Modelling collective motion with orientation-based rewards

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
PHYSICS

Author :	André van Delft
Student ID :	s1121367
Supervisor :	Dr. L. Giomi
2^{nd} corrector :	Prof. dr. ir. T.H. Oosterkamp

Leiden, The Netherlands, July 4, 2020

Modelling collective motion with orientation-based rewards

André van Delft

Huygens-Kamerlingh Onnes Laboratory, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

July 4, 2020

Abstract

The recent popularization of machine learning as a new paradigm in computer science provides interesting opportunities for explaining phenomena of collective motion in living systems, as for example flocks of birds or schools of fish. In this thesis we develop a model for collective motion using multi-agent reinforcement learning with orientation-based rewards, a new type of reward system that has not yet been found in literature. While the developed model is in principle generally applicable to all forms of collective motion observed in nature, we use the language of the flocking behaviour of birds as a particular example to frame our model. The birds have the option to either fly into an insticnctive direction or act based on a Viscek-type of interaction with their neighbors, and are rewarded maximally when the resulting direction of movement is some predetermined prefered direction. The model distinghuishes between leaders that instinctively move towards this direction and followers that do not. We show that collective motion into this prefered direction emerges from this model, but only with a minimum of 1.23 encounters with neighbours on average, of which a minimal fraction of 0.2 should be leaders, which on average roughly corresponds to at least one encounter with a leader every four timesteps. These lower bounds are rudimentary estimates, as the present study serves mainly as a proof of concept that collective motion can emerge from this new type of model. Additionally it is suggested that, using deep reinforcement learning, this model can be viewed as a reinforcement learning extension of the Vicsek model.

Many thanks to Ireth Garcia Aguilar for her extensive reviews and critical questions, to Leindert Boogaard for his read-through and general scientific advice during our refreshing lunch breaks, and to my wife Fae for her unconditional support with bending the last of my efforts into finishing this thesis. A thesis that marks off the end of a long and intensive, but also a diverse and enriching study track.

Contents

Chapter 1

Introduction

In the past decades, many models have been proposed that simulate collective motion in active matter. Classical examples of such models are Reynolds' boids and the Vicsek model [1, 2]. Models similar to these exist in a great variety [3], and have been very succesful in describing collective motion, but always in a very *artificial* or *mechanical* way. By this we mean that the individual particles usually behave as prescribed by a handful of rules that are given a priori.

For example, in the Vicsek model, N self-driven particles are located in a two-dimensional field with periodic boundaries with some randomly initalized positions $\boldsymbol{x}_0^i$ and velocities $\boldsymbol{v}_0^i$. The speed at which the particles move is fixed at some constant value v_0 and at integer timesteps t the direction of movement of each particle is updated as follows:

$$\theta_t^i = \langle \theta_{t-1}^i \rangle_d + \eta, \tag{1.1}$$

i.e., it is averaged over the flight direction of neigbouring particles within some distance d from it and a noise term η is added which is picked at random from some uniform distribution.

While the original aim in the study by Vicsek et al. [2] was to provide an example of phase transitions in active matter,[1] this model has, among others, driven the study of collective motion both in living [4, 5] and non-living systems [6, 7]. However, while a model like this might be sufficient for the latter type of systems, such an approach does not satisfactorily explain the much more widely studied cases of collective motion in living systems, as for example flocks of birds or schools of fish. Birds or fish do not seem to be governed by simple natural laws like particles or planets do; as animals they should be seen as agents that *learn* from their environment and adapt their behaviour to it.

Now, with the popularization of machine learning, which has recently influenced the study of active matter greatly [8], *reinforcement learning* in

[1] Where in particular a critical value η_c of the size of the noise distribution is found, distinghuising between an ordered phase (i.e., collective motion) below η_c and a disordered phase above η_c [2].

particular [9] can be used to adress this problem. In reinforcement learning, agents inhabit some complex environment and are rewarded based on their behaviour. Using machine learning algorithms, agents then adjust their behaviour with the aim of maximizing the total received reward. This avoids the problem of imposing mechanical laws on the agents as classical models do. A lot of research has been done at the intersection of collective motion and reinforcement learning, where sometimes a general model is proposed [10–14], but often models are developed with real-world applications in mind, such as flocks of birds [15, 16], schools of fish [17, 18], ant colonies [19], locusts [20], groups of people [21, 22], bots [23] or microswimmers [24].[2]

At least three types of reinforcement learning models can be distinguished in literature, categorized by their reward system:

1. *Flock-rewarding* models, i.e., models with a reward system that rewards agents based on their alignment with their neighbours [11, 15, 20], or some other explicit rule for fomation control [12, 14]. It might be disputed however wether such a model solves the problem the classical models have satisfactorily, since collective behaviour is still explicitly imposed on the agents via such a reward system.

2. *Predator-prey* models [13–16]. In such a model, a predator and several preys are put into some environment, where the predator attempts to catch the preys. Rewards might then be given at each timestep to the preys, to encourage these agents to find strategies that allow them to survive longer (and receive more rewards). Collective motion has regularly been observed to be a possible strategy in such models.

3. *Energy-minimizing* models [17, 18]. In this type of model, a hydrodamic fluid is modeled in which agents are rewarded that choose energetically preferable configurations, i.e., minimize the required energy for their movement. This has been a common explanation for collective motion observed in nature [25, 26].

In this thesis we develop an additional type of model that has not yet been found in literature: one with *orientation-based* rewards. The model we develop uses a leader-follower system in an environment where agents are rewarded for flying in the right direction. Such a reward system is reminiscent to how migratory birds take up cues of the environment to navigate, like temperature gradients or magnetic fields [27–29], but might also find its applications to any other situation in which a group of agents has to navigate through a given environment.

In order to arrive at such a model, we first explain the theory of reinforcement learning in chapter 2, where in particular we introduce a widely

[2] It is obviously disputable whether we can call the latter two applications living systems, but they are nevertheless included because of the use of reinforcement learning in the cited studies. Wether this means that we should stretch the definition of living systems, or we should include some notion of learning in non-living systems,will be left as a (philosophical) problem not relevant for our present study.

used algorithm called *Q-learning* [30]. We then generalize this to the theory of *multi-agent* reinforcement learning, and respond to the theoretical difficulties associated with it. After this we formulate our own model in chapter 3 and report the results of this model in chapter 4. Finally, we present our conclusions in chapter 5.

Chapter 2

Theory

2.1 Reinforcement learning

In *reinforcement learning* (RL), an agent is let to explore some interactive environment that can occupy different states, in which the agent can perform certain actions. Based on the action the agent chooses and the state the environment is in, the agent is rewarded with some reward signal, while the environment transitions into a new state. The goal of the agent is to adjust its behaviour such that the reward signal is maximized. An instance of RL is thus always characterized by three different sets: a set $\mathcal{S}$ that parametrizes the different states of the environment, a set $\mathcal{A}$ that parametrizes the different possible actions the agent can perform and a set $\mathcal{R} \subset \mathbb{R}$ of possible reward signals the agent can receive. For most implementations of reinforcement learning (including ours) it is necessary for $\mathcal{R}$ to be bounded from above and $\mathcal{S}$ and $\mathcal{A}$ to be finite (which often is sufficient, although it might mean in some cases that the environment has to be discretized), though generalizations exist of reinforcement learning with for example a continuous state or action space [9]. In addition, the dynamics between the agent and the environment is parametrized by the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$ and the transition function $p : \mathcal{S}^2 \times \mathcal{A} \to [0, 1]$. Given a *state-action pair* $(s, a) \in \mathcal{S} \times \mathcal{A}$ (i.e., the environment is in state s and the agent chooses action a consequently), $r(s, a)$ is the reward given to the agent and $p(s' \mid s, a)$ is the probability that the environment will transit to state s'.[1] Finally, the behaviour of the agent is determined by its so-called *policy* $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, where $\pi(a \mid s)$ is the probability that an agent chooses action a, given that the environment is in state s. These components together allow RL to be be formulated in terms of what is called a *Markov Decision Process* (MDP) [9, 31].

An episode in reinforcement learning hence consists of a sequence of timesteps parametrized by an integer t in which the environment occupies

[1] The probabilistic transition function is introducted for generality. In our case we will have a deterministic environment, which of course can be reobtained by for each state-action pair (s, a) choosing $p(s' \mid s, a) = 1$ for exactly one $s' \in \mathcal{S}$ and zero otherwise.

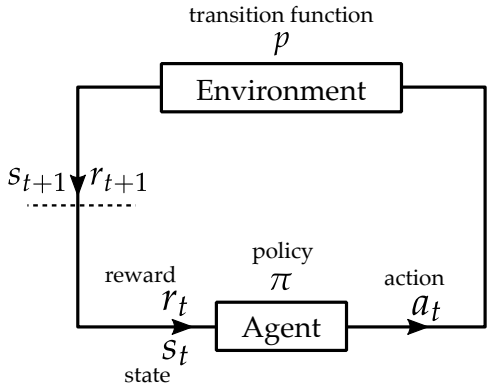

Figure 2.1: *Schematic depiction of the learning procedure in reinforcement learning (RL). The agent observes the state s_t of the environment at timestep t and chooses an action a_t. Based on this state-action pair, the agent is then rewarded with the reward $r_t = r(s_t, a_t)$ and the environment updates to state s_{t+1}. This cycle then repeats for timestep $t + 1$.*

state $s_t \in \mathcal{S}$, the agent performs performs some action $a_t \in \mathcal{A}$ and receives a reward $r_t = r(s_t, a_t) \in \mathcal{R}$ accordingly. The environment then updates to the next state s_{t+1} based on its transition function p, after which the whole cycle is repeated (cf. figure 2.1 for a schematic depiction of these quantities).

In a typical RL-problem, the environment and its dynamics, i.e., p and r, is assumed given, though not necessarily known to the agent. The goal of the agent is then to find the optimal policy π in this environment such that the *discounted cumulative future reward signal*

$$G_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n} \tag{2.1}$$

is maximized. The parameter $\gamma \in [0, 1)$ is called the *discount factor*, which is introduced to ensure that this sum does not diverge. Specifically, given some $R \in \mathbb{R}$ such that $r_t \leq R$ for all t (which always exist because $\mathcal{R}$ is bounded from above), it is guaranteed that

$$G_t \leq \sum_{n=0}^{\infty} \gamma^n R = \frac{R}{1-\gamma} \tag{2.2}$$

for all t.

2.2 Q-learning

Now that we explained the framework and the goal of RL, it is time to look at *how* RL attempts to achieve this goal. Or, to put it more concretely: how can we optimize the policy π of the agent such that G_t is maximized?

A lot of different algorithms have been developed that help achieve this goal, but probably the most widely known and generally applicable of these is called *Q-learning*, developed by Watkins [30]. Because we will

use it in our model, it is worthwile to explain the inner workings of this algorithm here.

In Q-learning, the agent is provided with a specific value $Q(s,a) \in \mathbb{R}$ corresponding to each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ which is called a *Q-value*. These Q-values are initialized with random values $Q_0(s,a) \in \mathbb{R}$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, and are updated during a reinforcement learning episode. This happens according to the following rule: given state s_t, action a_t, reward $r_t = r(s_t, a_t)$ and updated state s_{t+1}, the Q-value belonging to the state-action pair (s_t, a_t) gets updated as

$$Q_t(s_t,a_t) = (1-\alpha)Q_{t-1}(s_t,a_t) + \alpha\left(r_t + \gamma \max_{a\in\mathcal{A}} Q_{t-1}(s_{t+1},a)\right) \tag{2.3}$$

while the other values remain constant ($Q_t(s,a) = Q_{t-1}(s,a)$ for $(s,a) \neq (s_t,a_t)$). In this equation, $\alpha \in [0,1]$ is called the *learning rate* and γ is the same discount rate as in equation (2.1) (we will see how these are related in section 2.2.2). The collection of Q-values $Q(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ is called the *Q-table* of the agent.[2]

In words we can describe this equation as follows: the new Q-value is equal to the weighted average of the old Q-value and a new value, consisting of the direct reward signal r_t and the estimated maximal Q-value attainable one timestep ahead, reduced by the discount factor γ. The learning rate α determines the relative weight of these terms, i.e., how much the new value will influence the current value.

