Studying the Flocking Dynamics of Birds from a Multi-Agent Reinforcement Learning Perspective

Reinforcement Learning

Postgraduate Diploma Programme - Term 2

QLS, ICTP

June 30, 2025

Why do we study birds?

- Historical and scientific interest:
  Early rule-based models

- To understand collective behavior:
  Local interactions lead to emerging global behavior

- Inspiration for intelligent systems:
  Robotics and swarm AI

- Designing scalable algorithms:
  Decentralized control

… and maybe also because they are just beautiful?

"…At sunset we can see starlings forming fantastic patterns, thousands of milling black specks against the changing colors of the sky. We witness them all moving in unison, without touching or dispersing- avoiding obstacles, moving apart and coming together again, continuously reconfiguring their spatial arrangement, almost as if an orchestra conductor were giving order to everything that is happening.

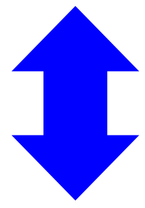
We could watch these birds endlessly, because the show they put on is perpetually altered and renewed, taking different and unexpected forms. "

Giorgio Parisi,
In a Flight of Starlings

Reinforcement learning:
A natural framework for studying bird flocking

RL Framework

- Agents interact within a complex and dynamic environment
- Receive rewards based on their actions
- Learn and adapt behavior over time



Living Systems (e.g. bird flocks)

- Birds interact with neighbors and respond to environmental cues
- Feedback through survival, alignment, or cohesion
- Continuous behavioral adaptations to maintain survival

MARL components

1- Environment

Square field with periodic boundary conditions

2- Agents

Agents are birds - each is initialized with:

i- a random position $(x, y) \in [-L, L] \times [-L, L]$ updated at each time step

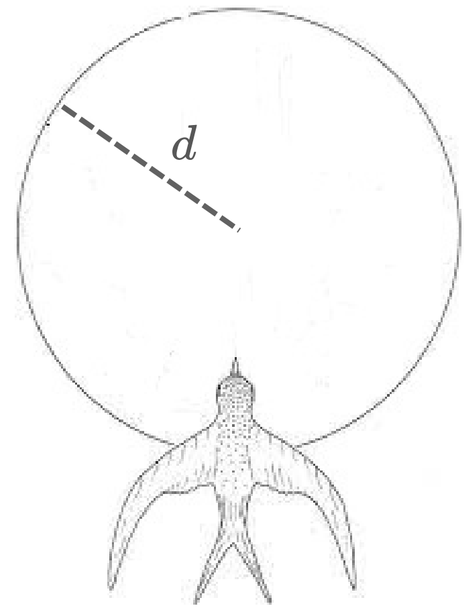
ii- a random orientation, θ, sampled continuously from $[0, 2\pi]$ updated upon taking an action

$\longrightarrow$ The position update at each time step is given by: $\vec{r}_{t+1} = \vec{r}_t + \vec{v}$, with $\vec{v} = 1 \cdot \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$

3- Observation Space

- The observation space ($\mathcal{O}$) needs to be finite
- Define an observational radius d within which a bird can detect neighbors

$\Rightarrow$ Birds receive local information only: partially observable setting



4- Action Space

The action space ($\mathcal{A}$) is very small constituting of two actions only:

1) Instinctive (I): Birds follow their instinct and fly in one of the 4 cardinal direction [leaders]

2) Vicsek-like (V): Birds align themselves along the average flight direction of its neighbors [followers]

Note:

To keep things finite, the visited directions are discretized.
After performing an action, birds find themselves in one of 8 directions

Index i	**Angle** θ_i	**Direction**	**Unit Vector**
0	$-\pi$ or π	West	$[-1, 0]$
1	$-3\pi/4$	Northwest	$[-0.71, -0.71]$
2	$-\pi/2$	North	$[0, -1]$
3	$-\pi/4$	Northeast	$[0.71, -0.71]$
4	0	East	$[1, 0]$
5	$\pi/4$	Southeast	$[0.71, 0.71]$
6	$\pi/2$	South	$[0, 1]$
7	$3\pi/4$	Southwest	$[-0.71, 0.71]$

5- State Space

- The state space $(\mathcal{S})$ is large

- There are no other objects so the state of an environment is determined by the N directions of the birds

- Along each direction, we have three possibilities:
  (i) No neighbors
  (ii) One neighbor
  (iii) Two or more neighbors

→ Total number of states = $3^8 = 6561$

6- Reward System

- The reward is based on flying in a predefined preferred direction
  (this is what allows us to test if collective behavior emerges as a result of learning the best policy)

- Introduce collision cost/ penalty

$Q-$ Learning Algorithm

- Widely known and generally applicable algorithm
- Q is initialized randomly
- The agent interacts with the environment and updates its Q-values:

$$Q_t(s_t, a_t) = (1-\alpha)Q_{t-1}(s_t, a_t) + \alpha\left[r_t + \gamma \max_{a \in \mathcal{A}} Q_{t-1}(s_{t+1}, a)\right] \qquad \text{where} \qquad \begin{matrix} \alpha \in [0,1] \\ \gamma \in [0,1) \end{matrix}$$

$\varepsilon-$ Greedy policy for $Q-$ Learning

$$\pi(a \mid s) = \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{|A|} & \text{if } Q(s,a) = \max\limits_{a' \in A} Q(s,a') \\ \dfrac{\varepsilon}{|A|} & \text{otherwise} \end{cases}$$

Final comments

- Each bird maintains its own Q-table
- Each Q-table is updated via local interactions
- Agents learn concurrently and independently
- Agents do not have personalized goals → Their goal is shared

→ This allows us to check whether cooperation emerges from individual learning

Initialization

- Initialize Q-tables Q_i for each bird i
- Initialize bird positions and directions (randomly)
- Assign leaders and followers
- Set parameters: $N, \alpha, \gamma, \varepsilon$, etc...

Observation, Action Selection, and Q-learning

1: **for** each bird i **do**
2: Observe neighbors within radius d
3: Encode local state s_i
4: Select action a_i using ε-greedy policy:

- With probability $1 - \varepsilon$: choose $a_i = \arg\max_a Q_i(s_i, a)$
- With probability ε: pick random $a_i \in A$

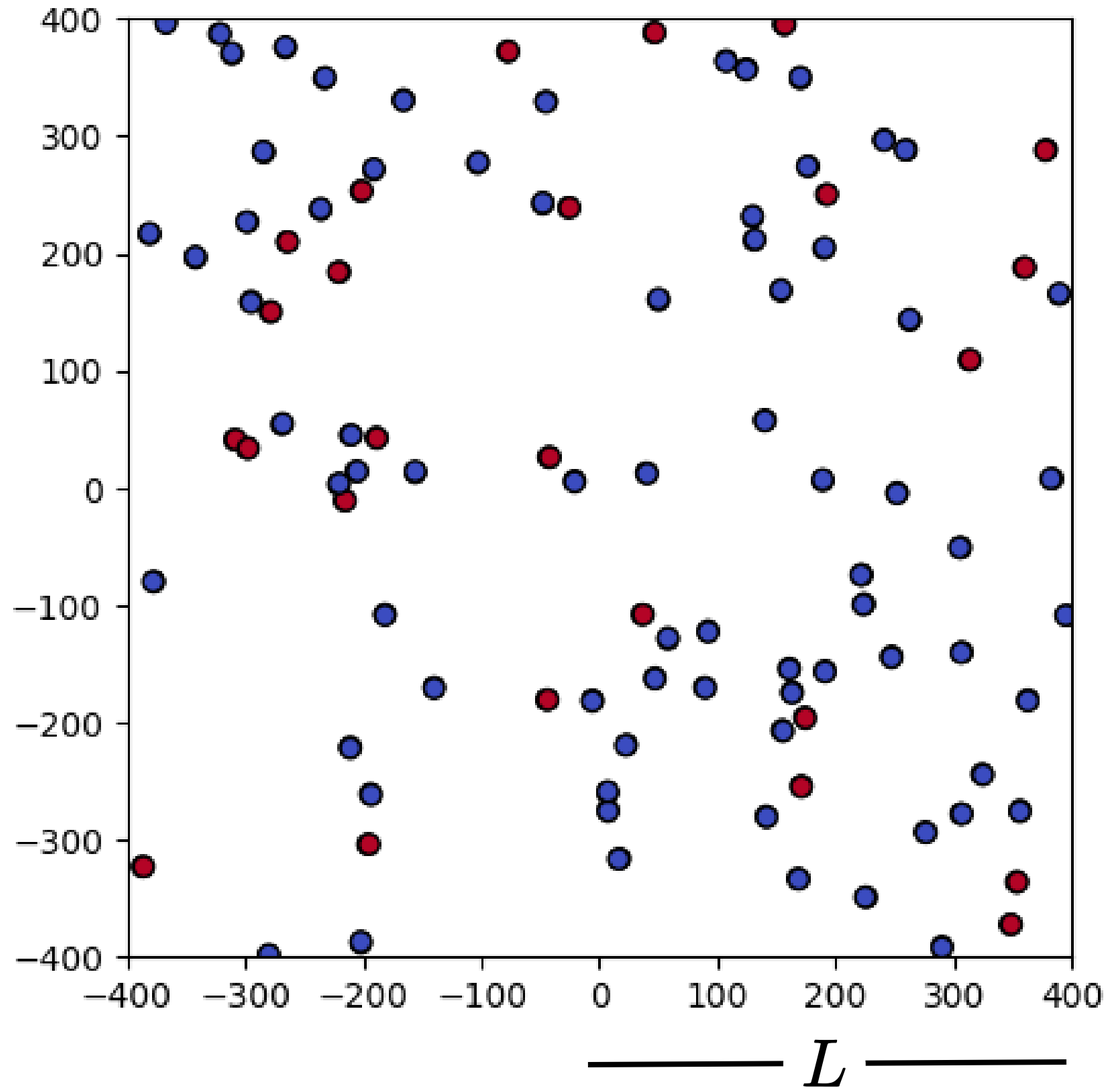
5: Execute action a_i and observe reward r_i and new state s_i'
6: Update Q-value:

$$Q_i(s_i, a_i) \leftarrow Q_i(s_i, a_i) + \alpha \left[r_i + \gamma \max_{a'} Q_i(s_i', a') - Q_i(s_i, a_i) \right]$$

7: **end for**

Fixed simulation parameters

Parameter	Value
$L :=$ Field length	400
$N :=$ Number of birds	100
$d :=$ Observation radius	100
$r :=$ Collision radius	10
$\frac{l}{f} :=$ Leader-to-follower ratio	0.25
$\alpha :=$ Learning rate	0.1
$\gamma :=$ Discount factor	0.9
$\varepsilon :=$ Exploration parameter (ε-greedy)	0.5



