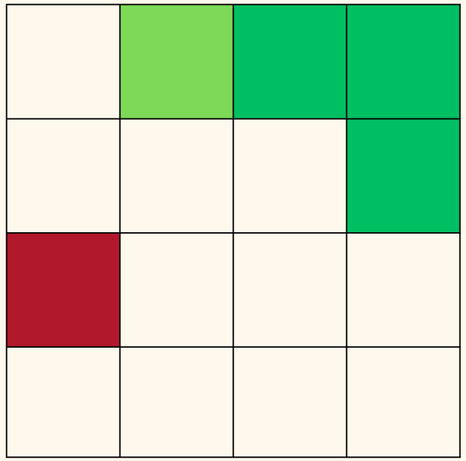
1 environment definition

2 algorithms implementation

3 learning and evaluation

UNIVERSITÀ DEGLI STUDI DI TRIESTE

environment MDP



STATE REPRESENTATION

matrix m x n with 4 possible values

- Empty (0)
- Food (1)
- Head (2)
- Tail (3)

ACTIONS movement in 4 direction

TRANSITION

perfectly deterministic, except for food repositioning

REWARD

- +Re for eating
- -Rd for dying
- -eps for living

DIMENSION OF THE STATE SPACE?

$$|S| \approx (m \times n)^2 \times 2^{m \times n}$$

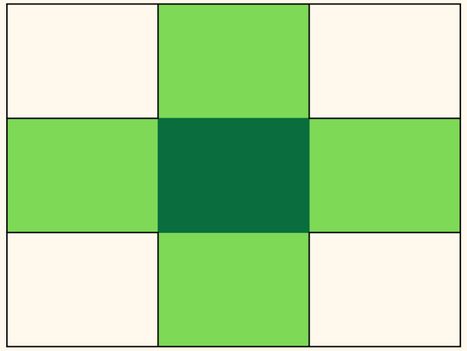
THIS IS A ROUGH APPROXIMATION

UNIVERSITÀ DEGLI STUDI DI TRIESTE

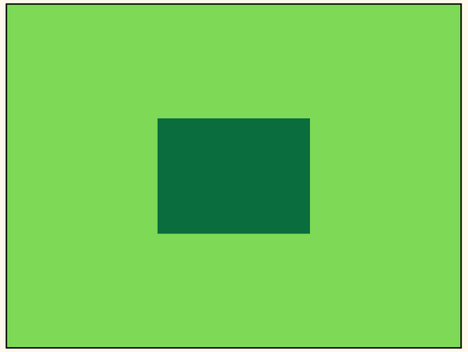
binning

Binning (state agglomeration or state discretization) is a technique used to **reduce the complexity of the state space** by **grouping** (or aggregating) similar states **into bins** or clusters, effectively coarsening a continuous or high-dimensional state space into a smaller, more manageable discrete one.

neighborhood



Von Neumann radius 1



Moore raidus 1

UNIVERSITÀ DEGLI STUDI DI TRIESTE

binning

FOOD INFORMATION

FOOD RELATIVE POSITION
FOOD DIRECTION

The food information is a **vector** encoding the position on the food wrt the position of the head. Given the **food position** (a_y, a_x) and the **head position** (h_y, h_x) we compute:

RELATIVE POSITION

$$(a_y - h_y, a_x - h_x)$$

DIRECTION

$$(a_y < h_y, a_y > h_x, a_x < h_x, a_x > h_x)$$

NEIGHBORHOOD INFORMATION

MOORE NEIGHBORHOOD
VON NEUMANN NEIGHBORHOOD

Information in the neighborhood is **encoded as vector**. Each element of the vector represents the state of a cell in the neighborhood: cell **occupied** by the tail are encoded by a **1**, **otherwise** by a **0**.