One should interpret these Q-values as estimates for the discounted cumulative future reward signal G_t. Since they are initialized randomly, these estimates are not very good at the beginning, but the theory of Q-learning guarantees that they should get better over time, eventually converging to G_t. In section 2.2.2 we will show why this is the case, but first we want to explain how these Q-values relate to the policy π of the agent.

2.2.1 The ϵ-greedy policy

We mentioned that the goal of RL is to find the optimal policy π. In Q-learning, the Q-table is in principle directly related to the policy: given the state s_t, the agent will choose the action $a \in \mathcal{A}$ that corresponds to the hightest Q-value $Q(s_t,a)$ (i.e., the hightest value in the row of the Q-table corresponding to s_t).[3] Note that Q-learning thus provides the agent with a deterministic policy.

In practice such a policy does not always yield the most optimal result however. This is especially the case in the beginning of an episode, where all the Q-values are randomly initialized. To overcome this problem, the agent should be allowed to explore other states that do not necessarily correspond to the maximum Q-value, giving room for all the Q-values to

[2] It is called a Q-table because it can be made into an $|\mathcal{S}| \times |\mathcal{A}|$ table with the states on one axis and the actions on the other.

[3] Or one of the maximum values at random, if there are more than one. However, since this is a very exceptional case, we will ignore this complication in the present exposition.

converge to their optimal values. One way in which this is often done is by making use of what is called an *ϵ-greedy policy* [9].[4] For this policy, a parameter $\epsilon \in [0, 1]$ is introduced, and the following rule holds: at each timestep the agent will either perform the action that has the maximum Q-value, with a chance $1 - \epsilon$, or perform some action at random with a chance ϵ. Quantitatively this means that, given that the environment is in state s,

$$\pi(a \mid s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } Q(s,a) = \max_{a' \in \mathcal{A}} Q(s,a') \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise.} \end{cases} \tag{2.4}$$

It is a common practice when using this policy in an episode, to start with some non-zero value of ϵ, and to either let ϵ slowly decrease to zero or to keep it constant and set $\epsilon = 0$ after a certain number of timesteps. In such an episode we can hence distinguish between two phases: the *training* or *exploration* phase, where $\epsilon \neq 0$ and the *trained* phase where $\epsilon = 0$ and the agent thus acts according to the (deterministic) policy as prescribed by Q-learning.

2.2.2 The dynamics of the update rule

To understand the specific form of the update rule (2.3) better, it can be instructive to get a feel for the dynamics of this equation. This can help for example with making an informed choice of the value of the parameters α, γ and ϵ. In order to do this, assume that each Q-value by repeated iteration converges to some value $Q^*(s,a)$. Because it has converged, by equation (2.3) there should hold

$$Q^*(s_t, a_t) = (1 - \alpha) Q^*(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a \in \mathcal{A}} Q^*(s_{t+1}, a) \right),$$

which we can rearrange as

$$Q^*(s_t, a_t) = r_t + \gamma \max_{a \in \mathcal{A}} Q^*(s_{t+1}, a).$$

Note that this is a recursive equation. Furthermore, note that if the exploration phase of the agent has ended (i.e., $\epsilon = 0$), the a for which in the above equation $Q^*(s_{t+1}, a)$ is taken, will also be the next action that will be chosen by the agent, since its Q-value is the highest. Therefore

$$\max_{a \in \mathcal{A}} Q^*(s_{t+1}, a) = r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q^*(s_{t+2}, a)$$

and so

$$\begin{aligned} Q^*(s_t, a_t) &= r_t + \gamma \left(r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q^*(s_{t+2}, a) \right) \\ &= r_t + \gamma r_{t+1} + \gamma^2 \max_{a \in \mathcal{A}} Q^*(s_{t+2}, a). \end{aligned}$$

[4] Other choices for exploration policies include a softmax policy and weighted roulette action selection [15].

By reapplying this same logic recursively we see that, by equation (2.1), $Q^*(s_t, a_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... = G_t$. Thus we should interpret the Q-values as estimates for the discounted future reward signal G_t, which should get better over time. This means that once the exploration phase has ended and the Q-values are sufficiently close to Q^*, the goal of RL, which is acting such as to maximize G_t, has been achieved.

Of course, this reasoning crucially depends on the assumption that the Q-values do in fact converge to a certain set of values. This has been theoretically proved in literature. Firstly, for Markovian systems like these, it has been shown that at least one optimal deterministic policy π^* does indeed exist such that G_t is maximized [32, 33]. Consequently, Watkins and Dyan [31] proved that for each finite action and state space and reward space that is bounded from above, and given sufficient exploration of the possible state-action pairs,[5] this Q-learning algorithm should eventually converge to a fixed set of Q-values. We have shown above that such a Q-table corresponds to an optimal policy π^*.

How quickly this convergence happens is of course very dependent on the specific model—i.e., the form of $\mathcal{A}$, $\mathcal{S}$ and the dynamics of the environment—and the choice of the learning parameters α, γ and ϵ. This is the main challenge in the application of many machine learning techniques however,[6] and can only be addressed by carefully tracking the model-specific learning process. We will adress how we do this in our model in chapter 3.

2.3 Multi-agent reinforcement learning

The theoretical framework discribed thus far is only applicable to a single learning agent. Since we are interested in collective motion of a group of N agents, we want to generalize this model to what is called *multi-agent reinforcement learning* (MARL). In order to do this, we simply change our action state to a vector $\boldsymbol{a} \in \mathcal{A}^N$ and our reward system to a vector $\boldsymbol{r} \in \mathcal{R}^N$ accordingly, where the i-th component a^i of $\boldsymbol{a}$ represents the performed action of agent i and the i-th component r^i of $\boldsymbol{r}$ its received reward (i.e., $r^i = r(s, a^i)$, given that the environment was in state s). The state of the environment is still parametrized by a single $s \in \mathcal{S}$, though it is common in MARL that not all agents have full access to the whole environment. Rather, each agent observes its own localized subset of the environment. All of the possible observations the agents can make are parametrized by a new set $\mathcal{O}$, the *observation space*. Additionally, an observation function $\varphi^i : \mathcal{S} \to \mathcal{O}$ is introduced that translates the state s of the environment to the observation $o^i = \varphi^i(s)$ of the agent. Since the observation is all the information the agent has access to, its policy is now a function $\pi^i : \mathcal{O} \times \mathcal{A} \to [0, 1]$

[5] They have, of course, precisely defined what they mean by 'sufficient' in their mathematical proof, but I will not go into these details here.

[6] This is also a great challenge for example in the design of neural networks [34].

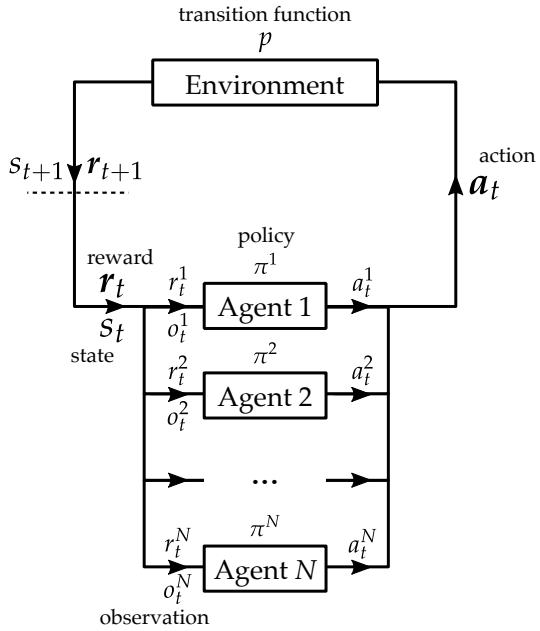

Figure 2.2: *Schematic depiction of the learning procedure in mulit-agent reinforcement learning (MARL). The i-th agent performs an observation o_t^i of the state s_t of the environment at timestep t and chooses an action a_t^i. Based on this observation-action pair, the agent is then rewarded with reward $r_t^i = r^i(o_t^i, a_t^i)$ and the environment updates to state s_{t+1}. This cycle then repeats for timestep $t+1$.*

At each timestep, the procedure of a single RL agent, discussed in section 2.1, is performed simultaneously for all agents: each agent observes the state of the environment (which now is limited to the observation $o \in \mathcal{O}$), decides individually on the action he will perform, after which the environment transits to a new state. This transition of the environment is now dictated by the $(N+2)$-argument transition function $p : \mathcal{S}^2 \times \mathcal{A}^N \to [0,1]$, where $p(s' \mid s, \boldsymbol{a})$ is the chance that the environment transits to s', given the previous state s of the environment and action vector $\boldsymbol{a}$ of the agents. After this transition, the cycle repeats. (cf. figure 2.2 for a schematic depiction of all these quantities). Each agent is then either assigned the task of maximizing their own cumulative discounted future reward signal

$$G_t^i = \sum_{n=0}^{\infty} \gamma^n r_{t+n}^i, \tag{2.5}$$

or they might be asigned the task of maximizing some collective of reward signals (introducing the option of having groups with opposed goals, for example).

2.3.1 Formalizations of MARL

While this step from single-agent to multi-agent RL is conceptually easy to make, the theoretical framework has to be reconsidered. Specifically, the

convergence toward an optimal policy in Q-learning is no longer guaranteed, since it relies on the fact that the that the transition of the environment is predictable for the agent, i.e., only dependent on the state it observes and the action that it preforms.[7] With MARL however, this transition function also depends on the performed actions of the *other* agents, which are simultaneously learning, and thus act in an unpredictable way.

A lot of different studies have been published on the topic of MARL [35], with a variety of different learning algorithms and strategies that attempt to overcome the mentioned difficulties associated with having multiple learning agents at once [36, 37]. Notable examples of this are the introduction of Nash equilibria to calculate optimal strategies for agents with opposed goals [38], the theory of Markov games as a formalization of a MARL-problem, similar to MDP's in the single agent case [39]. Also the Deepmind team has recently published a lot of different complex, and sometimes highly specialized algorithms in the field of MARL [40–43].

Despite all these different strategies to formalize MARL and the algorithms that have been developed for this, we still choose to use the Q-learning algorithm of section 2.2, for a couple of reasons:[8]

1. A lot of MARL algorithms use some very sophisticated ways of anticipating on the behaviour of other agents (e.g., the minimax policy used by [39]). This has to do with the fact that a lot of these studies were performed with applications to game theory in mind. It seems unlikely however that we need the full strategic power of a chess player in order to have a bird learn to flock.

2. Consequently, while many of these algorithms are usually developed as (steps toward) a general framework for MARL, in practice they have only been applied to games with a few players (e.g., [38]). This raises the question whether the algorithms developed are computationally feasible for a system with a number of agents $N \sim 10^2$.

3. If it is the case that the single-agent Q-learning algorithm is not sufficient for explaining collective motion, than that is interesting information on its own, indicating that in real-life applications agents have more complicated considerations that initially thought. Conversely, if it is the case that the Q-learning algorithm is sufficient for our purposes, despite not meeting the formal requirements, than that might indicate that Q-learning has a wider applicability than initially thought.

[7] This still holds when this transition function is probabilistic, in which case one can still define an expectation value of the expected reward signal at, for example, one timestep ahead as

$$r_{t+1} = \max_{a \in \mathcal{A}} \mathbb{E}[r(s_{t+1}, a)] = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} p(s' \mid s_t, a_t) r(s', a).$$

Algorithms like Q-learning formally require at least these expectation values to be constant [31], which in general is not the case anymore for MARL.