Results

1. Cost vs No cost:
   Exploring the effect of penalty on Q-learning

2. Varying the learning parameters

Policy Divergence Metric

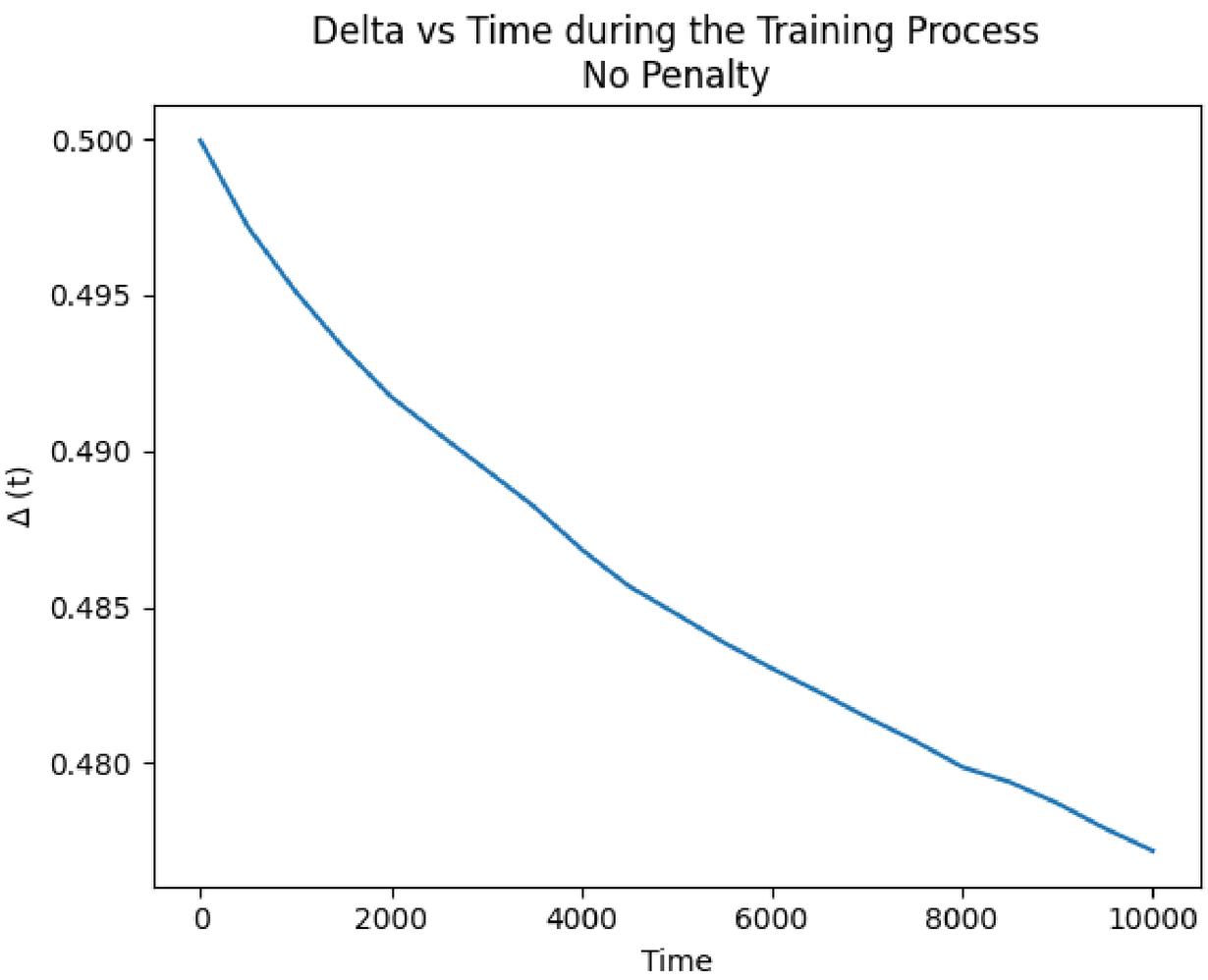
In words, it is the normalized distance between the optimal policy and the learnt policy

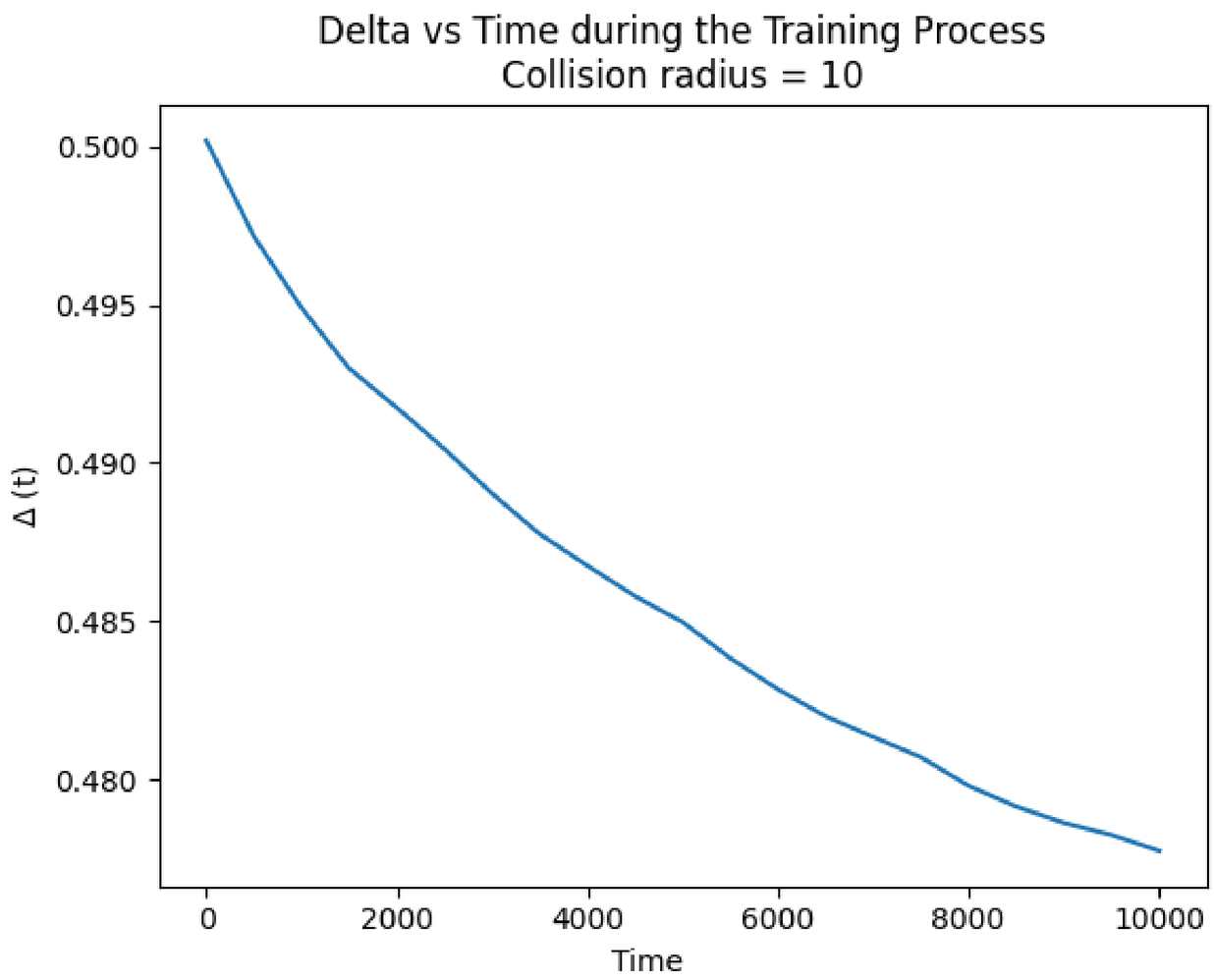
$$\delta_i^{\mathcal{L}}(o) = \begin{cases} 0 & \text{if } Q^i(o,l) > Q^i(o,V) \\ 1 & \text{if } Q^i(o,l) \leq Q^i(o,V) \end{cases}$$

$$\delta_i^{\mathcal{F}}(o) = \begin{cases} 0 & \text{if } Q^i(o,V) > Q^i(o,l) \\ 1 & \text{if } Q^i(o,V) \leq Q^i(o,l) \end{cases}$$

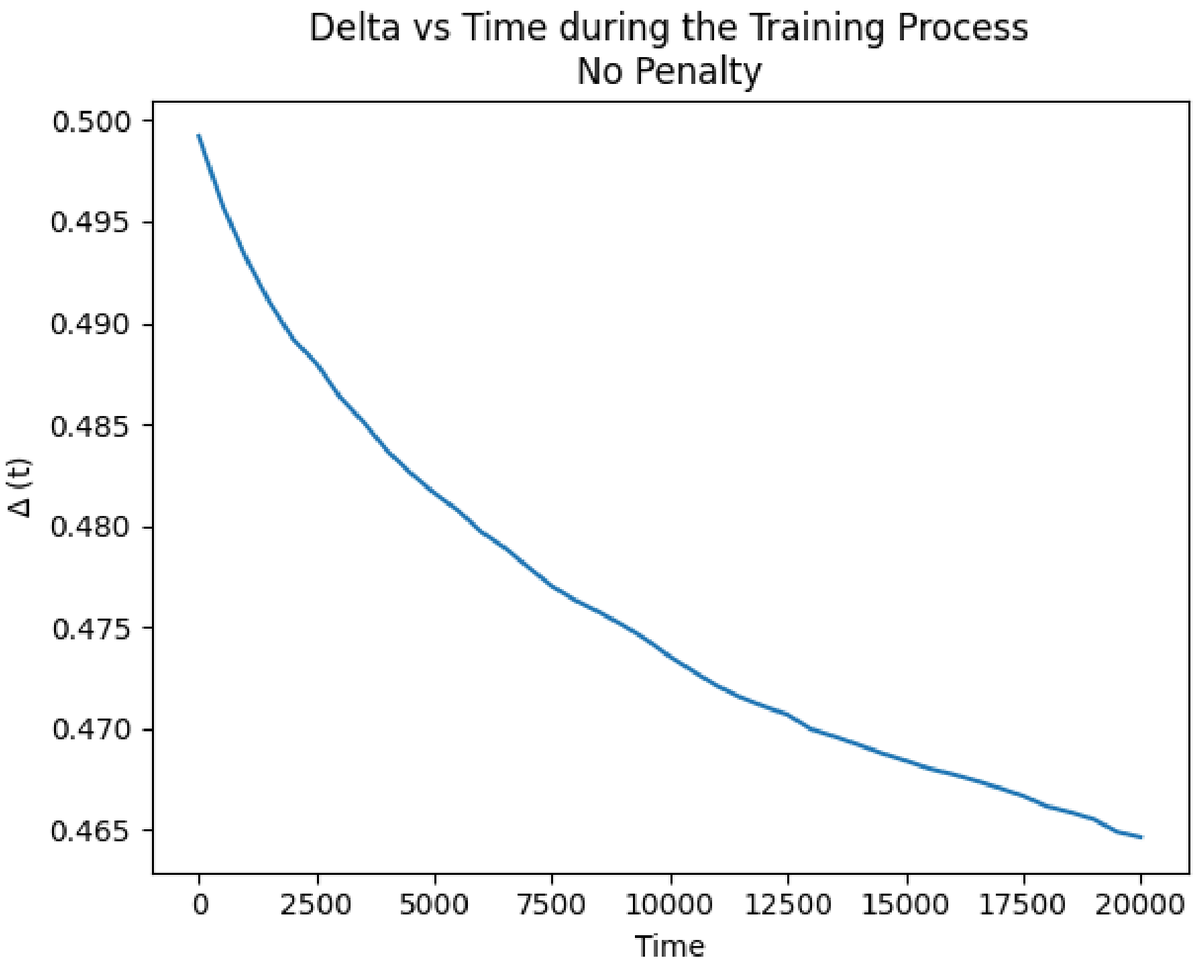
$$\Delta = \frac{1}{N|\mathcal{O}|} \sum_{o \in \mathcal{O}} \left(\sum_{i \in \mathcal{L}} \delta_i^{\mathcal{L}}(o) + \sum_{i \in \mathcal{F}} \delta_i^{\mathcal{F}}(o) \right)$$