ALL THESE BINNINGS CAN BE COMBINED TO PRODUCE MORE COMPLEX STRUCTURES

*we actually tried to encode also tail information without results

UNIVERSITÀ DEGLI STUDI DI TRIESTE

model free

On-policy

Experience is generated from the learned policy

Sarsa & MonteCarlo

Off-policy

Experience is generated from a policy different to the one learned

Q-Learning

Bootstrap

Experience is used to estimate states' values

- Lookup-tables methods
- Use of default dict to manage with unfixed state dimensions

COMPLETELY ARBITRARY AND NOT OPTIMIZED

UNIVERSITÀ DEGLI STUDI DI TRIESTE

model control

OUR BASELINE

Q-VALUES MAP

$$Q(s,a) = Q(s,a) + l_r \delta(s,a)$$

THIS IS A WELL DEFINED QUANTITY

1 **Monte Carlo**

$$\delta = \sum_{t=k}^{\infty} \gamma^{k-t} R_k - Q(s,a)$$

2 **SARSA**

$$\delta = R + \gamma Q(s', a \sim \pi(s)) - Q(s,a)$$

3 **Q-Learning**

$$\delta = R + \gamma \max_a Q(s',a) - Q(s,a)$$

*we implemented various version of these algorithms, using also eligibility trace and other tricks

UNIVERSITÀ DEGLI STUDI DI TRIESTE

policy gradient

We parametrize $$\pi_\theta(a|s) = \frac{e^{\theta_{sa}}}{\sum_{a'} e^{\theta_{sa'}}}$$

We follow
$$\begin{aligned}\nabla_\theta G &= \mathbb{E}[Q(S,A)\nabla_\theta \log \pi(A|S)] \\ &= \mathbb{E}[\ \big(Q(S,A) - V(S)\big)\ \nabla_\theta \log \pi(A|S)] \\ &= \mathbb{E}[Ad(S,A)\nabla_\theta \log \pi(A|S)]\end{aligned}$$