[8] Other studies in the field of collective motion have also done this, e.g., [16, 22].

Our hypothesis therefore is that Q-learning is a sufficient framework for the application in mind, and whether this is true or not should be judged using the results of our model.

Chapter 3

Formulating the model

As mentioned in the introduction, the main goal of this thesis is to develop a model that describes collective motion using reinforcement learning with orientation-based rewards. There are two important properties that such a model should have:

1. Collective motion should not be put into the model a priori (as is the case for both the classical and flock-rewarding models). Otherwise the model can not provide a proper *explanation* of the phenomenon of collective motion.

2. At least some agents should actually learn to follow the others. If this is not the case, that means that each agent is simply learning to move toward the right direction on its own. Consequently, while all individual agents might have learned to move toward the right direction, the movement of the group can not really be called *collective*, since the learning process is very individual and independent from the other agents.

In this chapter we explain the model that has been developed and the motivation behind the assumptions of the model. In the explanation that follows, we choose to use the language for describing the flocking behaviour of birds. While this is a very common example of collective motion, there is no reason not to generalize this model to other well-studied applications. We use terms related to the behaviour of birds primarily for convenience, so that general terms like *collective motion* and *agents* can be replaced by their shorter and more tangible counterparts *flocking* and *birds* respectively.

The general framework of the model is the following: N birds are initialized with random positions $\boldsymbol{x}_0^i$ and flight directions $\theta_0^i \in (-\pi, \pi]$ in some two-dimensional square field with sidelength L and periodic boundaries. The birds all fly at the same constant speed v_0. Following the reinforcement learning paradigm, the birds perform an observation $o^i \in \mathcal{O}$, adjust their flight direction by means of the possible actions available in the action space $\mathcal{A}$, and are then rewarded accordingly with some reward

r^i. In the following sections we explain the specific form of each of these quantities separately.

3.1 The reward system r

A natural choice for an orientation-based model is to reward the birds that are flying toward some preferential direction. This preferential direction can in general vary from place to place, but for simplicity we choose to reward the same direction everywhere, namely the eastward direction ($\theta = 0$). This can be a discrete reward system, e.g.,

$$r_t^i = \begin{cases} R & \text{if } \theta_t^i = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

for some $R \in \mathbb{R}$, or one could implement a gradient reward system

$$r_t^i = R \cos \theta_t^i. \tag{3.2}$$

Especially the latter might be reminiscent to orientational cues in nature like temperature, or magnetic fields. We will experiment with both reward systems, however.

3.2 The state space $\mathcal{S}$ and observation space $\mathcal{O}$

Since there are no other objects or dynamics in the field other than the flying birds themselves, a full state $s \in \mathcal{S}$ of the environment is thus simply given by all N positions $\boldsymbol{x}^i$ and flight directions θ^i of the different birds. Not every agent has this full knowledge however. As is common in models for collecitve motion, a bird only has information about its neighborhood. Thus we define an observation $o_t^i \in \mathcal{O}$ of bird i at timestep t as follows: the bird observes all neighbouring birds within distance d from it, and tracks the flight direction θ^j of all these birds.

As discussed in the previous chapter, Q-learning requires that the observation space $\mathcal{O}$ is finite. Therefore we discretize the possible flight directions. In order to maintain the symmetry of the square field, there should hold $|\mathcal{D}| = 2^k$ with $k \geq 2$, where $\mathcal{D}$ is the set of all allowed flight directions.[1] For example, when $|\mathcal{D}| = 2^2 = 4$, these directions correspond to the four cardinal directions. However, this choice introduces undesired artifacts in the model,[2] so we take $k = 3$ as a lower bound.

Additionally, it is preferable to put an upper bound M on the observed neighbours per direction, mostly from a computational point of view. Combined with the discretization of the flight directions, this means we can

[1] By $|\mathcal{C}|$ we denote the size of a finite set $\mathcal{C}$, i.e., the number of elements it contains.

[2] Specifically because of the Vicsek action V that we include in the action space in section 3.3. The details for why this is the case (which we have chosen to omit in this thesis to maintain clarity) can be found in a report on `http://github.com/andredelft/flock-learning`, under `observations/20200323.md#problem-in-the-ideal-policies`.

formally define an observation $o \in \mathcal{O}$ as a tuple $(n_\theta)_{\theta \in \mathcal{D}}$ where, given l_θ neighbouring birds flying in direction $\theta \in \mathcal{D}$,

$$n_\theta = \begin{cases} l_\theta & \text{if } l_\theta < M \\ M & \text{otherwise.} \end{cases} \tag{3.3}$$

For example, the observation performed by bird 1 in figure 3.1 will be

$$o^1_{t+1} = (0, 0, 0, 2, 1, 0, 0, 0)$$

where the components are arranged from 0 to 2π.

Since an observation now is a tuple of length $|\mathcal{D}|$ with each component having the possible values $0, 1, ..., M$, there holds

$$|\mathcal{O}| = (M+1)^{|\mathcal{D}|}. \tag{3.4}$$

Thus $|\mathcal{O}|$ grows exponentially with $|\mathcal{D}|$. Since $|\mathcal{O}|$ dictates the number of rows in a Q-table, this can quickly become very large, which is a computational problem.[3] Therefore we choose to fix $|\mathcal{D}|$ at the lower bound $2^3 = 8$, i.e.,

$$\mathcal{D} = \left\{ \frac{n\pi}{4} \;\middle|\; n \in \{ 0, 1, ..., 7 \} \right\}. \tag{3.5}$$

Additionally we choose $M = 2$, so that $|\mathcal{O}| = 3^8 = 6561$.

3.3 The action space $\mathcal{A}$

We considered several different actions that might constitute the action space of the birds:

1. **The four cardinal directions** $\{ \mathsf{N}, \mathsf{E}, \mathsf{S}, \mathsf{W} \}$**.** When one of these actions is chosen, the bird changes its direction of motion to the associated cardinal direction North ($\theta_t = \pi/2$), East ($\theta_t = 0$), South ($\theta_t = -\pi/2$) or West ($\theta_t = \pi$) respectively. These actions essentially represent the (discretized) free movement of the birds.

2. **An instinctive direction** I**.** When introducting this action, each bird is provided with a certain direction (which is taken to be one of four cardinal directions) which represents the direction the bird would fly to by its own instinct, which it flies toward when choosing action I. Intruducing this allows us to distinguish between two types of birds: *leaders* for which $\mathsf{I} = \mathsf{E}$, meaning their instinctive direction is the 'right' direction (i.e., that which is rewarded maximally), and *followers* for which $\mathsf{I} \neq \mathsf{E}$. The frequency at which a given bird chooses I can be seen of as a measure of how much the bird *trusts* its own instinct.

[3] The total amount of Q-values we should store is $(M+1)^{2^k} \cdot |\mathcal{A}| \cdot N$. Our choice $k = 3$ and $M = 2$, given that $|\mathcal{A}| = 2$ and $N = 100$, means that we already have $1.3 \cdot 10^6$ Q-values. This will be much larger for $M = 3$ or $k = 4$ ($1.3 \cdot 10^7$ or $8.6 \cdot 10^9$ respectively). This is not only a memory issue, but also means that the Q-values converge much more slowly, since there are many more available policies to explore.

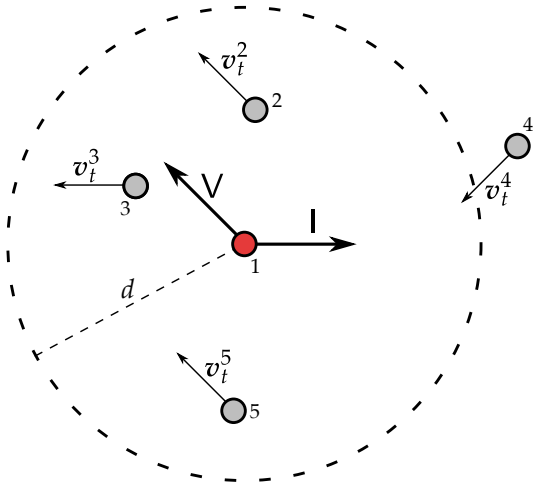

Figure 3.1: *A visualization of the possible actions in the action space* $\mathcal{A} = \{ \mathsf{V}, \mathsf{I} \}$ *for a bird in the field surrounded by some neighbours. When bird 1 chooses the action* V *at timestep* $t + 1$*, it will fly into the direction* $\theta \in \mathcal{D}$ *that is closest to the average flight direction of the neighbours within distance* d *from it. In this case, these are bird 2, 3 and 5, and the resulting flight direction will be* $\theta_{t+1}^1 = \frac{3\pi}{4}$*. If it chooses* I*, it will fly into its predetermined instinctive direction, which is one of the cardinal directions* $\{ \mathsf{N}, \mathsf{E}, \mathsf{S}, \mathsf{W} \}$*. In this case* $\mathsf{I} = \mathsf{E}$ *(i.e.,* $\theta_{t+1}^1 = 0$*), which means this particular bird is a leader (since flying into that direction will be rewarded maximally).*

3. **The Vicsek interaction** V**.** When choosing this action, the agent decides to adjust its flight direction to the average flight direction of the observed neighbours, i.e., those within a distance d of the bird. This coincides with the Vicsek update rule (1.1), only with zero noise, i.e., $\theta_t = \langle \theta_{t-1} \rangle_d$. Because we are dealing with a discrete number of flight directions however, this Vicsek step also has to be discretized. Thus when V is chosen, $\langle \theta_{t-1} \rangle_d$ is calculated and then rounded off to the nearest available $\theta \in \mathcal{D}$. Introducing this action means that this model can be seen as an extension of a discretized Vicsek model with zero noise,[4] which can be reobtained when $\mathcal{A} = \{ \mathsf{V} \}$.

   Quantitatively, $\langle \theta_t \rangle_d$ can be computed using the *two-argument arctangent*:[5]

$$\langle \theta_t \rangle_d = \operatorname{arctan2}\left(\sum_{j \in \mathcal{N}_{i,d}} \sin \theta_t^j, \sum_{j \in \mathcal{N}_{i,d}} \cos \theta_t^j \right), \tag{3.6}$$

[4] The possibility of adding the noise term η to this action (or others) has been investigated, but it has been proven difficult to implement, because of the discretization of the flight directions. We will discuss this problem further in section 5.1, where deep Q-learning is discussed as an extension of the current model, allowing for a continuous state space and thus contiuous flight directions.

[5] $\operatorname{arctan2}(y, x)$ is defined to always be the angle between the vector (x, y) in the Euclidean plane and the positive x-axis. For $x > 0$ this coincides with $\arctan(y/x)$, but this latter value does not represent the angle between (x, y) and the positive x-axis anymore in the regions where $x < 0$ (where it is off by $\pm\pi$ depending on the value of y) or $x = 0$ (where it is undefined). The function arctan2 corrects this.

where $\mathcal{N}_{i,d}$ is the set of indices of the birds that are within distance d of bird i.

While all of these are sensible choices for the action space, it turns out that including free motion in the model raises a problem. The reason for this is that both leaders and followers tend to choose E for all $o \in \mathcal{O}$ very quickly. While this in principle does lead to collective eastward motion, it violates the second property that we mentioned in the beginning of this chapter, since all birds are learning independently.[6] Therefore, we limit ourselves to the action space $\mathcal{A} = \{\mathsf{V}, \mathsf{I}\}$. Cf. figure 3.1 for a visualization of this action space.

3.4 Tracking the quality of the learning process: v and Δ

As with any machine learning technique, it is important to track the quality of the learning process. For this we use the quantities v and Δ. The first is the *normalized average flight direction*, and is given by

$$\boldsymbol{v}(t) = \frac{1}{v_0 N} \sum_{i=1}^{N} \boldsymbol{v}_t^i = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{\boldsymbol{x}} \cos\theta_t^i + \hat{\boldsymbol{y}} \sin\theta_t^i \right). \tag{3.7}$$

This is a measure for the alignment of the flock: when $|\boldsymbol{v}| = 0$ the birds are flying incoherently, and when $|\boldsymbol{v}| = 1$, the whole flock is aligned. Given the constraints of our model, this usually means that the flight angle is $\theta = 0$, though this should be checked either by calculating the angle explicitly, or checking the x-component v_x of $\boldsymbol{v}$.