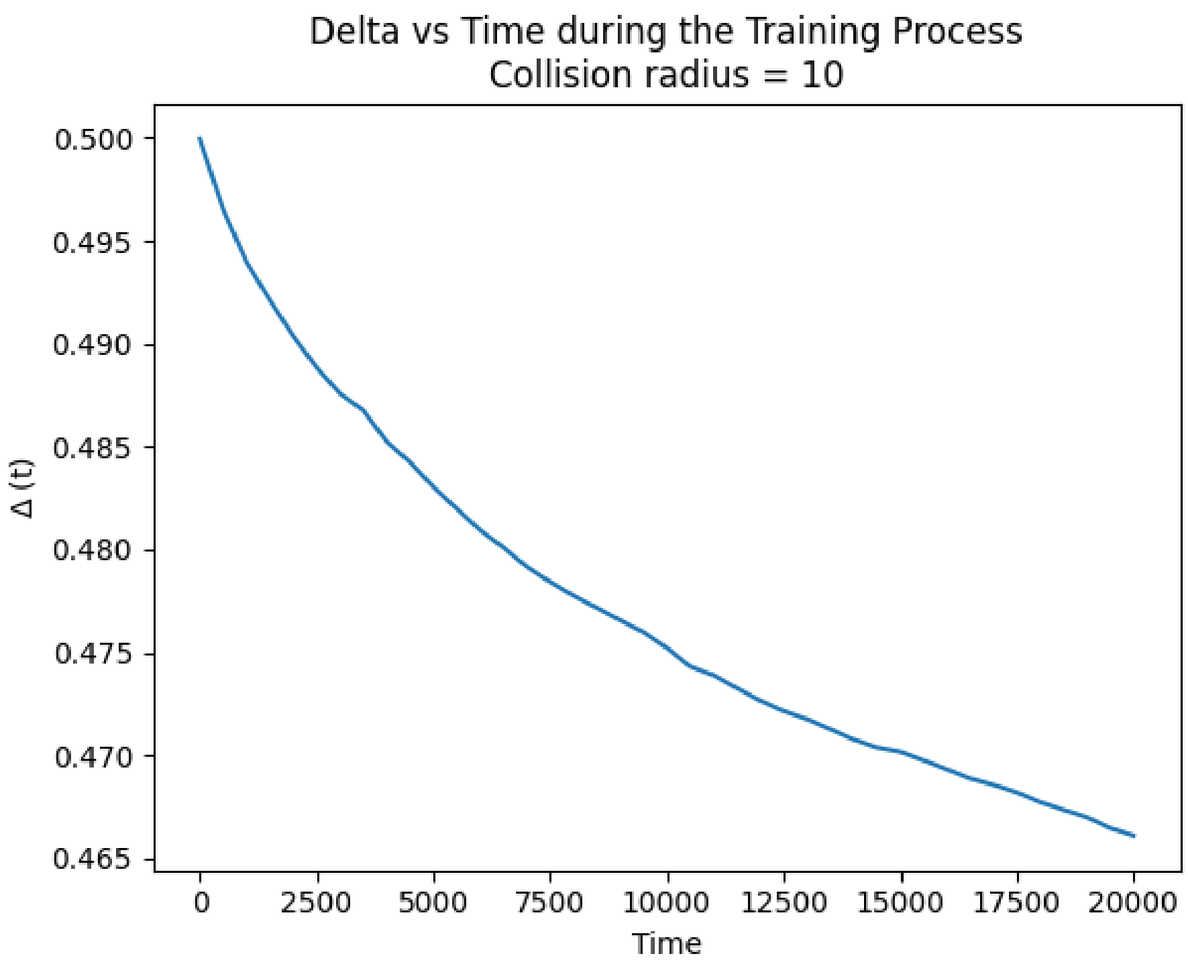
Post-training evaluation: $t = 10000$





Post-training evaluation: $t = 20000$





Findings

- In both cases, Δ decreases indicating that emergent learning is occurring

- Collision penalty has a mild effect on the learning as both reach the same value at the end of training for both training times

- However, penalty does make the delta less smooth possibly reflecting an unstable learning dynamics

Why adding a collision cost may have no effect?

- Penalized states are rare

- Penalty may be too small to counteract the benefit of flocking.

- Robust Q-learning: policy updates remain driven by alignment rewards more than occasional penalties.

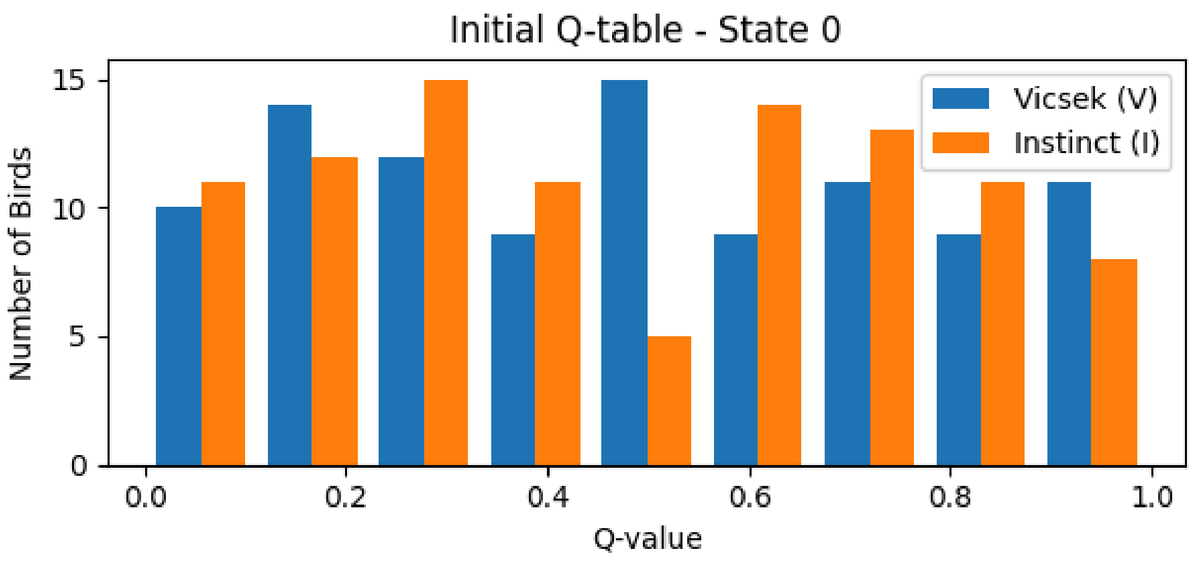
Why is Δ less smooth with penalty?

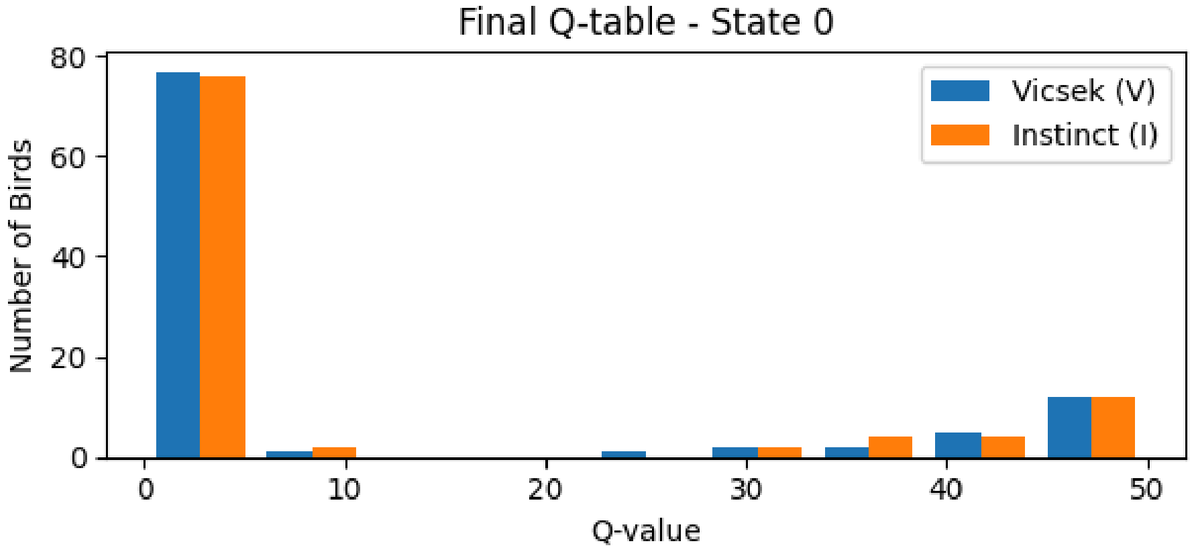
- Reward becomes dependent on the local configuration (number of neighbors and distance from them)