This enables the policy to be **stochastic**

UNIVERSITÀ DEGLI STUDI DI TRIESTE

policy gradient
actor critic

for each episode

$$S \sim \rho_0$$

for $t = 0, 1, \ldots, T$

$$A_t \sim \pi_\theta(\cdot \mid S_t)$$

BOOTSTRAPPING

$$\delta_t = R_t + \gamma V(S_{t+1}) - V(S_t)$$ Advantage

$$V(S_t) = V(S_t) + \alpha_V \delta_t$$ Critic

$$\theta = \theta + \alpha_A \delta_t \nabla_\theta \log(\pi(A_t|S_t))$$ Actor

UNIVERSITÀ DEGLI STUDI DI TRIESTE

policy gradient

GAE

for each episode

run the full episode using π_θ

for $t = T, T-1, \ldots, 0$

$$\delta_t = R_t + \gamma V(S_{t+1}) - V(S_t)$$

$$Ad_t^{GAE} = \sum_{\tau=0}^{T-t} (\gamma\lambda)^\tau \delta_{t+\tau}$$

$$V(S_t) = V(S_t) + \alpha_V Ad^{GAE}$$

$$\theta = \theta + \alpha_A Ad^{GAE} \nabla_\theta \log(\pi(A_t|S_t))$$

TRADE-OFF BETWEEN

- **BOOTSTRAPPING = STABILITY**
- **MONTECARLO = ACCURACY**

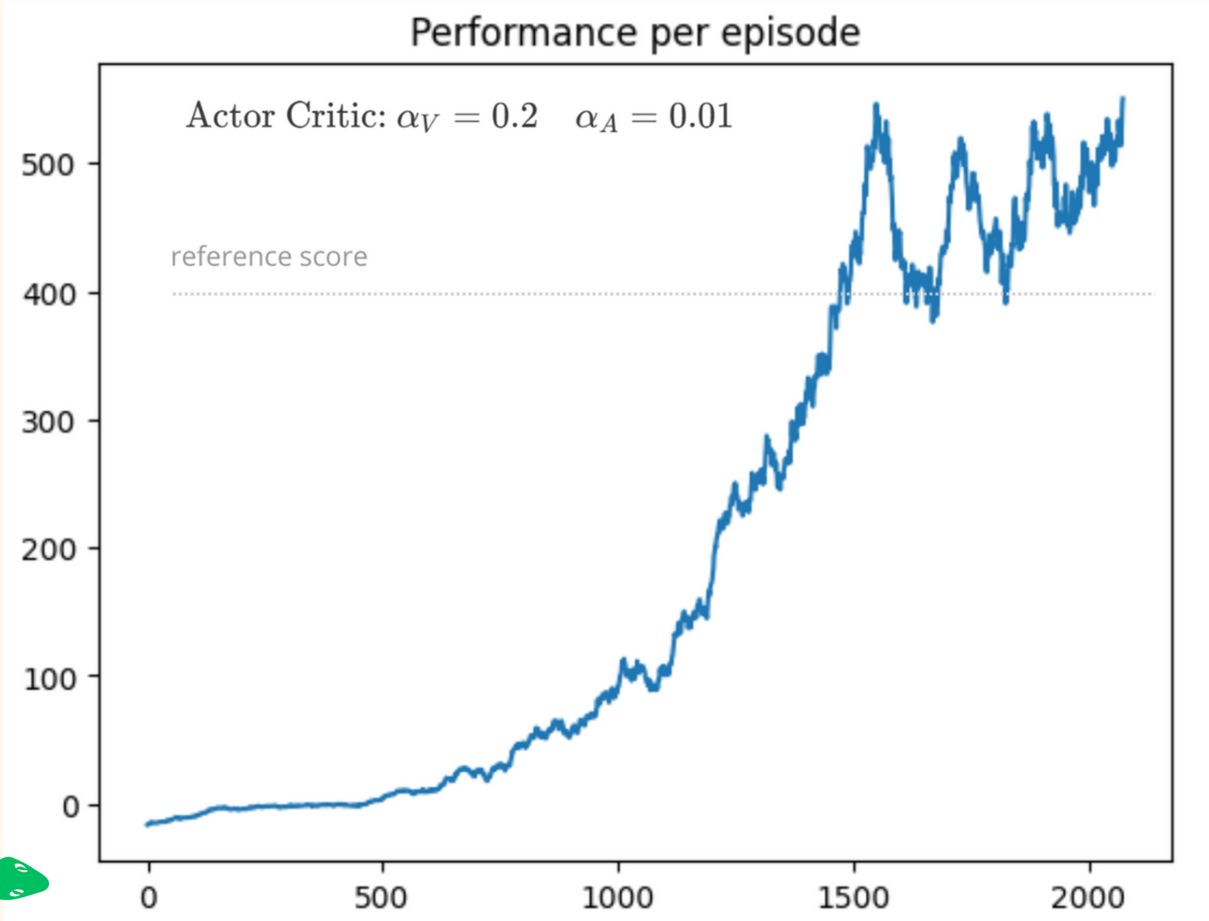
UNIVERSITÀ DEGLI STUDI DI TRIESTE

policy gradient

results

Von Neumann radius 1

$$\alpha_V = 0.9 \; \alpha_A = 0.09$$



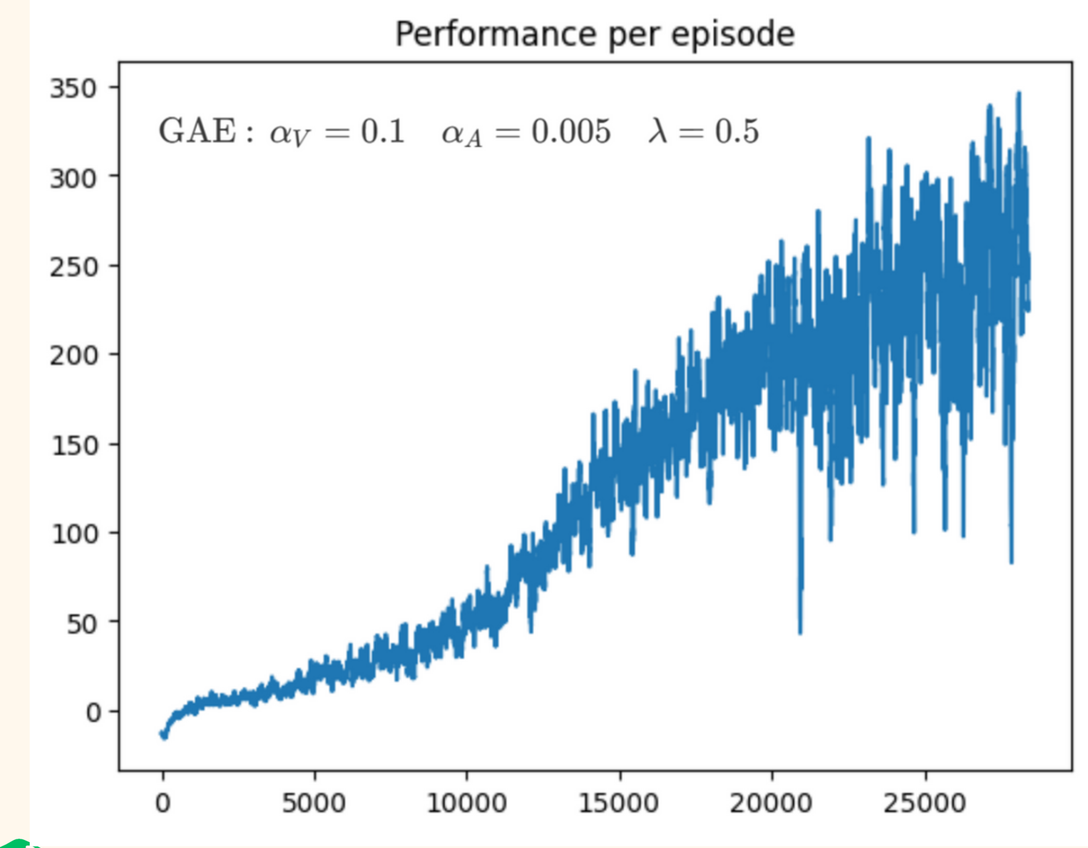

```
food
E
Neighborhood:
[[8 1 8]
 [1 0 0]
 [8 0 8]]
      1.00
0.00        0.00
      0.00
```