We refer to Δ as the *normalized distance from the optimal policy*. It is a quantity that is defined for $\mathcal{A} = \{\mathsf{V}, \mathsf{I}\}$ specifically and is derived from the Q-tables of the birds. Remember from chapter 2 that the Q-values in a Q-table of a bird, when trained properly, reflect its expected future reward signal. When the birds are trained and stop exploring, these Q-values are directly related to the policy of the bird, such that the bird will always choose the action $a \in \mathcal{A}$ that corresponds to the highest Q-value $Q(o, a)$, given the observation $o \in \mathcal{O}$. See table 3.1 for a sample of a possible Q-table, for the action and observation space as we have defined them in this chapter.

Given the action space $\mathcal{A} = \{\mathsf{V}, \mathsf{I}\}$, a natural policy for the leaders would be to always choose I, since that action will always be maximally rewarded. Conversely, the followers should always choose V, since their own instinctive direction by definition will not lead to the maximum reward R. Following their neighbours and trusting that the collective will end up flying eastward might thus be the best they can do. As we will see in section

[6] Just as with the unwanted artifacts for $|\mathcal{D}| = 4$ (cf. note 2), we choose to omit the details here to maintain clarity. We refer the interested reader again to `http://github.com/andredelft/flock-learning`, specifically the data in `data/20200229`.

Table 3.1: *An example of a Q-table of a bird, with the action and observation space as developed in this chapter. Each row represents an observation $o \in \mathcal{O}$ and each column an action $a \in \mathcal{A}$. If this is the Q-table of a trained bird (i.e., this Q-table is fixed and the bird does not explore), it will always choose action* I *when there are no neighbouring birds, action* V *when there is one neighbouring bird flying in the direction $\theta = 0$, and so on.*

	V	I
$(0,0,0,0,0,0,0,0)$	0.1	5.0
$(1,0,0,0,0,0,0,0)$	3.5	0.9
$(2,0,0,0,0,0,0,0)$	10.0	0.2
$(0,1,0,0,0,0,0,0)$	8.1	-5.2
...	...	...
$(2,2,2,2,2,2,2,2)$	-1.2	1.9

4.1, this particular configuration indeed leads to collective motion toward $\theta = 0$, even for a surprisingly low fraction of leaders ($\gtrsim 1\%$).

Given this fact, which will be justified by our results, this policy can be referred to as an *optimal policy* in the sense that is described in section 2.2.2. This is because, given that this policy leads to full collective eastward motion, each bird will at each timestep receive the maximum reward R, and thus for each bird G_t^i is at its theoretical maximum $R/(1-\gamma)$ (cf. equation (2.5)).

For this reason, we would like to judge how *close* a particular configuration of birds is to this optimal policy. For this we define the following function for leaders

$$\delta_l^i(o) = \begin{cases} 0 & \text{if } Q^i(o,\mathsf{I}) > Q^i(o,\mathsf{V}) \\ 1 & \text{if } Q^i(o,\mathsf{I}) \leq Q^i(o,\mathsf{V}) \end{cases} \tag{3.8}$$

and, conversely, for followers

$$\delta_f^i(o) = \begin{cases} 0 & \text{if } Q^i(o,\mathsf{V}) > Q^i(o,\mathsf{I}) \\ 1 & \text{if } Q^i(o,\mathsf{V}) \leq Q^i(o,\mathsf{I}) \end{cases} \tag{3.9}$$

for each possible observation $o \in \mathcal{O}$. If we sum over these functions, we essentially count how many rows in the Q-tables deviate from the above defined optimal policy. The normalized distance is then calculated by performing this sum and normalizing:

$$\Delta = \frac{1}{N|\mathcal{O}|} \sum_{o \in \mathcal{O}} \left(\sum_{i \in \mathcal{L}} \delta_l^i(o) + \sum_{i \in \mathcal{F}} \delta_f^i(o) \right), \tag{3.10}$$

where by $\mathcal{L}$ and $\mathcal{F}$ we refer to the set of indices of the leaders and followers respectively (so $N = |\mathcal{L}| + |\mathcal{F}|$). From this definition follows that $\Delta \in [0,1]$ and that $\Delta = 0$ if and only if the Q-tables of the birds prescribe the defined optimal policy.

Note that it is not guaranteed that the described policy is the *only* optimal policy of the system, but our hypothesis will be that all other optimal

policies will at least have a value of Δ that is close to 0. Whether or not that is the case, at the very least Δ can be a treated as a point of reference for tracking the evolution of the Q-tables in an episode.

Chapter 4

Results

We now present the results for the orientation-based MARL model outlined in the previous chapters. The model has been developed using Python 3 and is publicly available under the MIT licence, provided with documentation.[1]

4.1 The role of Δ

Before reporting the results of our model, we first want to explore wether the normalized distance from the optimal policy Δ defined in section 3.4 is a good indicator of the quality of the learning process. For though v is a common and straightforward way of quantifying collective motion, Δ is very specific for our model with action space $\mathcal{A} = \{ \mathsf{V}, \mathsf{I} \}$.

To investigate this, we performed several runs with randomly initialized Q-tables with the constraint of having a certain predefined value of Δ.[2] We regarded these as trained birds (i.e., $\epsilon = 0$ and $\alpha = 0$) and measured v for 1500 timesteps. For the other parameters of the system, we used the default values listed in table 4.1. We then averaged the magnitude of v over the last 1000 steps and plotted the resulting value $\langle v \rangle$ against Δ (figure 4.1). Initially, we scanned over the whole range $\Delta \in [0, 1]$. However, since all simulations start at around $\Delta = 0.5$ and generally decrease afterwards,[3] we additionally looked more closely at the region $\Delta \in [0, 0.5]$.

A definite negative trend can be observed in the latter region, starting from $\Delta = 0$ and ending at $\Delta = 0.5$. Additionally, in line with our hypothesis formulated in section 3.4, $\langle v \rangle = 1$ for $\Delta = 0$, meaning that this policy indeed yields the optimal result (i.e., the maximal long-term reward signal).

[1] `https://github.com/andredelft/flock-learning`

[2] This can be achieved by starting from the Q-tables of the optimal policy (i.e., $Q(o, \mathsf{I})$ is maximal for all leaders, $Q(o, \mathsf{V})$ for all followers), and altering as much rows in the Q-tables of the birds at random untill Δ reaches the desired value.

[3] The Q-tables are randomly initialized, so in theory they can start at each possible value of Δ. However, it is statistically much more probable that the initial value of Δ is around 0.5, since the possible states form a binomial distribution over $\Delta \in [0, 1]$.

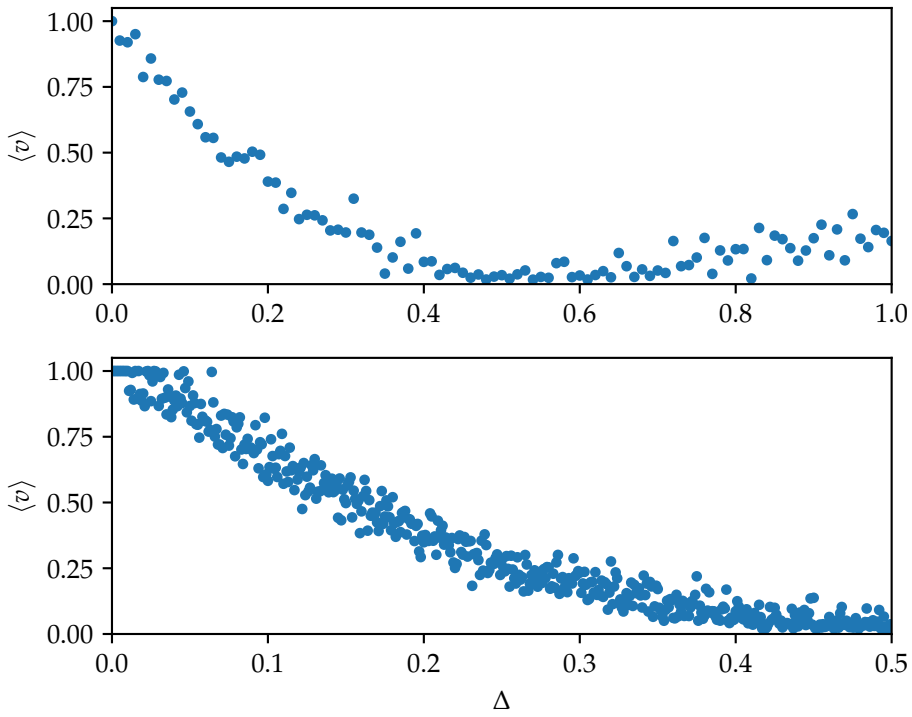

Figure 4.1: *The average magnitude of v for trained birds with randomly initialized Q-tables for a given value of Δ, scanned over the whole range $\Delta \in [0, 1]$ (top) and in more detail over the range $\Delta \in [0, 0.5]$ (bottom).*

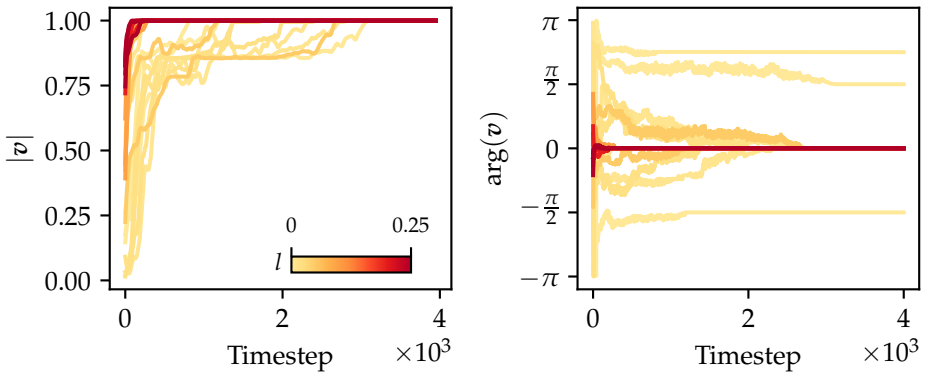

Figure 4.2: *The evolution of v at $\Delta = 0$ with varying leader fractions $l \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.25\}$ (three runs are graphed for each value of l). We observe that the flock converges very quickly to $\theta = 0$ for $l \geq 0.05$ (i.e., in the first 500 timesteps) and does eventually converge for $0 < l < 0.05$ as well, only much more slowly. For $l = 0$, i.e., an absence of leaders, the flock also converges, but does so in a random direction $\theta \in \mathcal{D}$.*

Table 4.1: *A list of all parameters of the model and their default values.*

Symbol	Parameter name	Default value
L	Dimension of the field (width and height)	800
N	Number of birds	100
l	Leader fraction	0.25
d	Observation distance	100
R	Maximum reward signal	5
v_0	Flight speed of the birds	1
α	Learning rate	0.1
γ	Discount factor	0.9
ϵ	Exploration parameter (ϵ-greedy)	0.5

Furthermore, in this data, other optimal policies also have a low value of Δ. Specifically, the highest value of Δ for which $\langle v \rangle > 0.99$ is $\Delta = 0.064$.