- Noisy reward $\rightarrow$ unstable Q-updates/ slow convergence

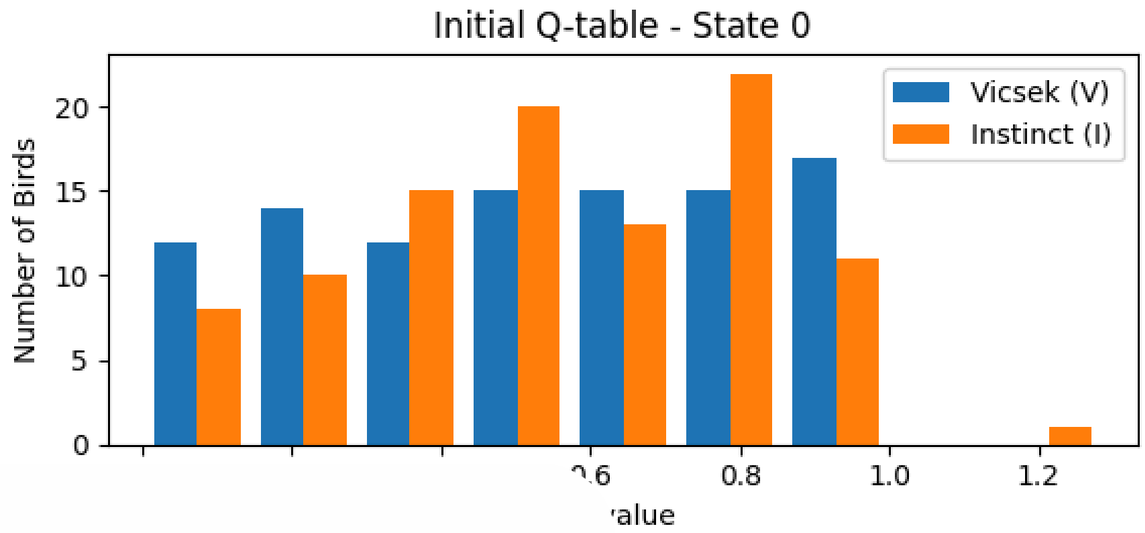
Q-value Distributions (Initial vs Final) at a Given State

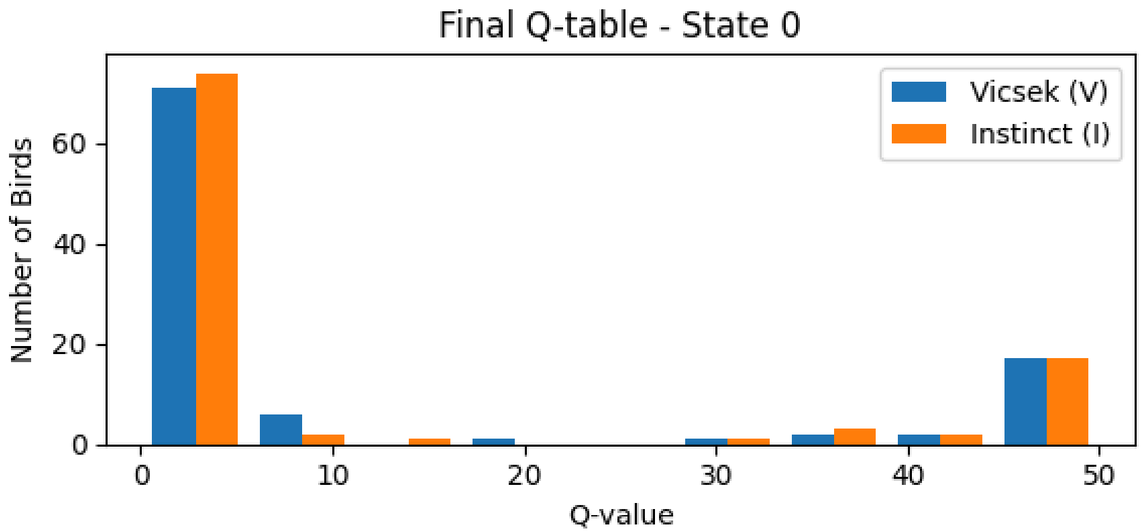
Without cost





With cost





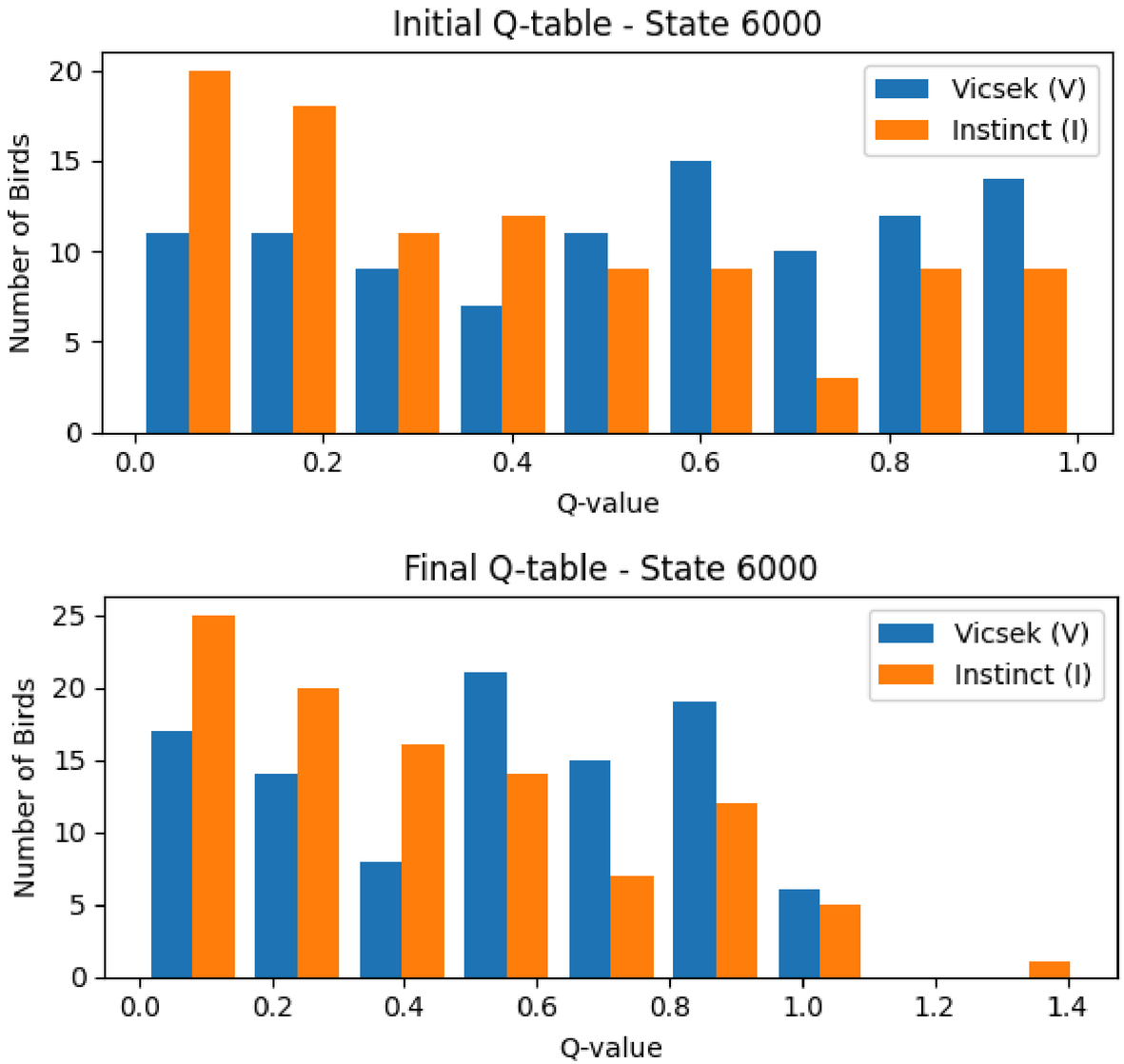
Findings

- State 0 behaves unusually → Could be due to initialization bias or early visit frequency

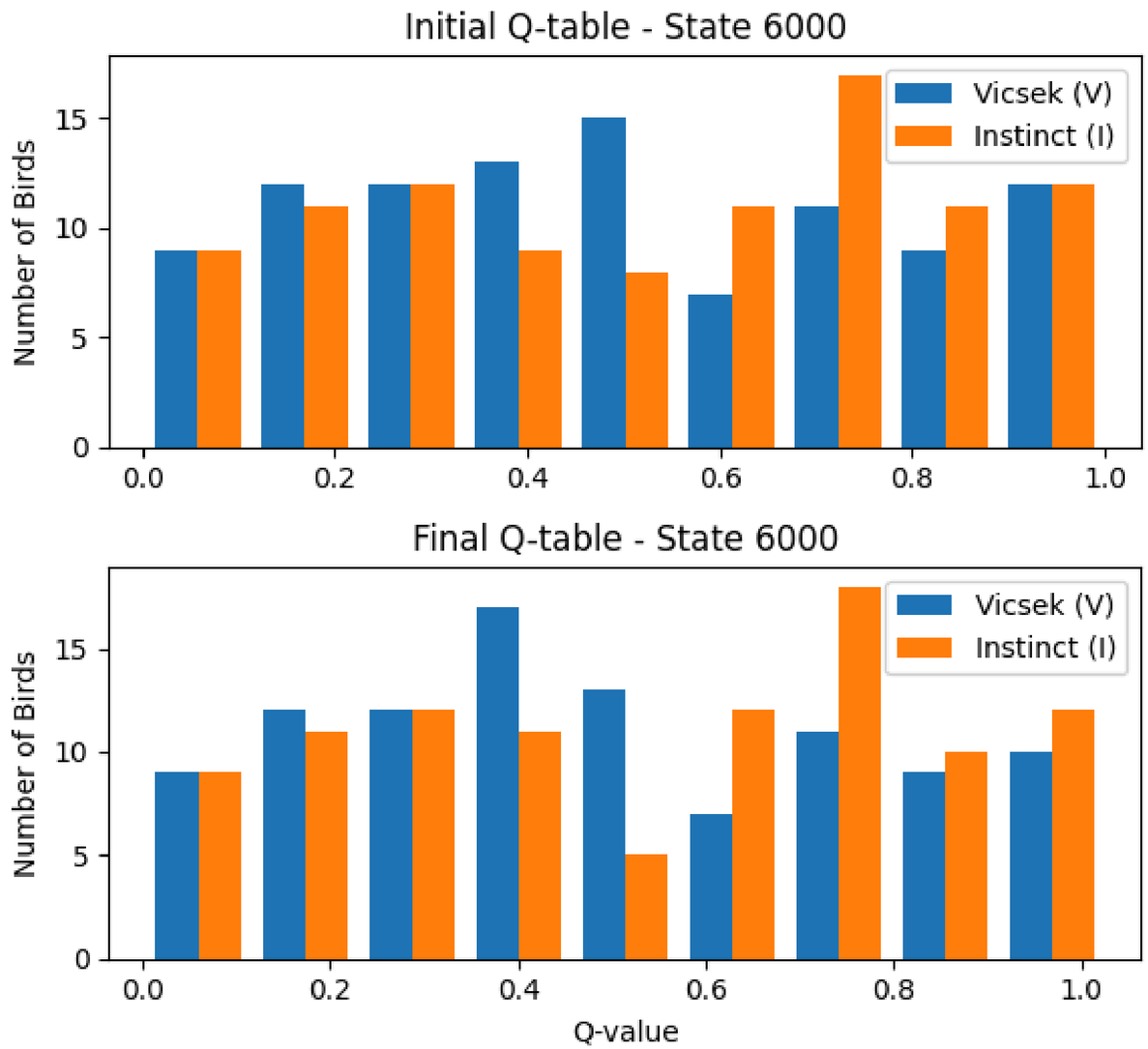
- The distribution shape is nearly identical for both cases