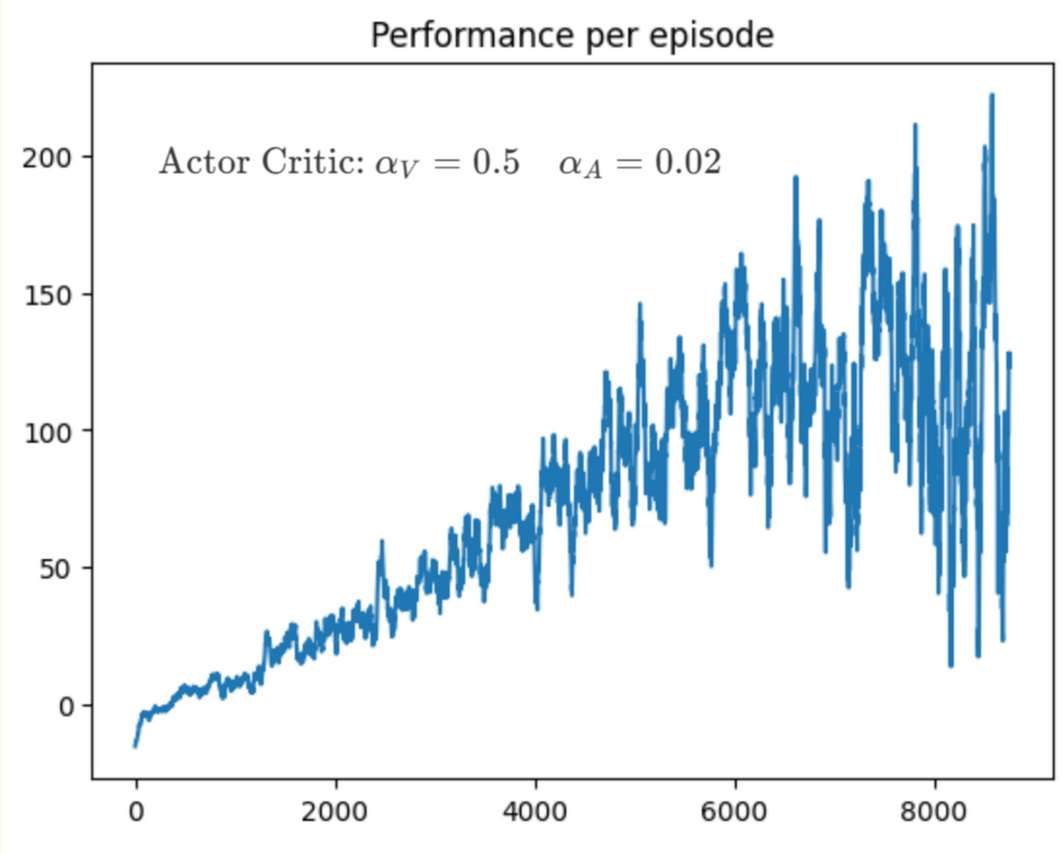
UNIVERSITÀ DEGLI STUDI DI TRIESTE

policy gradient

results

Von Neumann radius 2





UNIVERSITÀ DEGLI STUDI DI TRIESTE

double DQL

why double?

Overoptimism in Q-learning

$$\max_{a \in \mathcal{A}} Q_t(s,a) \geq V^*(s) + \sqrt{\frac{C}{m-1}}$$

Idea: decompose max operation
into action selection and action evaluation
using **two independent** neural networks

$$Y_t^{\mathrm{DDQL}} = R_{t+1} + \gamma Q\left(S_{t+1}, \arg\max_{a \in \mathcal{A}} Q(S_{t+1}, a, \theta_t), \theta_t^-\right)$$

UNIVERSITÀ DEGLI STUDI DI TRIESTE

double DQL
pseudocode

Algorithm 1 Double Deep Q-Learning (DDQL)

Initialize replay memory $\mathcal{D}$, online network $\tilde{Q}$, target network $\hat{Q}$
Loop for each episode:
  Initialize initial state S_1
  For $t = 1, T$ **do**:
    With probability ϵ select a random action A_t
    otherwise select $A_t \leftarrow \arg\max_a \tilde{Q}(S_t, a; \theta)$
    Execute action A_t, observe reward R_t and next state S_{t+1}
    Store transition (S_t, A_t, R_t, S_{t+1}) in $\mathcal{D}$

    Sample a batch $(s_j, a_j, r_j, s_{j+1}) \sim \mathcal{D}$
    For each transition j in the mini-batch **do**:
      if s_{j+1} is terminal:
        $y_j \leftarrow r_j$
      else:
        $a^* \leftarrow \arg\max_{a'} \tilde{Q}(s_{j+1}, a'; \tilde{\theta})$
        $y_j \leftarrow r_j + \gamma \hat{Q}(s_{j+1}, a^*; \hat{\theta})$
      end if
    end for

    Perform a gradient descent step on $(y_j - \tilde{Q}(s_j, a_j; \theta))^2$ w.r.t $\tilde{\theta}$