Additionally, we further investigated the configuration $\Delta = 0$, i.e., the proposed optimal policy where the leaders always choose I and the followers always choose V. We have tracked the evolution of v at this configuration for different leader fractions (cf. figure 4.2). We observe that the flock converges in this configuration to full eastward motion even for a surprisingly low fraction of leaders $l \geq 0.01$. Given our choice of parameters, this corresponds to the presence of at least one leader in the field. While this convergence typically happens in less than 500 timesteps, for low l ($0.01 \leq l \leq 0.02$, or 1 or 2 leaders) this takes up to 2500 timesteps. When no leaders are present ($l = 0$) the flock also converges eventually, but does so in a random direction $\theta \in \mathcal{D}$.

4.2 Exploring the parameter space

All parameters of the model that we have developed in the previous chapters are listed in table 4.1. These are separated into two categories: parameters relating to the birds and the environment, and the learning parameters that are used by the Q-learning algorithm. In this section we report our exploration of this parameter space to find the conditions for collective motion in our model. For this, we first investigated the effect of the learning parameters α, γ and ϵ, from which we make an informed choice for their respective values. After this, we explored the effect of the other parameters on the learning procedure.

But first some notes regarding the first set of parameters listed in table 4.1. There are six parameters listed that determine the dynamics of the birds and their environment. However, from the viewpoint of the individual birds and their policy-making there are only two things that really matter: the frequency of encounters between birds and how much of these observed neigbouring birds are leaders or followers. The latter is relevant because leaders and followers generally develop different policies. In particular, since leaders by definition have the option of choosing the maxi-

mally rewarded direction at each timestep, they are more likely to choose this direction than followers. Choosing to follow a leader can therefore in general be expected to lead to a higher reward than following a follower.

One might estimate the expected encounters n per timestep by multiplying the density ρ of the birds in the field by the area A that is observed each timestep by an individual bird, i.e.,

$$n = \rho A = \left(\frac{N}{L^2}\right) \cdot \pi d^2 = \frac{\pi N d^2}{L^2}. \tag{4.1}$$

The expected encounters of leaders and followers separately are then given by multiplying n with l and $(1-l)$ respectively.

Note that, since there is no specific lengthscale defined, decreasing the observation distance d is equivalent to increasing the dimension L of the field. This is also reflected in equation (4.1). And both of these changes have the same effect as decreasing N, namely lowering the average encounters. Given that the total computation time for all N nearest neighbour searches scales like $O(N^2)$,[4] it is preferable to keep N fixed at a computationally feasible number. We therefore choose to set $N = 100$, $L = 800$, and will vary d and l.

As for the remaining parameters v_0 and R, we argue that their specific value is not very relevant, provided they are sufficiently small and sufficiently large respectively. We can fix v_0 by noting that we also do not have some predefined timescale. However, to ensure relatively continuous motion and minimize the influence of the periodic boundaries, it should hold that $v_0 \Delta t \ll L$. We use integer timesteps, i.e., $\Delta t = 1$, and set $v_0 = 1$ for simplicity.

R in turn relates to the policy of the birds via the update rule (2.3). But the only thing that is important in the policy-making of the birds is which Q-value is maximal, not what its specific value is. And since there is a theoretical upper limit of $R/(1-\gamma)$ on the Q-values of the birds, only the value relative to this maximum is relevant, hence R factors out. What does help the optimal policies stand out however, is making sure that this maximum $R/(1-\gamma)$ is significantly bigger than the initial Q-values (i.e., distinguishing the 'signal' from the 'noise'). Therefore we take the initial Q-values $Q_0(o,a)$ to be uniformly distributed over the interval $[0,1]$ and choose $R = 5$, such that $R/(1-\gamma) = 5/(1-0.9) = 50$.

4.2.1 The learning parameters α, γ and ϵ

To explore the parameter space of the learning parameters α, γ and ϵ, we performed several runs where we varied one learning parameter at a time over the whole range $[0,1]$ (and $[0,1)$ for γ), while all other parameters have been fixed at their default values listed in table 4.1. For these runs we have used the gradient reward system.

[4] We use `scipy.spatial.KDTree` to do this, for which it is stated in the documentation that it should be expected not to be significantly faster that brute force [44].

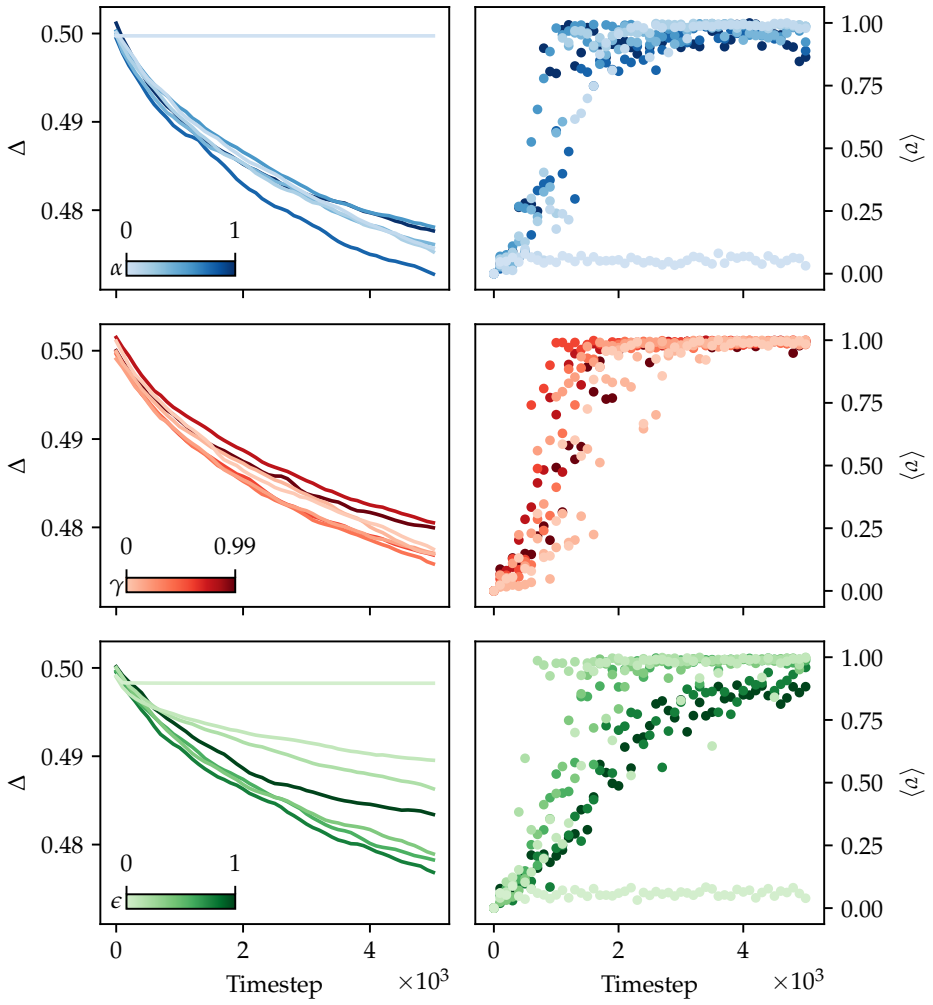

Figure 4.3: *We performed with varying values of the learning parameters α (top), γ (middle) and ϵ (bottom), while keeping all other parameters fixed at their values listed in table 4.1. We saved the Q-tables of these runs every 100 timesteps, from which Δ (left) and $\langle v \rangle$ (right) have been calculated and graphed as a function of time. In order to calculate the latter, we started new runs with the saved Q-tables as trained values, from which $\langle v \rangle$ is obtained by averaging v over 1000 timesteps (skipping the first 500 as initalization time). The values of the learning parameters that are graphed in these figures are $\{ 0, 0.1, 0.2, 0.4, 0.6, 0.8, 1 \}$ for α and ϵ and $\{ 0, 0.1, 0.2, 0.4, 0.6, 0.8, 0.99 \}$ for γ.*

The default values of the learning parameters have been partially informed by other studies. In particular, it is a common practice to choose a low value of α in order to keep the Q-values relatively stable and a high value of γ in order to factor in the long-term reward [9, 22, 31]. With regard to ϵ, choosing $\epsilon = 0$ corresponds to birds that do not explore at all. On the other hand, if $\epsilon = 1$, the policy of the birds in the learning phase would be completely random, meaning that it is impossible for the birds to anticipate upon the others (e.g., no significant difference between the policy of the leaders and followers will be observed in this case). We thus chose $\epsilon = 0.5$ as the default value, as a middle ground between these two extremes.

In order track the learning process, we saved the Q-tables regularly (every 100 steps). We used these calculate Δ. Additionally, we calculated $\langle v \rangle$ from each of these Q-tables, in the same way as we did for the runs in figure 4.1. That is, we started a separate run with birds initialized with these Q-tables, which were regarded as trained birds. $\langle v \rangle$ is then calculated by averaging v in these runs over 1000 timesteps (skipping the first 500 steps for initializaition of the flock). We graphed the evolution of both Δ and $\langle v \rangle$ as a function of the timestep at which the corresponding Q-tables are saved in figure 4.3.[5]

We found no evolution in both Δ and $\langle v \rangle$ when the learning rate $\alpha = 0$, as is to be expected from equation (2.3). From $\alpha \geq 0.1$ onwards we observed a decrease in Δ and generally a convergence of the flock after 2000 timesteps. However, for $\alpha \geq 0.4$ we observed that this convergence is not very stable, since $\langle v \rangle$ regularly drops in this region to about $\langle v \rangle = 0.9$. This indicates that a high value of α is not optimal for the learning process, which is consistent with the common practice of choosing a low value of α. A possible explanation for this is that the Q-values fluctuate to much, since the previous Q-values have a relatively low weight (cf. equation (2.3)). From these results we concluded that optimal learning happens in the region $0.1 \leq \alpha \leq 0.2$ (given $\gamma = 0.9$ and $\epsilon = 0.5$).

We observed no significant differences when varying γ. Although for $\gamma \leq 0.1$ the flock is less stable initially, eventually (after 3500 timesteps) the flock does converge for all values of γ. Since γ factors in the long term reward signal, this might indicate that no long term strategies exist in this model. However, it should also be noted that the number of timesteps required for convergence (usually around 500–2000 timesteps) is much lower than the number of values in the Q-tables of the individual agents. We discuss this complication further in section 4.2.2.

Similar to the run with $\alpha = 0$, we found that $\epsilon = 0$ results in no signif-

[5] To maintain clarity, the angle $\arg(\boldsymbol{v})$ is not explicitly shown in these (and subsequent) graphs, but it has been observed that when $\langle v \rangle = 1$, there always holds $\arg(\boldsymbol{v}) = 0$. To understand why this happens, note that the leaders always have action I available that allows them to fly eastward 'on their own'. As a consequence, they very quickly learn to only fly in that direction and when trained, will almost always do that. Therefore, if $\langle v \rangle = 1$ it is guaranteed that $\arg(\boldsymbol{v}) = 0$, since there must always at least a fraction of the birds that is flying into this direction.

icant evolution for both Δ and $\langle v \rangle$. In the region $0.1 \leq \epsilon \leq 0.6$, we generally observed a stable convergence of the flock, with some exceptions for $\epsilon = 0.1$. For $\epsilon \geq 0.8$ however, we observed that the flock does not converge completely. This indicates that the birds have more difficulty with learning to flock when the policy of the other birds is random. We concluded that optimal learning happens in the region $0.1 \leq \epsilon \leq 0.8$ (given $\alpha = 0.1$ and $\gamma = 0.9$).

4.2.2 γ at longer timescales

The previous results indicate that it generally takes around 500–2000 timesteps for the flock to converge (given our default parameters). Additionally, no we observed no significant influence of the discount rate γ on the learning process. Since γ factors in the long-term reward signal (as can be seen from equation (2.3)), this might indicate that no long-term decision making is present in our model.