Q-value Distributions (Initial vs Final) at a Given State

Without cost



With cost



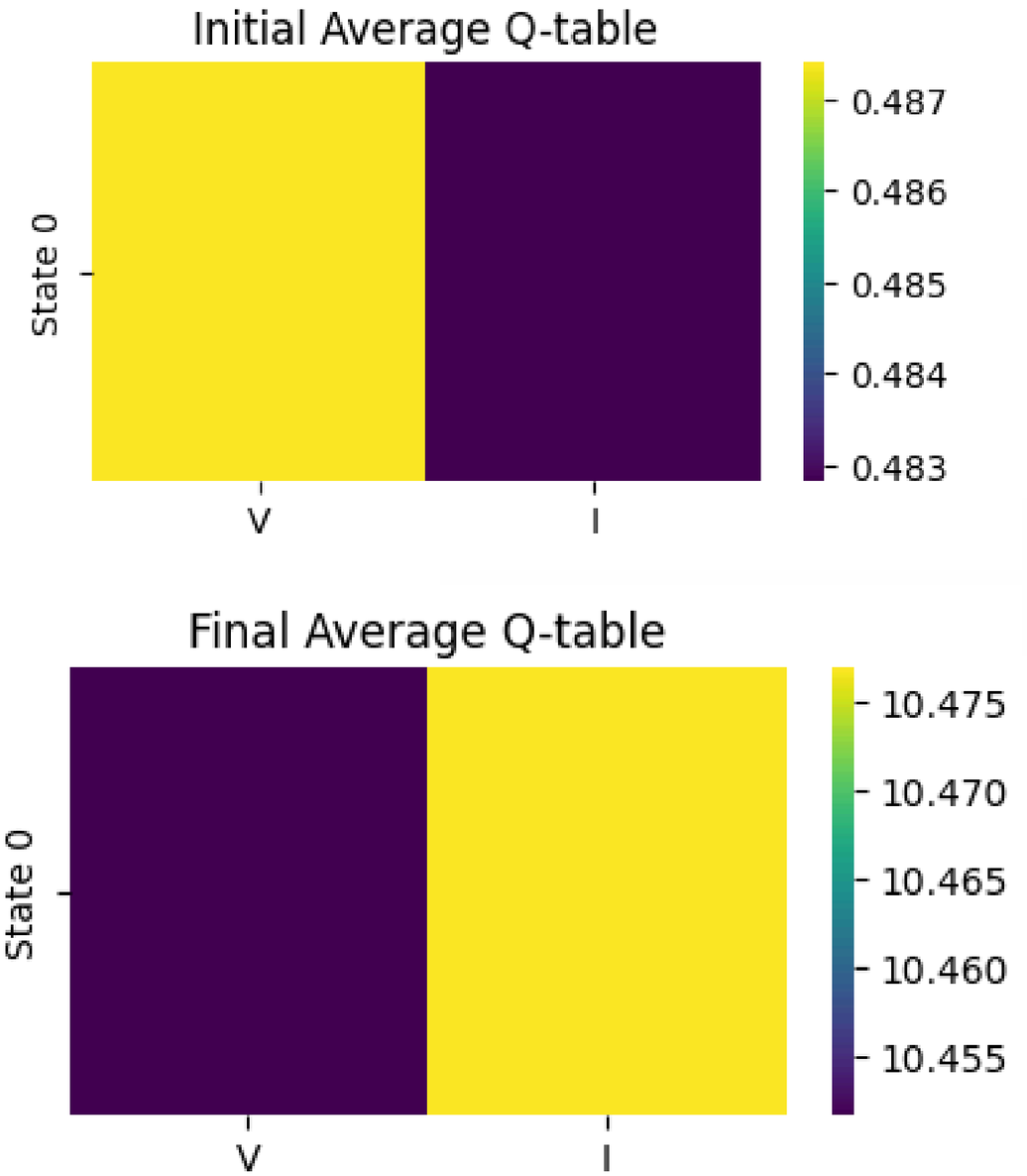
Findings

- Some Q-values are lower in the presence of a cost

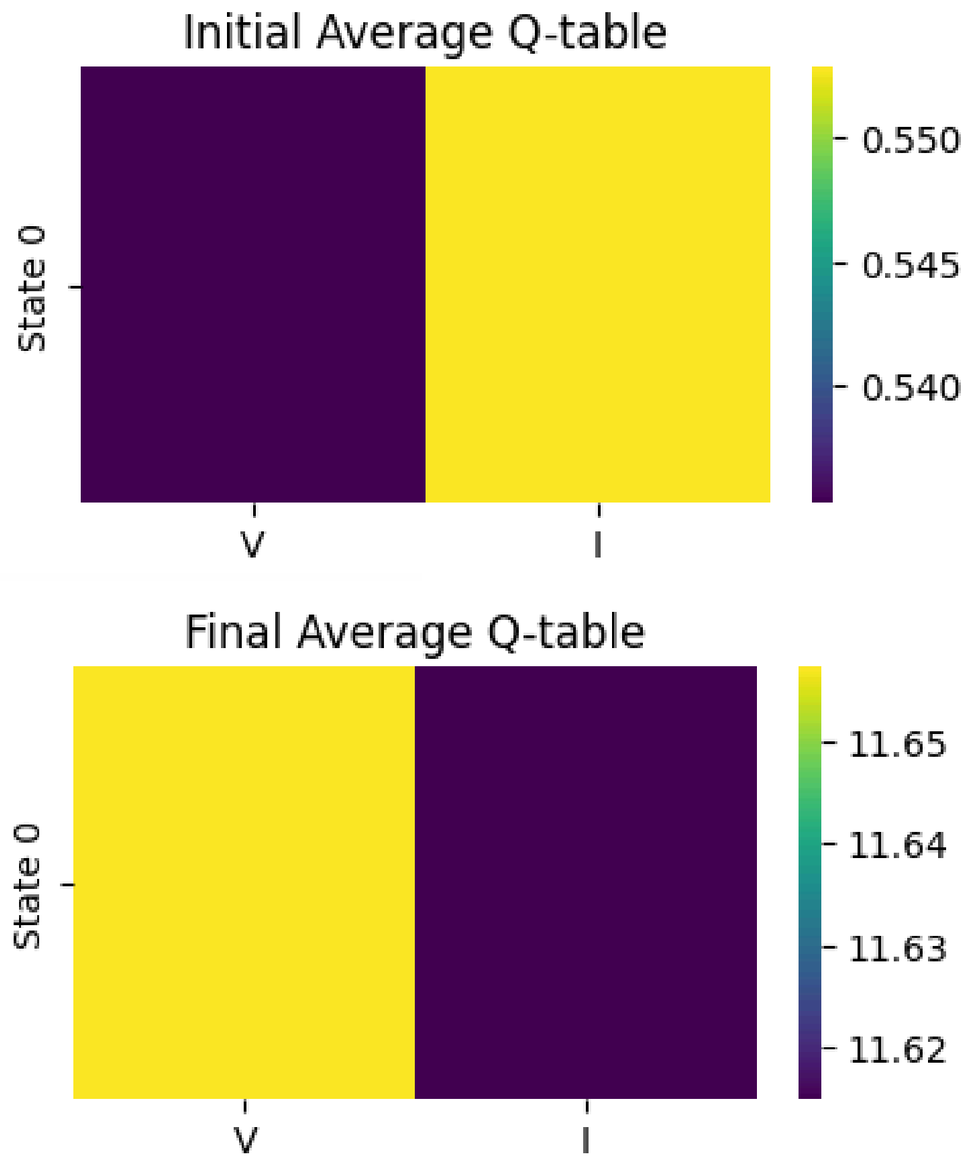
- Weaker separation between V and I policies:
  Birds don't develop strong preferences

Average Q-values for Selected States (Initial vs Final)