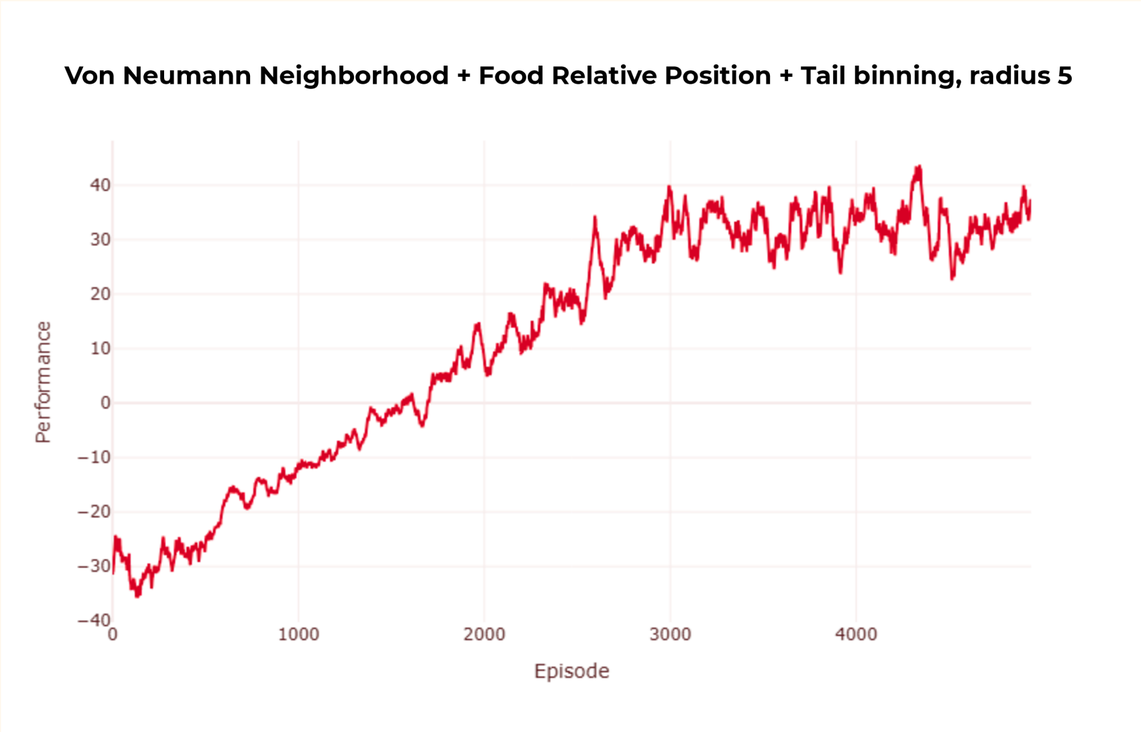
    Every C steps, update the target network: $\hat{\theta} \leftarrow \tilde{\theta}$
  end for
end loop

- **Online** & **Target** Neural Networks
- **Memory buffer**

UNIVERSITÀ DEGLI STUDI DI TRIESTE

double DQL results

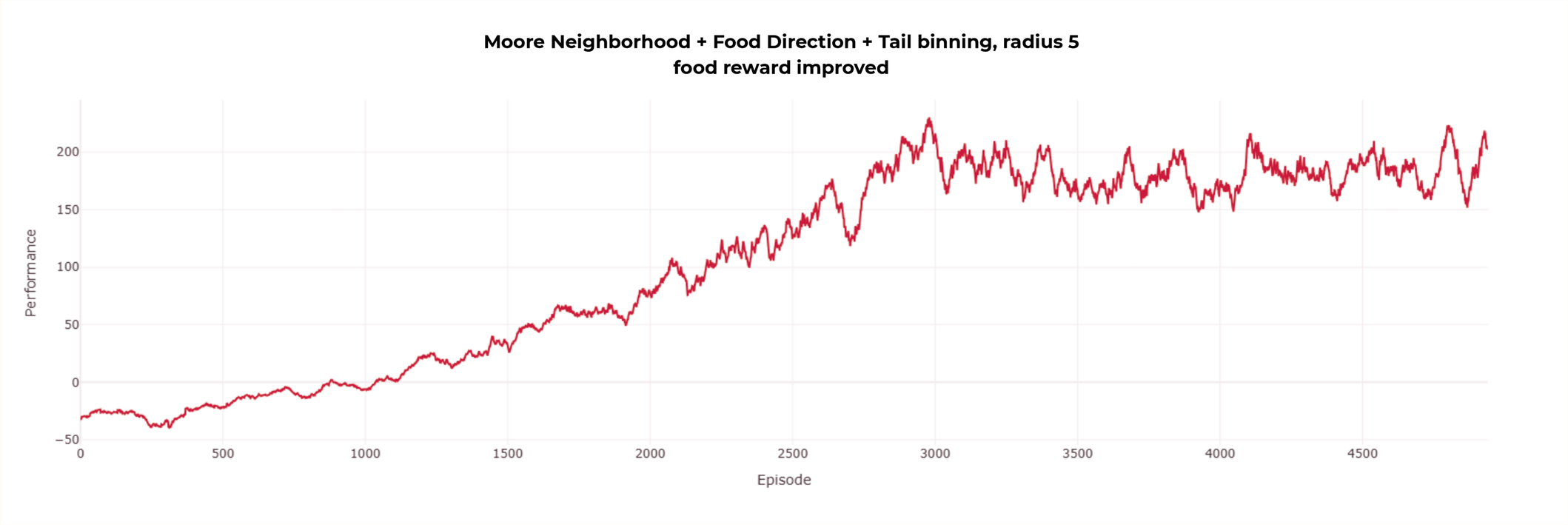
- **Similar results** across the different binnings
- Average food eaten for episode: **~16**
- **Early death** at the beginning: **~15%**