However, it should also be noted that these timescales are too low to observe any effect of γ. To understand this, note that the number of Q-values in the Q-table of each bird equals $|\mathcal{O}| \times |\mathcal{A}| = 6561 \times 2 = 13{,}122$. In order to factor in the long term reward, it is necessary that these Q-values have sufficiently converged to the expected future reward signal. Since one value of a birds Q-table is affected by the update rule each timestep, and all values should ideally be updated multiple times in order to converge properly—or at least those that are associated with frequently performed observations—this means that the timescale 500–2000 timesteps might be too low to observe any effect of γ.

Thus, if we want to observe any effect of γ at all, we should measure our runs at longer timescales. Additionally, from equation (2.3) we find that the discount factor γ competes with the direct reward signal r_t. Therefore it is prefereable to minimize the influence of the direct reward signal, which we can do in two ways. Firstly we can use the discrete reward system (3.1), which means that $r_t = 0$ for all flight directions except $\theta = 0$. Secondly, we can analyse instances where, given the discrete reward system, whatever action the bird chooses, it will not be rewarded. For example, if a follower observes that the neighbouring flock is flying to the North, both actions V and I will result in a direct reward signal of 0 (because choosing V will result in $\theta_t = \frac{\pi}{2}$ and choosing I will not result in $\theta_t = 0$ by definition).

Therefore, in our analysis we separated the Q-tables of the leaders from the followers, and for each of these bird types we isolated the rows of the Q-table that correspond to observations $o \in \mathcal{O}$ in which a majority of the neighbouring birds is flying toward one of the cardinal directions $\{ \mathsf{N}, \mathsf{E}, \mathsf{S}, \mathsf{W} \}$. This resulted in eight different categories, for each of which we calculated the normalized distance from the optimal policy as done in section 3.4. The results of these new simulations are shown in figure 4.4.

We observed that, as in figure 4.3, there still is no significant effect of γ on the policy-making of the birds for the leaders. The same holds for the followers, but only in the case in which a majority of the birds is flying

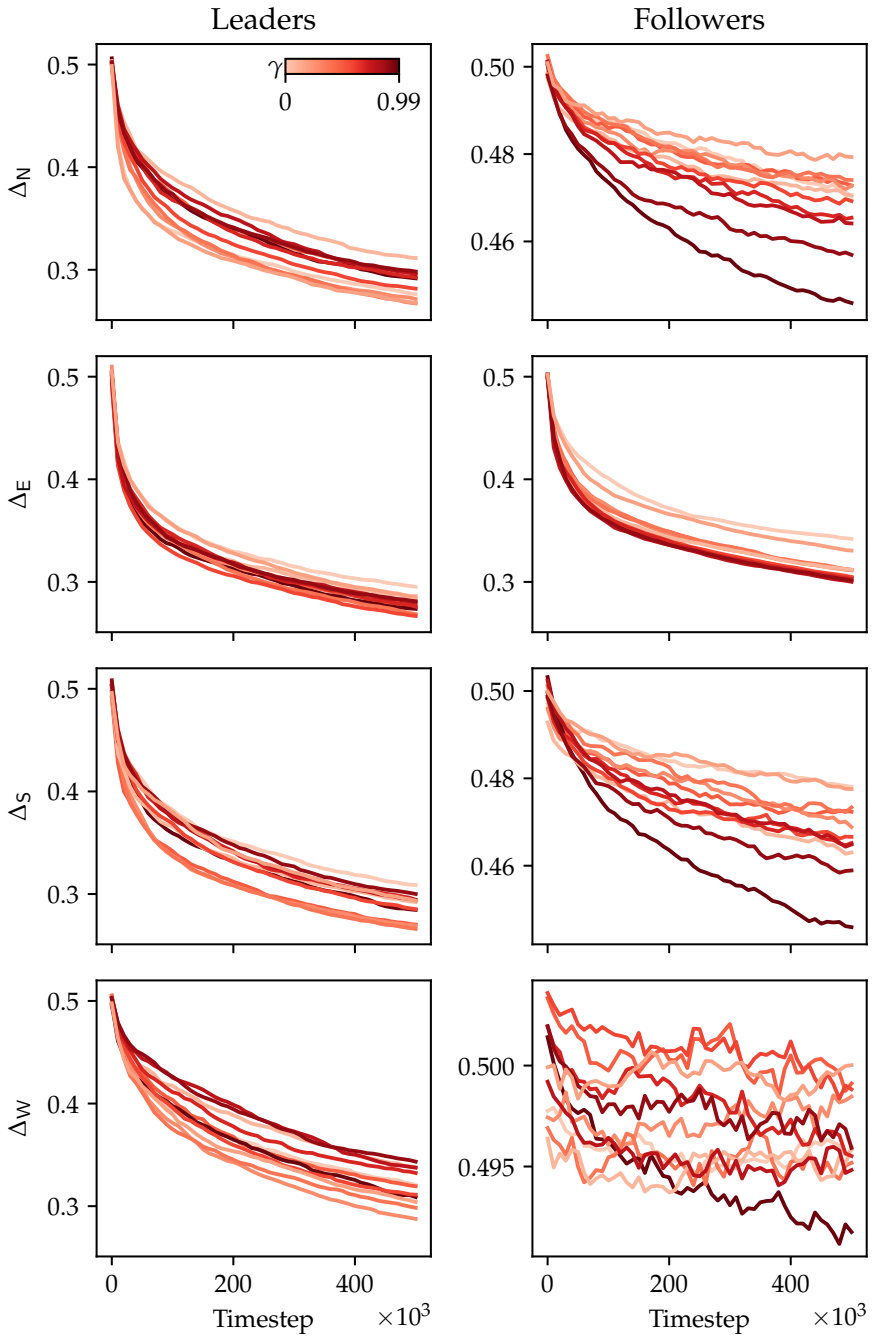

Figure 4.4: *The evolution of different sections of the birds' Q-tables on a longer timescale. We analyzed the Q-tables of the leaders (left) and followers (right) seperately, and for each of those additionally isolated the rows of the Q-tables that correspond to observations in which a majority of the neighbouring birds is flying into one of the cardinal directions* N *(first row),* E *(second row),* S *(third row) and* W*. For each of these we graphed the evolution of the normalized average distance* Δ_x *(with* $x \in \{ \mathsf{N}, \mathsf{E}, \mathsf{S}, \mathsf{W} \}$*) to the opimal policy. We did this with varying* $\gamma \in \{ 0, 0.1, 0.2, ..., 0.8, 0.99 \}$*. Note that we graphed* Δ_N*,* Δ_S *and* Δ_W *of the followers on a different vertical scale than the others.*

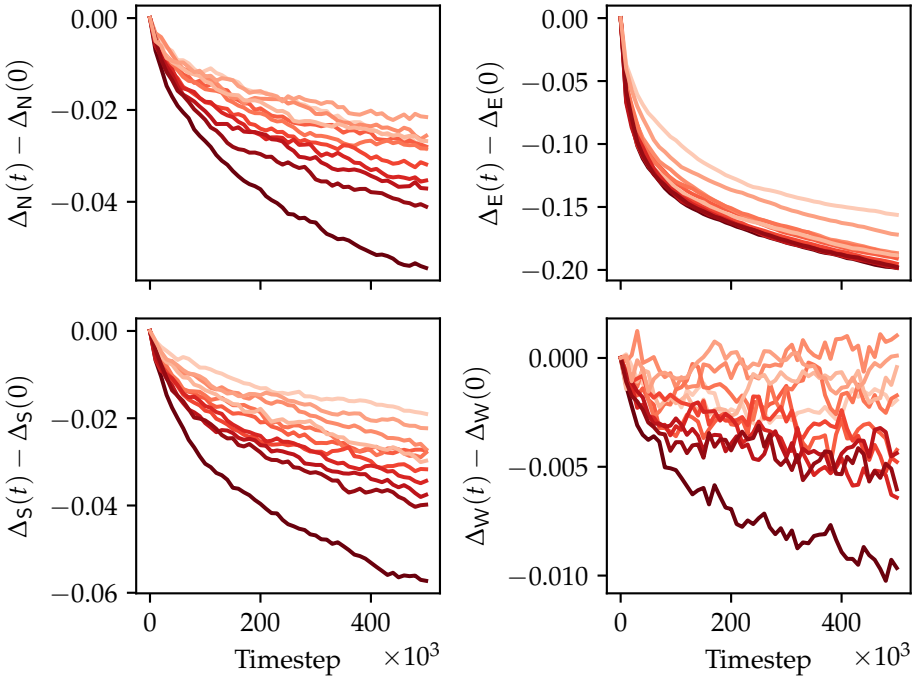

Figure 4.5: *The data of the graphs the right hand side of figure 4.4 repeated (i.e., the graphs of the followers), only we subtracted* $\Delta_x(0)$ *from each run (with* $x \in \{ N, E, S, W \}$*), such that all the curves start from the same point.*

eastward. We concluded that this is the effect of the direct reward signal r_t, since in each of those cases the birds have access to an action $a \in \{ V, I \}$ that directly results in the maximum reward R. However, when a majority of the birds is flying into any other direction $\{ N, S, W \}$, followers can not be directly rewarded at all (both actions V and I result in no reward). In this case we observed that when we increase γ, the distance to the optimal policy is in decreasing significantly faster. This is even more visible when we subtracted the initial value of Δ_x from the simulations ($x \in \{ N, E, S, W \}$), to compensate for the different (random) initializations of the Q-tables. This is shown in figure 4.5 for the Q-tables of the followers.

We concluded from this that, although the direct reward signal dominates in the general policy-making of the birds, long-term decision making is in fact present and visible in cases in which the direct reward signal is zero. More importantly, this long-term decision making does stimulate the followers to choose V more often, i.e., follow the neighbouring birds.

4.2.3 The parameters of the birds and the field: l and d

Finally, we performed runs varying leader fractions l and observation distances d using both the discrete reward system (see figure 4.6 for the results) and the gradient reward system (figure 4.7).

In the measured scope, we observed for the discrete runs that $\langle v \rangle$ only