Without cost

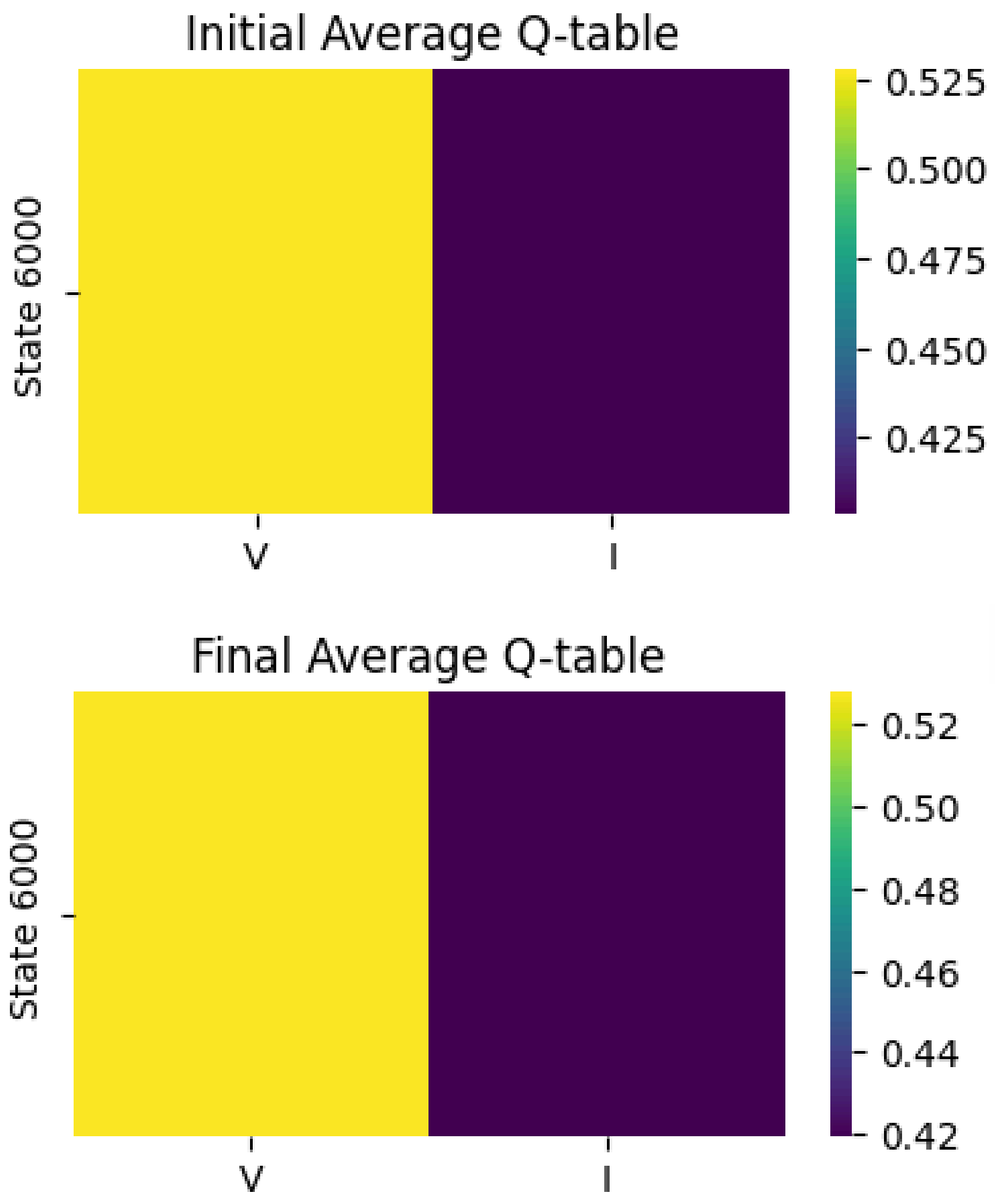


With cost

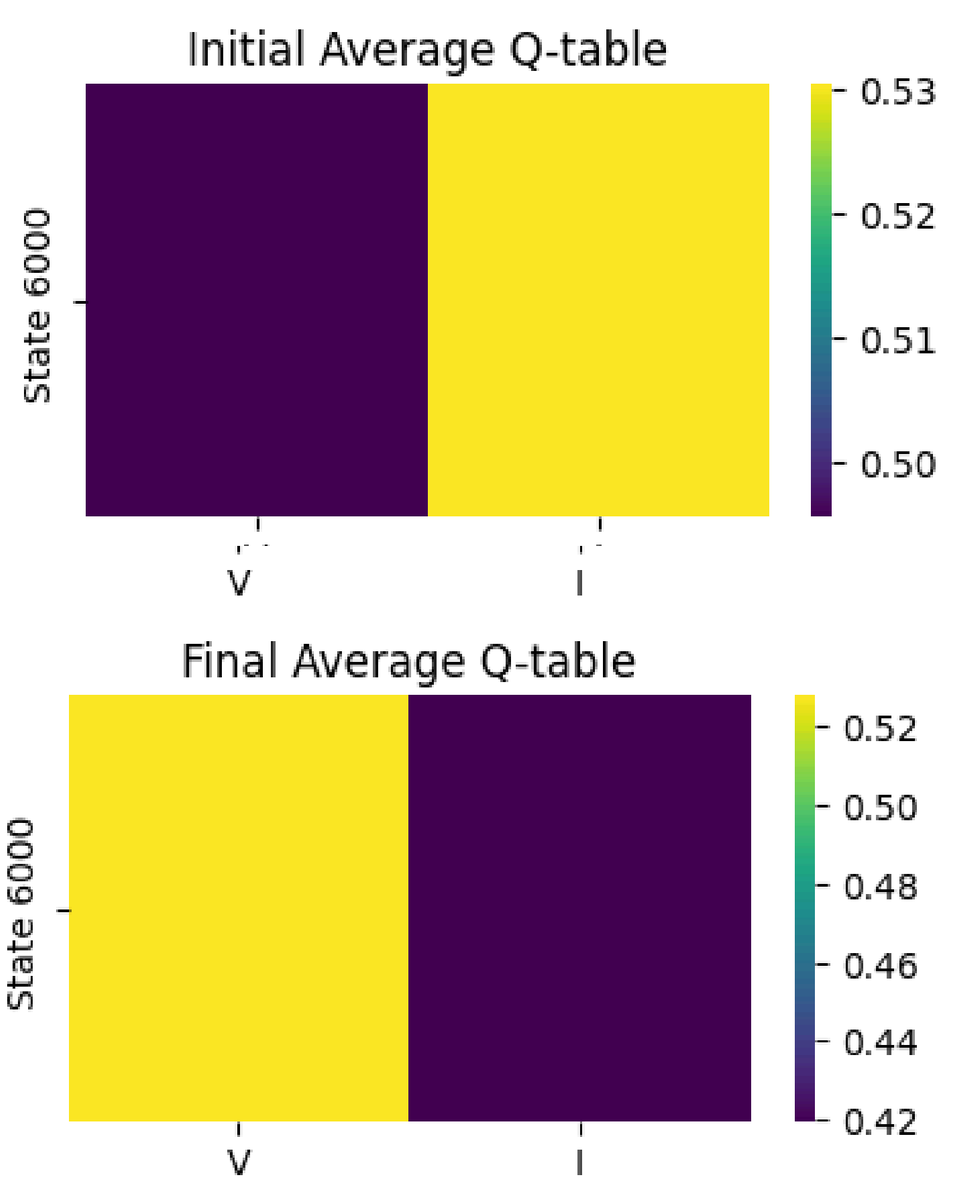


Average Q-values for Selected States (Initial vs Final)