UNIVERSITÀ DEGLI STUDI DI TRIESTE

double DQL results

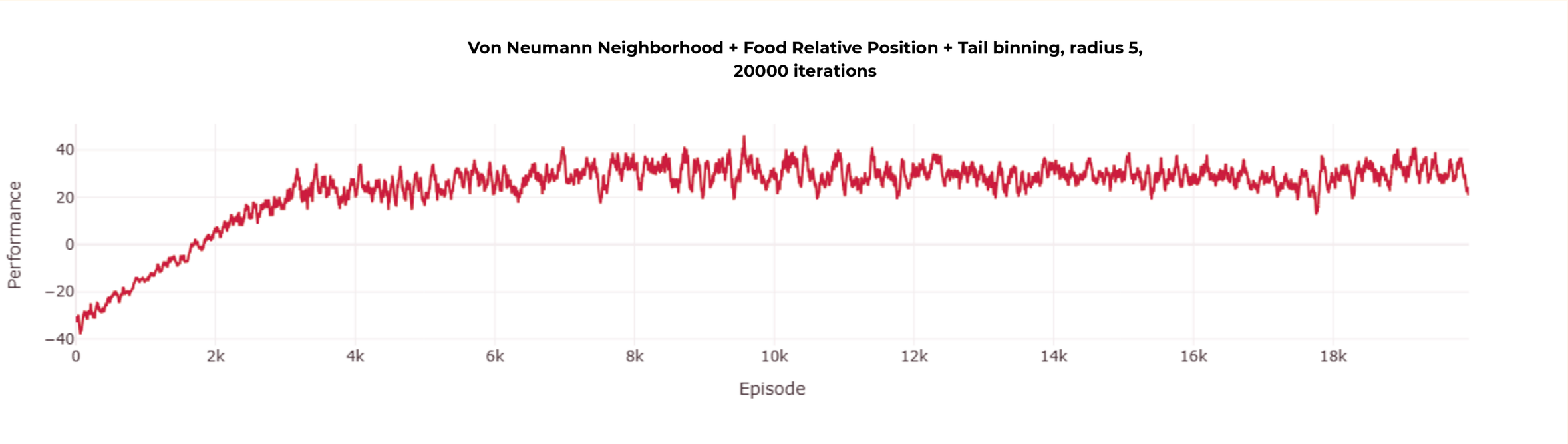
- **Similar results** across the different binnings
- Average food eaten for episode: **~16**
- **Lack of position knowledge** at the beginning leads to **early death**



UNIVERSITÀ DEGLI STUDI DI TRIESTE

double DQL results

- **Similar results** across the different binnings
- Average food eaten for episode: **~16**
- **Lack of position knowledge** at the beginning leads to **early death**



UNIVERSITÀ DEGLI STUDI DI TRIESTE

future ideas

1

Deep General Advantage Estimation

Asynchronous Advantage Actor-Critic (A3C)

2

3

(Deep-) Dynamic Binning

A. Y. 2024 / 2025
Bredariol F., Savorgnan E., Tic R.

THANK YOU