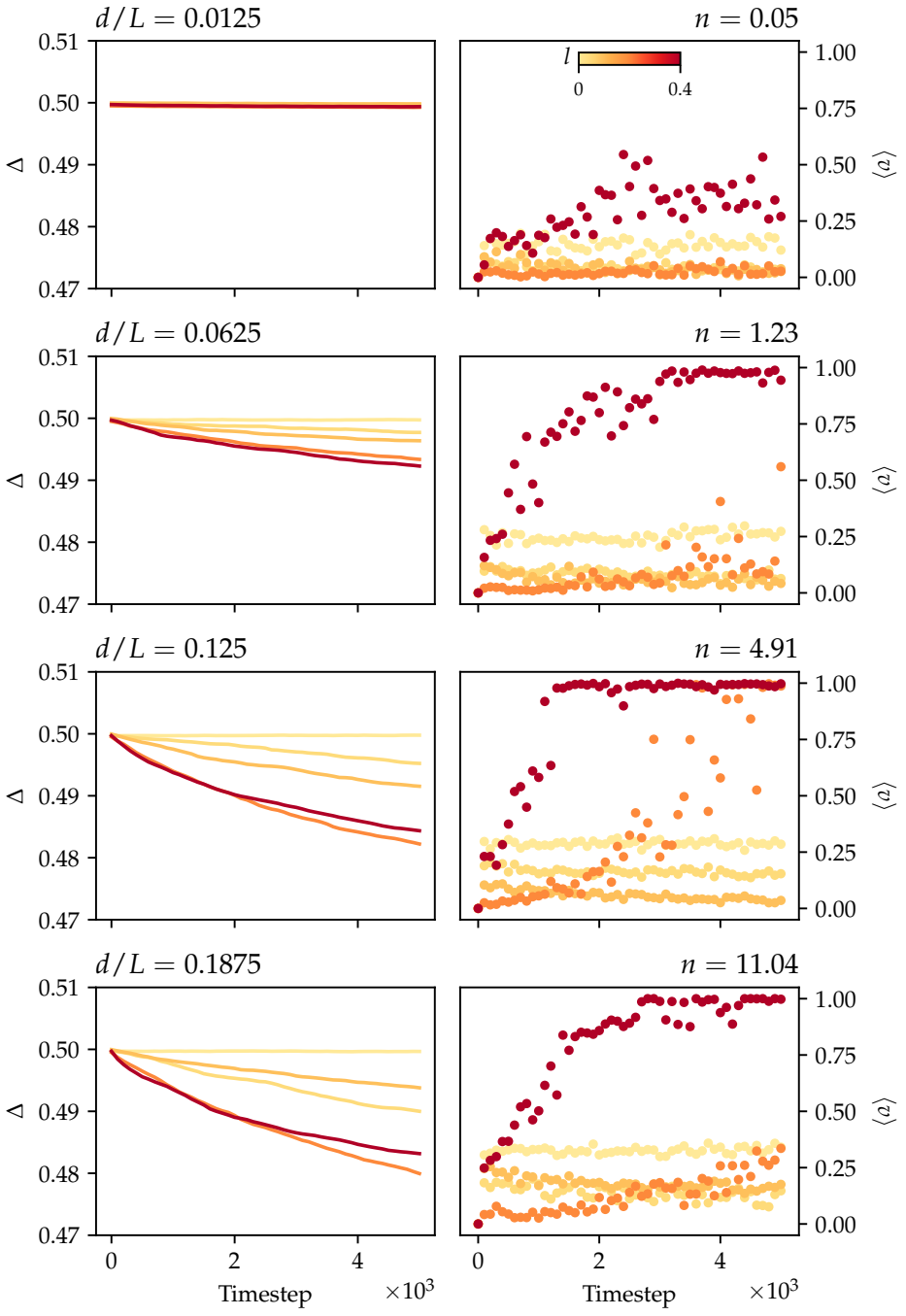

Figure 4.6: *The evolution of* Δ *(left) and* $\langle v \rangle$ *(right) for different leader fractions* $l \in \{ 0, 0.05, 0.1, 0.2, 0.4 \}$ *and observation distances* $d \in \{ 10, 50, 100, 150 \}$*, using the discrete reward system. We grouped all runs in sets of graphs by their observation distance* d *(in increasing value from top to bottom) and colored on a gradient scale based on the leader fraction* l*.*

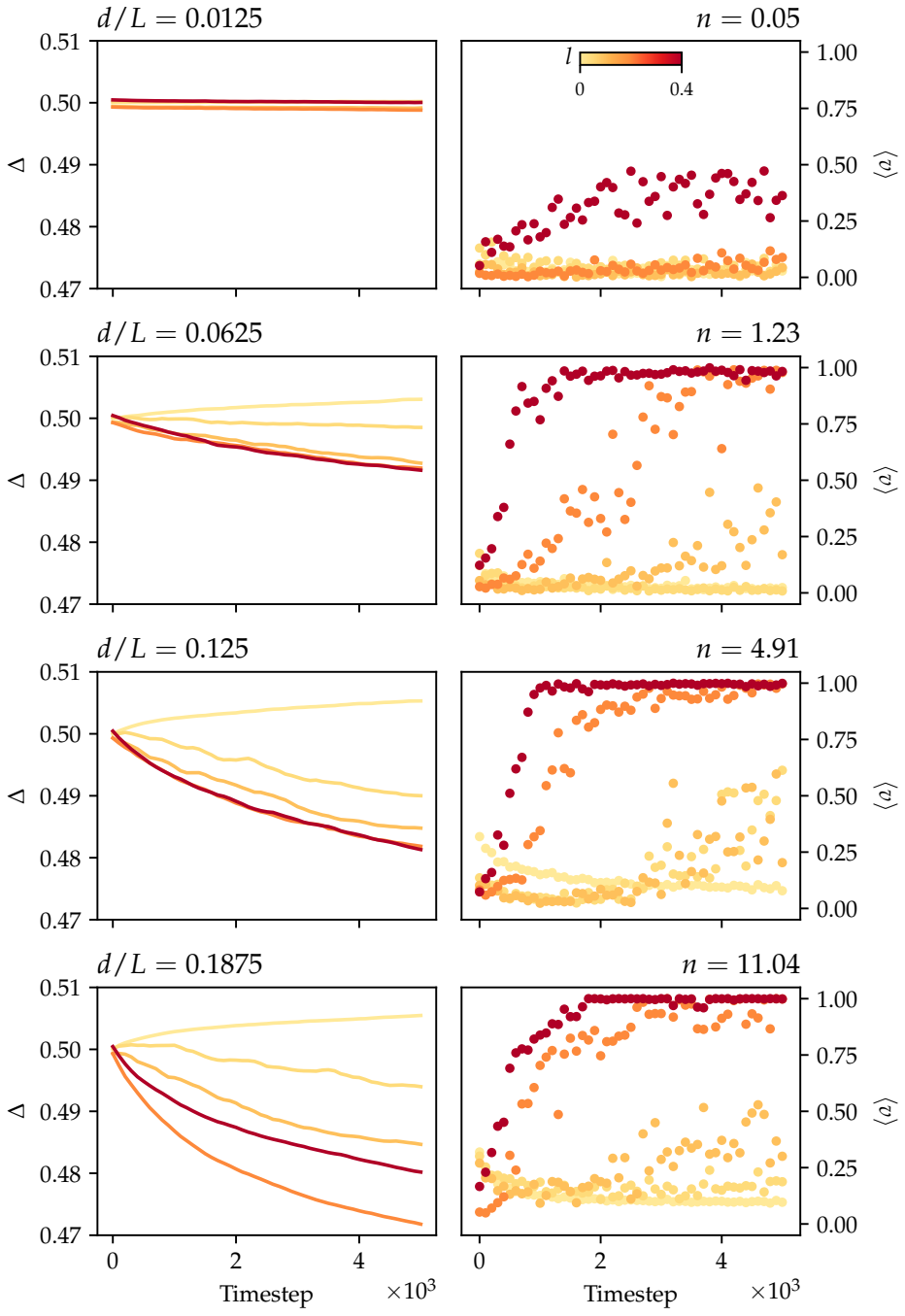

Figure 4.7: *The evolution of Δ (left) and $\langle v \rangle$ (right) for different leader fractions $l \in \{0, 0.05, 0.1, 0.2, 0.4\}$ and observation distances $d \in \{10, 50, 100, 150\}$, using the gradient reward system. We grouped all runs in sets of graphs by their observation distance d (in increasing value from top to bottom) and colored on a gradient scale based on the leader fraction l.*

definitively reaches 1 for observation distances $d \geq 50$ and leader fraction $l = 0.4$. Furthermore, notable differences in the gradient runs compared to the discrete runs are: (1) Δ actually increases for $l = 0$, (2) convergence in general happens earlier for $l = 0.4$ and (3) the threshold for convergence in the measured scope has lowered to $l \geq 0.2$.

That the thresholds for convergence have lowered for the gradient runs and the flock converges earlier, can be explained by the fact that there is more room for the birds to gradually learn to fly eastward. For the discrete reward system, a bird will only be positively rewarded when flying exactly toward $\theta = 0$. This means that a follower for example will only favour to perform V when the discretized average flight direction is exactly $\theta = 0$. However, for the gradient reward system, any movement that has a postive x-component will be positively rewarded. Additionally, movement with a negative x-component will be negatively rewarded, which means for example that birds with an instinct I = W will quickly have low Q-values for action I.

With this data we unambiguously showed that collective motion does emerge from our model. Moreover, in general we can derive the lower bounds $l \geq 0.2$ and $d \geq 50$ as necessary conditions for collective motion. Using the estimate of equation (4.1), this corresponds to an average number of encounters with neigbours of $n \geq 1.23$. Additionally, a minimal fraction of 0.2 of these should be leaders, which roughly corresponds to at least one encounter with a leader every four timesteps.

Chapter 5

Conclusion

In this thesis we developed a model for collective motion using multi-agent reinforcement learning and Q-learning as the learning algorithm, the theory of which we developed in chapter 2. We formulated our particular model in chapter 3, using the language of the flocking behaviour birds. These birds have the option to either fly into an insticnctive direction or act based on a Viscek-type of interaction with their neighbors. The model uses a new type of reward system with orientation-based rewards, meaning that the birds are rewarded maximally when the resulting direction of movement is some predetermined prefered direction. Finally, the model distinghuishes between leaders that instinctively move towards this direction and followers that do not.

With the results obtained in chapter 4, we unambiguously showed that collective motion emerges from this model. First, by tracking the evolution of the Q-tables with Δ and the convergence of the flock with $\langle v \rangle$, we have been able to optimize the learning paramters. In particular our results have shown that optimal learning happens in the regions $0.1 \leq \alpha \leq 0.2$ and $0.1 \leq \epsilon \leq 0.6$. No significant influence of the discount rate γ on the learning process has been found the timescales at which collective behaviour emerges ($\sim 10^3$ timesteps), though simulations with longer timescales ($\sim 10^5$ timesteps) indicate that the learning parameter γ does stimulate followers to follow the flock in cases where this action is not directly rewarded. Since the timescales at which the flock typically converges are much lower than this, we concluded

With learning parameters fixed within the optimal regions, we obtained a couple of quantitative thresholds for the parameters of the system as conditions for this collective motion. In particular, it we observed that collective motion happens for an observation radius $d \geq 50$, corresponding in our model to an average number of encounters with neighbours per timestep $n \geq 1.23$ for each bird. Additionally, of these encounters, we observed a minimal fraction of leaders $l \geq 0.2$ as a second condition for collective motion, suggesting that of these 1.23 encounters every timestep, at least $l \cdot n = 0.246$ per timestep should be leaders, which roughly corre-

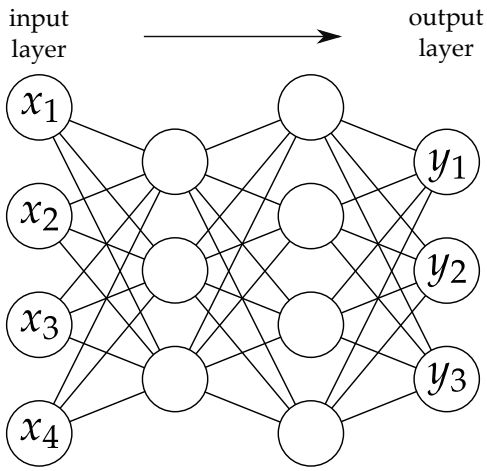

Figure 5.1: *Schematic depiction of a deep neural network. If the input layer represents an observation and the output layer the action policy, then this can function as a replacement for the Q-learning algorithm that is more suited for continuous observation spaces.*

sponds to at least 1 encounter with a leader leader every 4 timesteps.

Note that the original aim of this research has been the *development* of an RL-model explaining collective motion using orientation-based rewards, a type of reward system that has not been found in literature thus far. As such, the emergence of collective motion that has been observed serves as a proof of concept. A consequence of this is that the measured thresholds for convergence that we have observed in chapter 4 are rudimentary lower bounds that can be determined more precisely with additional simulations.

5.1 The implementation of noise: deep Q-learning

The model developed might be interpreted as an RL-extension to the Vicsek model, since a Vicsek-like model can be reobtained when choosing $\mathcal{A} = \{ \mathsf{V} \}$. There are two differences between this model and the Vicsek model, however:

1. The possible flight directions are discretized into a finite subset $\mathcal{D} \subset [0, 2\pi)$.

2. There is no noise term η in this model (cf. equation (1.1)).

The second difference is a direct consequence of the first: since $|\mathcal{D}| = 8$, a deviation from a certain flight direction can only come in discrete steps of $\pi/4$. Therefore continuous changes in the noise distribution, which is a central aspect of the phase transitions observed in the Vicsek model [2], are impossible. The flight directions have in turn been discretized as a direct consequence of Q-learning, which required the observation space to be finite. While it might be theoretically possible to discretize the flight directions more finely and hence approaching continuous behaviour, such

that the Vicsek model becomes a limiting case, this has been shown to be computationally infeasible in section 3.2.