Without cost

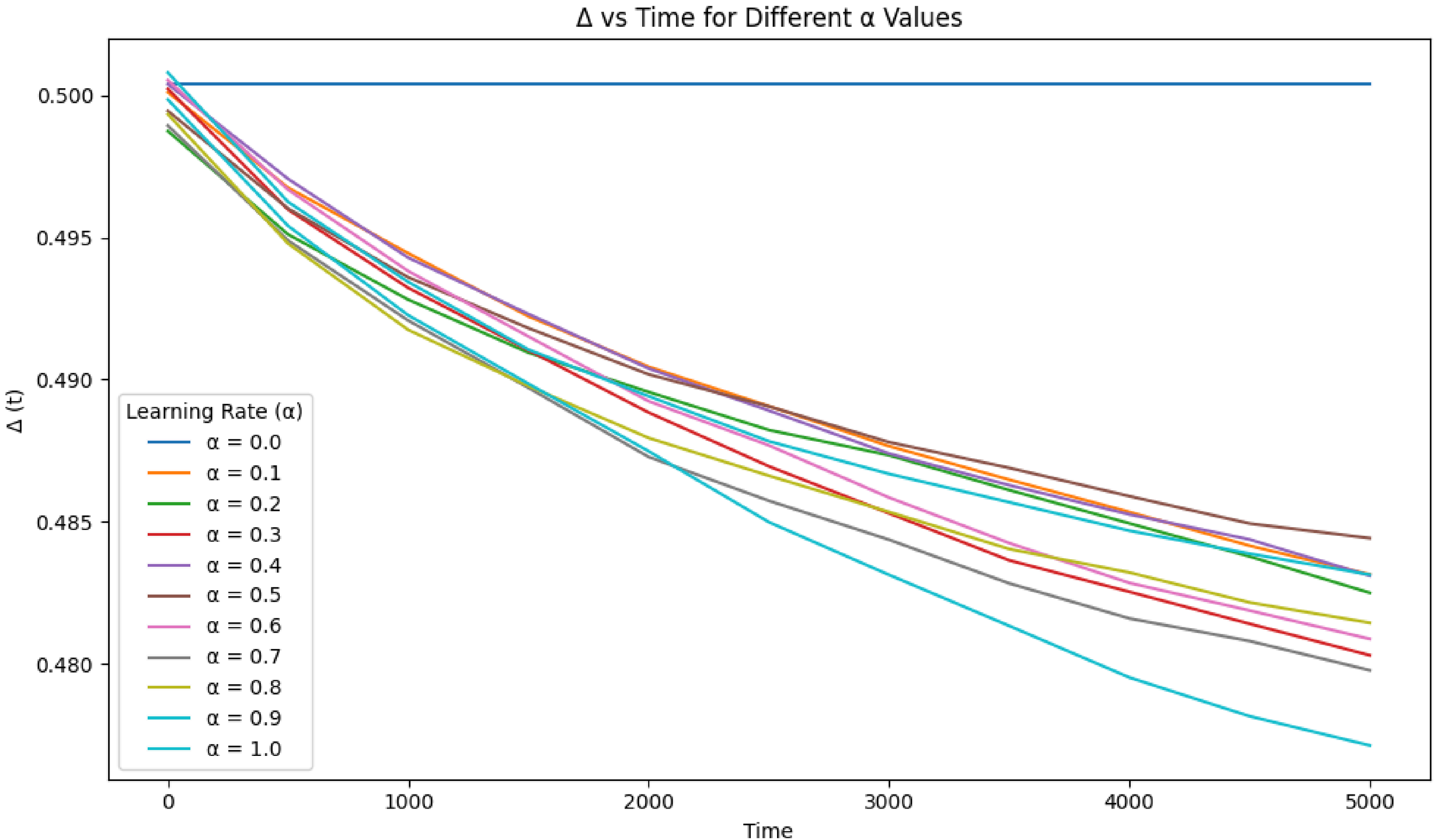


With cost

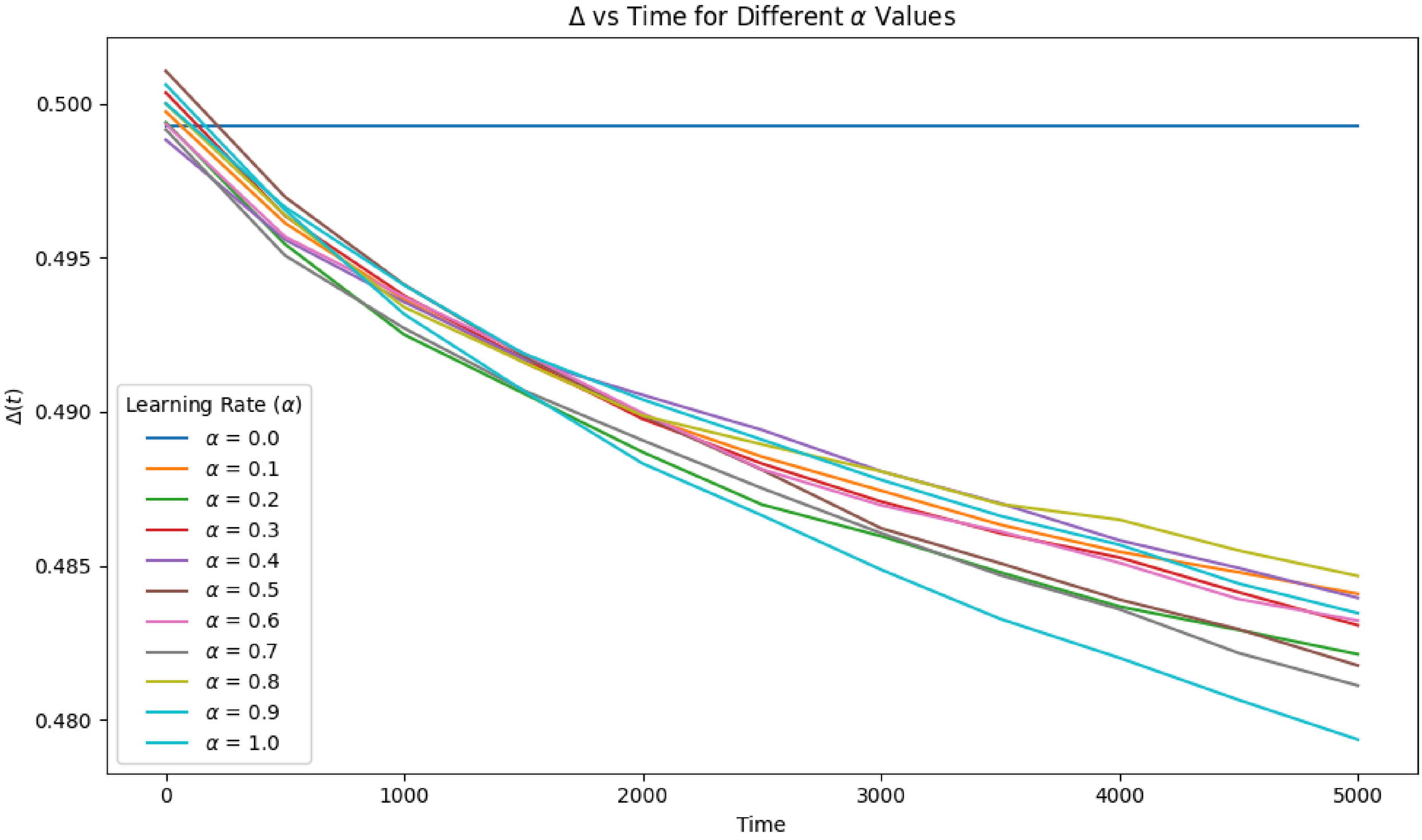


Varying the Learning Parameters

Without cost



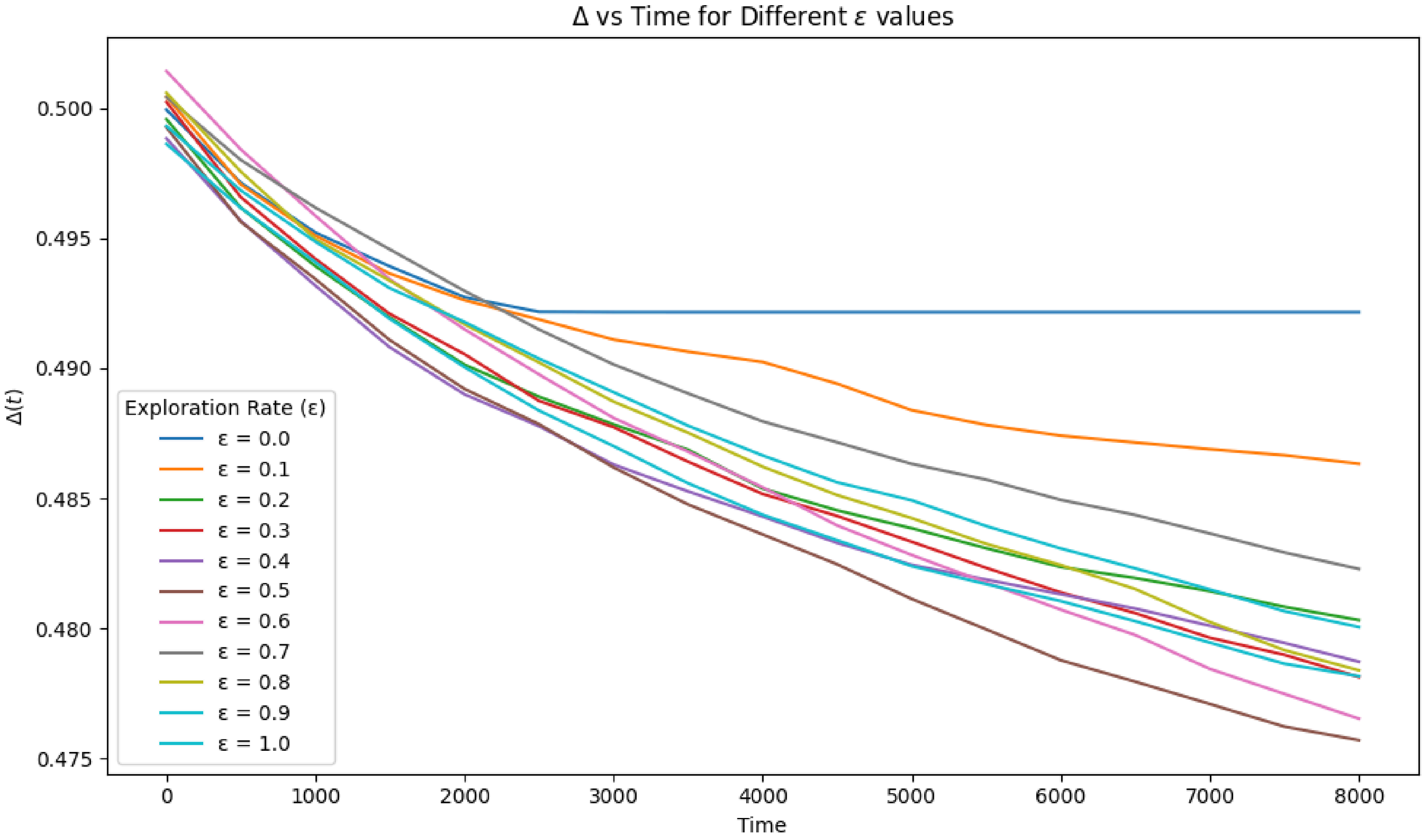
With cost


Δ vs Time for Different α Values
Δ(t)
Time
0.500
0.495
0.490
0.485
0.480
0
1000
2000
3000
4000
5000
Learning Rate (α)
α = 0.0
α = 0.1
α = 0.2
α = 0.3
α = 0.4
α = 0.5
α = 0.6
α = 0.7
α = 0.8
α = 0.9
α = 1.0

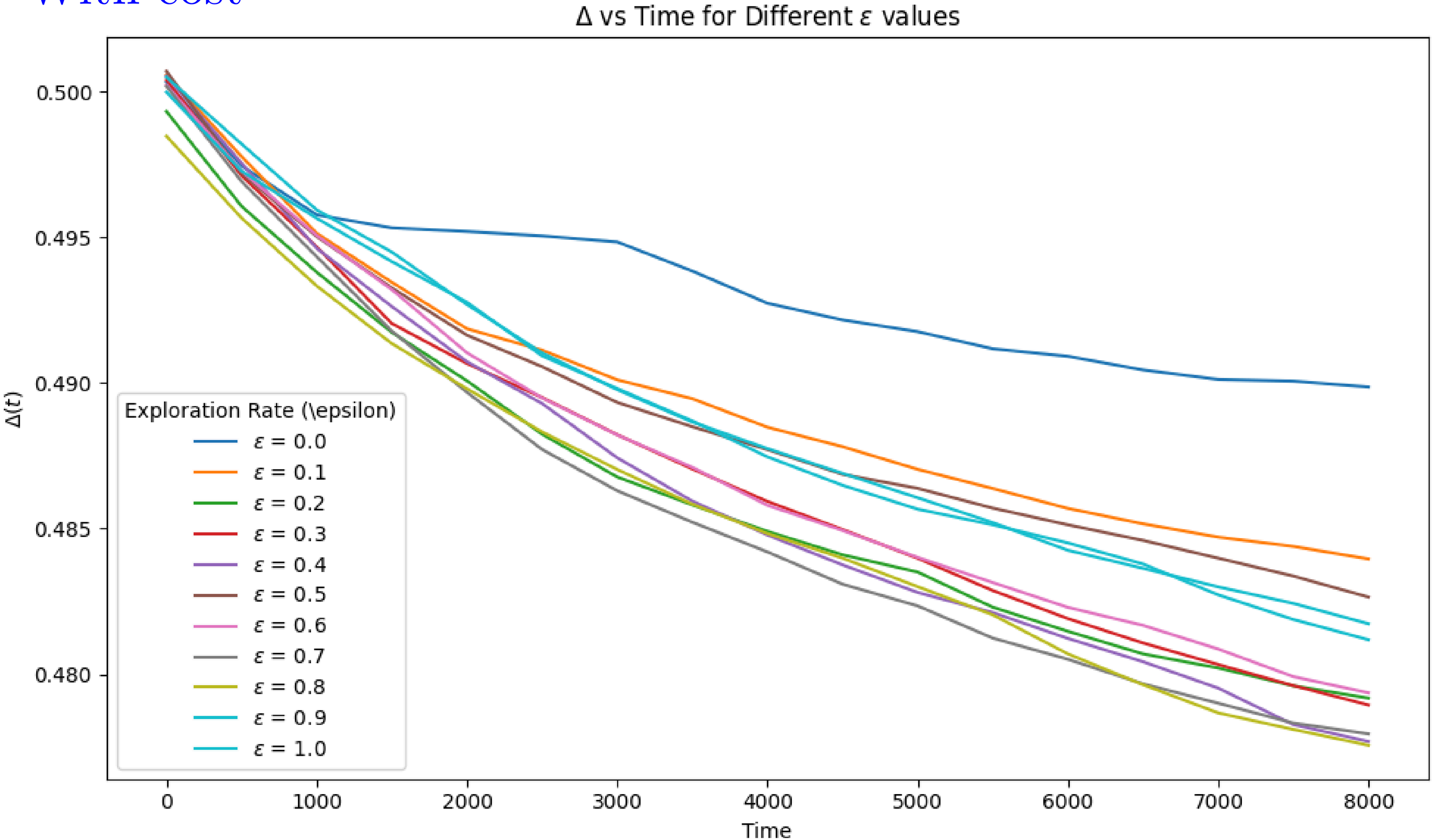
Findings

No Penalty	With Penalty
Sharper drop in slope	Smoother drop
More separated curves	Curves more compact
Best convergence for $\alpha = 1$	Converges but not dominant for $\alpha = 1$

Without cost



With cost


Δ vs Time for Different ε values
0.500
0.495
0.490
0.485
0.480
Δ(t)
0
1000
2000
3000
4000
5000
6000
7000
8000
Time
Exploration Rate (\epsilon)
ε = 0.0
ε = 0.1
ε = 0.2
ε = 0.3
ε = 0.4
ε = 0.5
ε = 0.6
ε = 0.7
ε = 0.8
ε = 0.9
ε = 1.0

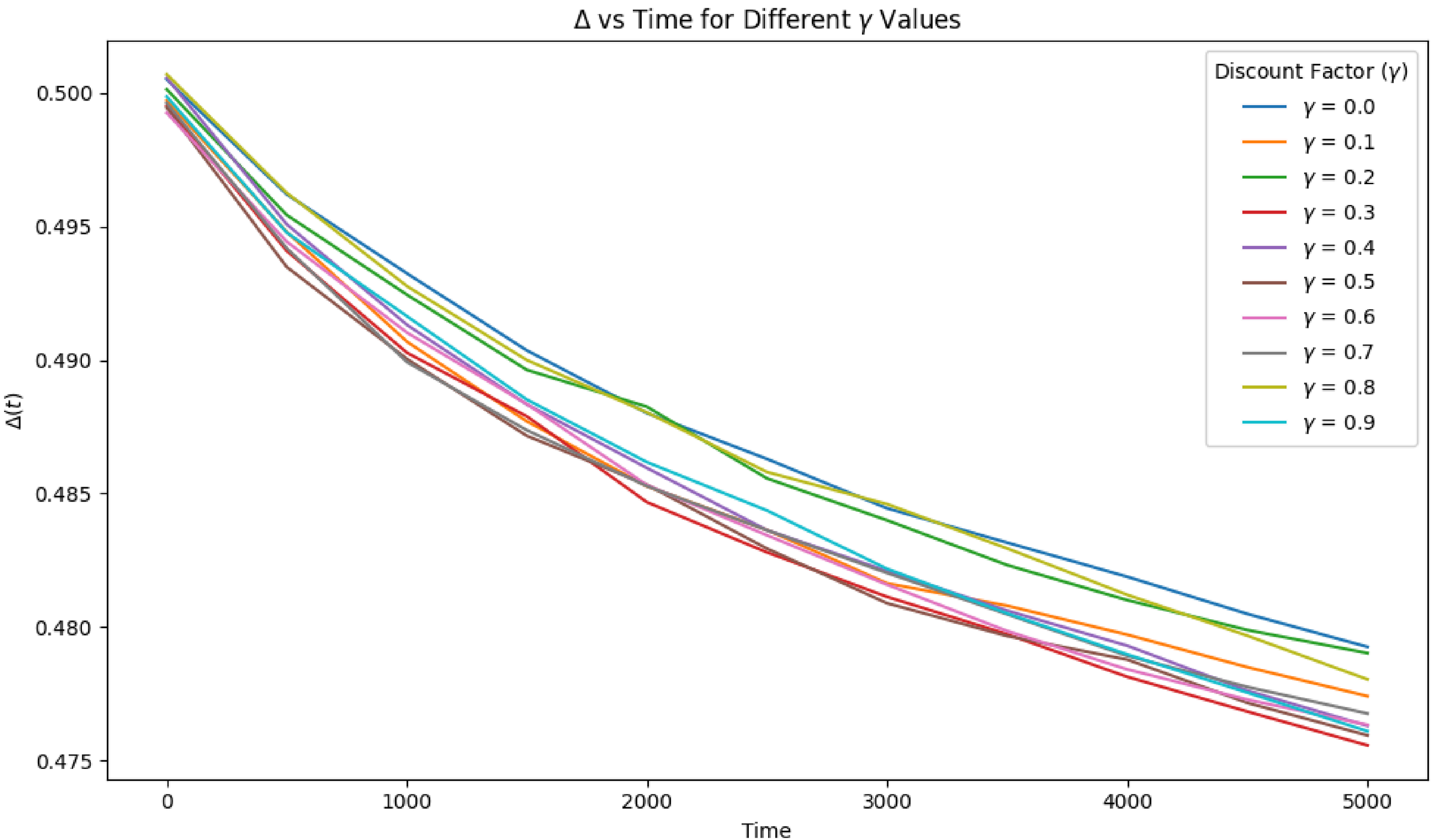
Findings

- For $\varepsilon = 0$, we would expect no delta evolution; however in both cases there is some drop before it is flat again

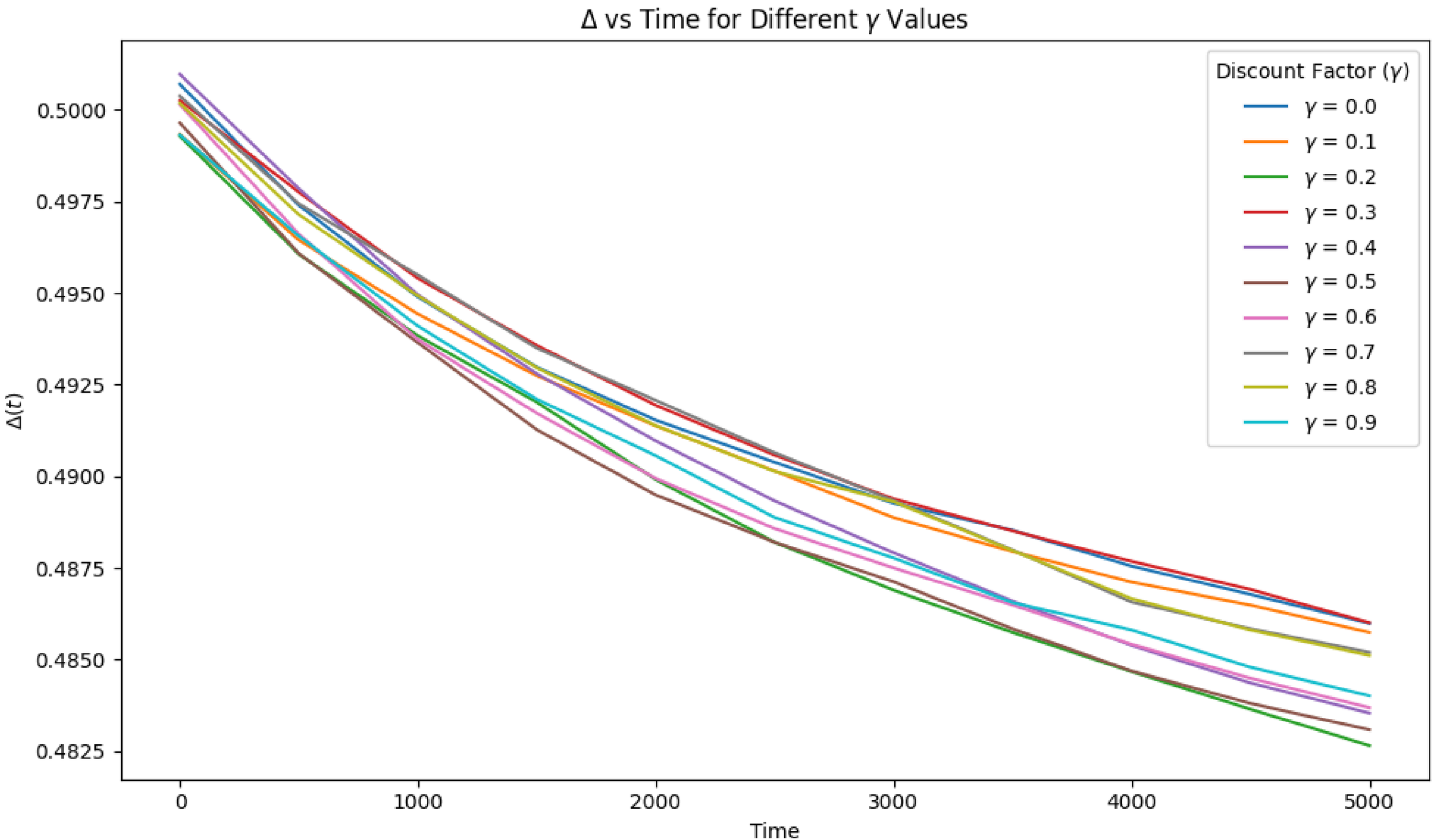
- In both cases, high values lead to slow or poor convergence

Feature	**Observation**
$\varepsilon = 0.0$	Flat or very slow decay (no exploration)
$\varepsilon = 0.1$–0.2	Still slow convergence
$\varepsilon = 0.3$–0.8	Mid-values exhibit faster decay

Without cost



With cost


Δ vs Time for Different γ Values
Discount Factor (γ)
γ = 0.0
γ = 0.1
γ = 0.2
γ = 0.3
γ = 0.4
γ = 0.5
γ = 0.6
γ = 0.7
γ = 0.8
γ = 0.9
Δ(t)
0.5000
0.4975
0.4950
0.4925
0.4900
0.4875
0.4850
0.4825
0
1000
2000
3000
4000
5000
Time

Findings

- For the case with penalty, the curves are at higher values and more dispersed

Conclusion

- Presented a very crude version of MARL model for flock learning

- Some things still work....

- Still left with many hyperparameters to tune

- Many other directions to explore...

Reference

Thesis work by André van Delft:

"Modelling collective motion with orientation-based rewards"

Huygens-Kamerlingh Onnes Laboratory, Leiden University

Github repository: flock-learning