Nevertheless, it might still be interesting to implement some form of noise into the current model. Introducing noise in the Vicsek step (which might correspond for example to a cloudy environment in which the flock of birds navigates), might lead to different strategies . Perhaps even some sort of 'phase transition' might be observed in the model, analogous to that in the classic Vicsek model. Moreover, since a noisy environment might generally slow down the learning process, the typical convergence time might decrease to timescales at which the long-term decision making dictated by γ would be able to play a significant role.

In order to implement noise in the model, we should thus be able to have continuous flight directions in our model, and therefore a continuous observation space. This can be achieved when we replace the Q-learning algorithm (which relies on finite Q-tables) with a *neural network* [34]. A neural network is a collection of 'neurons' that have a value between 0 and 1. These neurons are interconnected and can transmit ther value between each other. Usually this is arranged in propagating layers, meaning there is an *input layer* and an *output layer*, and possibly some *hidden layers* in between (see figure 5.1).[1] The input layer can represent many things, e.g., some sensorial input, or an image, where each individual input neuron might represent the greyvalue of a pixel.

These values are then transmitted into a neuron to which it is connected as a linear transformation $w \cdot x_i + b$, where w and b are called a *weight* and *bias* respectively. This is consequently mapped to the region $[0, 1]$ of a neuron, commonly using the *Sigmoid function* $\sigma(w \cdot x + b)$ given by

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{5.1}$$

All connections between neurons have a weight and bias associated with them, and the challenge in the theory of neural networks is to find the correct set of weights and biases that results in the desired output. A common use case is the field of image recognition, in which the input layer might represent the pixels of a given image that contains a digit, and the output layer might consist of 10 neurons, representing the digits 0–9. The challenge of such a neural network might then be to find the appropriate biases and weights such that the neural network would be able to recognize the digit in the image [34].

An interesting use case for this study in particular would be to combine the theory of reinforcement learning with neural networks into what is called *deep reinforcement learning* [45]. For this, we replace the Q-learning algorithm by such a neural network, where we parametrize an observation as a set of numbers $x_i \in [0, 1]$ and take those as the input of the neural network. The output values $y_i \in [0, 1]$ then represent the action space. For example, each neuron might represent a possible action of the action space,

[1] A neural network with hidden layers is also commonly called a *deep* neural network.

and the one that has the highest output value can be the action that will be performed by the agent.

A whole range of new challenges are associated with the study of neural networks however, which is why we mention the theory here as a suggestion for further improvement. In praticular, there is a lot of freedom in the design of neural networks. Not only is there a freedom of choice in the number of layers of the network and the nuber of neurons per layer, there also exist more advanced types of layers, e.g. *convolutional layers* [34]. Another complication is that the time for a neural network to learn typically is much larger that than that of Q-learning itself, partly because of the much larger size of the parameter space [45].

Despite these challenges, there do exist studies in the field of collective motion that use deep reinforcement learning in their models [14, 16]. An investigation into the role of continuous noise-like order parameters as 'phase transitions' for the agents' policies, has not yet been investigated however. Incorporating the Vicsek model into ours with deep reinforcement learning might open up this possibility.

Bibliography

[1] C. W. Reynolds, "Flocks, Herds and Schools: A Distributed Behavioral Model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87. New York, NY, USA: ACM, 1987, pp. 25–34.

[2] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Particles," *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, Aug. 1995.

[3] T. Vicsek and A. Zafeiris, "Collective motion," *Physics Reports*, vol. 517, no. 3, pp. 71–140, Aug. 2012.

[4] H. Chaté, F. Ginelli, G. Grégoire, F. Peruani, and F. Raynaud, "Modeling collective motion: Variations on the Vicsek model," *The European Physical Journal B*, vol. 64, no. 3, pp. 451–456, Aug. 2008.

[5] V. Mwaffo, R. P. Anderson, and M. Porfiri, "Collective Dynamics in the Vicsek and Vectorial Network Models Beyond Uniform Additive Noise," *Journal of Nonlinear Science*, vol. 25, no. 5, pp. 1053–1076, Oct. 2015.

[6] A. Kudrolli, G. Lumay, D. Volfson, and L. S. Tsimring, "Swarming and Swirling in Self-Propelled Polar Granular Rods," *Physical Review Letters*, vol. 100, no. 5, p. 058001, Feb. 2008.

[7] V. Narayan, N. Menon, and S. Ramaswamy, "Nonequilibrium steady states in a vibrated-rod monolayer: Tetratic, nematic, and smectic correlations," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2006, no. 01, pp. P01 005–P01 005, Jan. 2006.

[8] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, "Machine learning for active matter," *Nature Machine Intelligence*, vol. 2, no. 2, pp. 94–103, Feb. 2020.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018.

[10] H. M. La, R. S. Lim, W. Sheng, and J. Chen, "Cooperative flocking and learning in multi-robot systems for predator avoidance," in *2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, May 2013, pp. 337–342.

[11] C. Chen, Y. Hou, and Y. Ong, "A conceptual modeling of flocking-regulated multi-agent reinforcement learning," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 5256–5262.

[12] D. Gu and E. Yang, "Fuzzy Policy Reinforcement Learning in Cooperative Multi-robot Systems," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 1, pp. 7–22, Jan. 2007.

[13] P. Sunehag, G. Lever, S. Liu, J. Merel, N. Heess, J. Z. Leibo, E. Hughes, T. Eccles, and T. Graepel, "Reinforcement Learning Agents acquire Flocking and Symbiotic Behaviour in Simulated Ecosystems," in *Artificial Life Conference Proceedings*, vol. 31, Jul. 2019, pp. 103–110.

[14] M. Hüttenrauch, A. Šošić, and G. Neumann, "Deep Reinforcement Learning for Swarm Systems," *Journal of Machine Learning Research*, vol. 20, no. 54, pp. 1–31, 2019. [Online]. Available: http://jmlr.org/papers/v20/18-476.html

[15] K. Morihiro, H. Nishimura, T. Isokawa, and N. Matsui, "Learning Grouping and Anti-predator Behaviors for Multi-agent Systems," in *Knowledge-Based Intelligent Information and Engineering Systems*, I. Lovrek, R. J. Howlett, and L. C. Jain, Eds. Berlin, Heidelberg: Springer, 2008, vol. 5178, pp. 426–433.

[16] C. Hahn, T. Phan, T. Gabor, L. Belzner, and C. Linnhoff-Popien, "Emergent Escape-based Flocking behavior using Multi-Agent Reinforcement Learning," in *Artificial Life Conference Proceedings*, vol. 31, Jul. 2019, pp. 598–605.

[17] M. Gazzola, A. A. Tchieu, D. Alexeev, A. de Brauer, and P. Koumoutsakos, "Learning to school in the presence of hydrodynamic interactions," *Journal of Fluid Mechanics*, vol. 789, pp. 726–749, Feb. 2016.

[18] S. Verma, G. Novati, and P. Koumoutsakos, "Efficient collective swimming by harnessing vortices through deep reinforcement learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 23, pp. 5849–5854, Jun. 2018.

[19] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, Mass: A Bradford Book, 2004.

[20] K. Ried, T. Müller, and H. J. Briegel, "Modelling collective motion based on the principle of agency: General framework and the case of marching locusts," *PLOS ONE*, vol. 14, no. 2, p. e0212044, Feb. 2019.

[21] F. Martinez-Gil, M. Lozano, and F. Fernández, "MARL-Ped: A multi-agent reinforcement learning based framework to simulate pedestrian groups," *Simulation Modelling Practice and Theory*, vol. 47, pp. 259–275, Sep. 2014.

[22] ——, "Emergent behaviors and scalability for multi-agent reinforcement learning-based pedestrian models," *Simulation Modelling Practice and Theory*, vol. 74, pp. 117–133, May 2017.

[23] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, Jan. 2016.

[24] S. Muiños-Landin, K. Ghazi-Zahedi, and F. Cichos, "Reinforcement Learning of Artificial Microswimmers," Mar. 2018. [Online]. Available: http://arxiv.org/abs/1803.06425

[25] D. J. Hoare, J. Krause, N. Peuhkuri, and J.-G. J. Godin, "Body size and shoaling in fish," *Journal of Fish Biology*, vol. 57, no. 6, pp. 1351–1366, 2000.

[26] F. E. Fish, "Kinematics of ducklings swimming in formation: Consequences of position," *Journal of Experimental Zoology*, vol. 273, no. 1, pp. 1–11, 1995.

[27] A. P. Tøttrup, K. Rainio, T. Coppack, E. Lehikoinen, C. Rahbek, and K. Thorup, "Local Temperature Fine-Tunes the Timing of Spring Migration in Birds," *Integrative and Comparative Biology*, vol. 50, no. 3, pp. 293–304, Sep. 2010.

[28] T. Alerstam, "Bird Migration Across a Strong Magnetic Anomaly," *Journal of Experimental Biology*, vol. 130, no. 1, pp. 63–86, Jul. 1987. [Online]. Available: https://jeb.biologists.org/content/130/1/63

[29] P. Berthold, *Bird Migration: A General Survey*. Oxford University Press, 2001.

[30] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, 1989.

[31] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[32] R. Bellman, *Applied Dynamic Programming*. Princeton, NJ: Princeton University Press, 1962.

[33] S. M. Ross, *Introduction to Stochastic Dynamic Programming*, ser. Probability and Mathematical Statistics 810776022. New York, N.Y., [etc.]: Academic Press, 1983.

[34] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com

[35] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?" *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, May 2007.

[36] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent Reinforcement Learning: An Overview," in *Innovations in Multi-Agent Systems and Applications - 1*, J. Kacprzyk, D. Srinivasan, and L. C. Jain, Eds. Berlin, Heidelberg: Springer, 2010, vol. 310, pp. 183–221.

[37] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, "Game Theory and Multi-agent Reinforcement Learning," in *Reinforcement Learning*, M. Wiering and M. van Otterlo, Eds. Berlin, Heidelberg: Springer, 2012, vol. 12, pp. 441–470.

[38] J. Hu and M. P. Wellman, "Nash Q-Learning for General-Sum Stochastic Games," *Journal of Machine Learning Research*, vol. 4, no. Nov, pp. 1039–1069, 2003. [Online]. Available: http://www.jmlr.org/papers/v4/hu03a.html

[39] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*, W. W. Cohen and H. Hirsh, Eds. San Francisco (CA): Morgan Kaufmann, Jan. 1994, pp. 157–163.

[40] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents," *arXiv:1802.08757 [cs, math, stat]*, Feb. 2018. [Online]. Available: http://arxiv.org/abs/1802.08757

[41] S. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel, "Emergent Coordination Through Competition," *arXiv:1902.07151 [cs]*, Feb. 2019. [Online]. Available: http://arxiv.org/abs/1902.07151

[42] J. Z. Leibo, J. Perolat, E. Hughes, S. Wheelwright, A. H. Marblestone, E. Duéñez-Guzmán, P. Sunehag, I. Dunning, and T. Graepel, "Malthusian Reinforcement Learning," *arXiv:1812.07019 [cs, q-bio]*, Mar. 2019. [Online]. Available: http://arxiv.org/abs/1812.07019

[43] P. Muller, S. Omidshafiei, M. Rowland, K. Tuyls, J. Perolat, S. Liu, D. Hennes, L. Marris, M. Lanctot, E. Hughes, Z. Wang, G. Lever, N. Heess, T. Graepel, and R. Munos, "A Generalized Training Approach for Multiagent Learning," *arXiv:1909.12823 [cs]*, Sep. 2019. [Online]. Available: http://arxiv.org/abs/1909.12823

[44] "Scipy.spatial.KDTree – SciPy v1.5.0 Reference Guide." [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html

[45] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski,

S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.