

# Can predatory pressure explain flocking behavior?

Jordy de Lange

## Supervisors

Prof. dr. J. Kosinka      Rijksuniversiteit Groningen  
Prof. dr. L.C. Verbrugge      Rijksuniversiteit Groningen

Rijksuniversiteit Groningen  
August 2023



# Summary

Flocking-like behavior occurs in many species. Swarms of birds, schools of fish, and herds of large herbivores all display behavior that can be categorized as flocking. Research has been done not only into how this flocking works, but also to find out why flocking is prevalent. The fact that flocking-like behavior occurs in many very diverse environments, suggests that it is a solution to a common problem. Candidate problems here are the need to find food and the need for social contact. Another theory is that flocking-like behavior helps in evading predators. The theory behind this is that it is harder for a predator to attack a large, dense group of prey animals than single prey animals. This “confusion” effect can be further enhanced by the prey flock displaying visually complex movement patterns.

Carsten Hahn, Thomy Phan, Thomas Gabor, Lenz Belzner and Claudia Linnhoff-Popien, at the 2019 Conference on Artificial Life, presented research where agents learned to evade a predator in an empty toroidal 2D environment through Q-Learning. The agents could perceive the location and orientation of the five nearest neighbouring agents as well as the predator at all times. The predator loses hunting efficiency when more than one agent is close to it, mimicking the increased required effort seen in real predators when attacking a flock of prey animals. Their results showed the agents adopting flocking-like behavior. Agents randomly initialized throughout the environment would quickly group together and stay together.

In order to try and further strengthen the evidence for predation being a possible driver behind the widespread adoption of flocking-like behavior, this thesis attempts to extend the findings of Hahn et al. to a 3D environment. This was done by first recreating the experiment performed by Hahn et al. This 2D recreation model was trained and tested under a variety of parameter conditions. Among the varied parameters was the number of observable neighbours. Because flocking behavior requires coordination with neighbouring flockmates, all real flocking-like behavior should only be present in observable neighbour conditions. Next, a 3D version of the same model was built in order to see if any emergent flocking-like behavior in the 2D model still emerges in a 3D environment.

Unfortunately, the recreation of the results obtained by Hahn et al. in 2D was unsuccessful. The agents would learn to evade the predator, but no real flocking-like behavior could be observed. Any behavior observed which did resemble flocking was explainable by the agents simply fleeing the predator while close to other agents, as this behavior visually resembling flocking was also observed in conditions where agents could not see neighbouring agents.

A second smaller experiment where the predator pauses after eating an agent was performed. This experiment aimed to give the agents an advantage by allowing time for the agents to reform any groups that had had been scattered by the predator. This did not change the behavior the agents displayed.

Although no real flocking-like behavior emerged in the 2D experiments, the 3D model was still trained and tested. This, as expected, also did not result in flocking-like behavior. The agents displayed random looking behavior, not convincingly reacting to the predator when it would get close.

Potential reasons for the failure of the replication are errors in the code responsible for training the agents, undetected differences between the behavior of the predator Hahn et al. used and the one used in the recreation attempt, or significant differences in model parameters. While code used by Hahn et al. was referenced in the building of the recreation model, the referenced code was not complete and several assumptions about model parameters had to be made.

Potential future research could look further into the use of different learning algorithms like NEAT and DDPG. Pretraining the agents is also an option. One way this could be done is to first train the agents to flock and group before introducing a predator. Another possibility is to train the agents to evade the predator before introducing other agents into the model.



# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Background</b>	<b>9</b>
2.1	Flocking in nature . . . . .	9
2.1.1	Flocking and predation . . . . .	10
2.2	Boids . . . . .	11
2.2.1	Trained Boids . . . . .	12
2.3	Deep Q-Learning . . . . .	13
2.3.1	Q-Learning . . . . .	13
2.3.2	Deep Q-Learning . . . . .	13
2.3.3	Experience Replay . . . . .	14
2.3.4	Exploration strategy . . . . .	14
<b>3</b>	<b>Problem statement and overview</b>	<b>17</b>
<b>4</b>	<b>Methods</b>	<b>19</b>
4.1	The general flocking model . . . . .	19
4.2	Boids . . . . .	20
4.2.1	General characteristics . . . . .	20
4.2.2	Decision making . . . . .	20
4.2.3	Being eaten . . . . .	21
4.3	Predator . . . . .	22
4.3.1	General characteristics . . . . .	22
4.3.2	Target selection . . . . .	22
4.3.3	Chasing the target . . . . .	22
4.3.4	Eating the target . . . . .	23
4.4	Learning . . . . .	24
4.4.1	Why Deep Q-learning? . . . . .	24
4.4.2	Training the boids . . . . .	24
4.4.3	The flow of training . . . . .	24
4.5	Statistical analysis . . . . .	26
4.5.1	Objective of the analysis . . . . .	26
4.5.2	Measurements . . . . .	26
4.5.3	The performed analyses . . . . .	28
4.6	Code . . . . .	28
<b>5</b>	<b>Experiments</b>	<b>29</b>
5.1	General experiment design . . . . .	29
5.2	2D experiment 1 . . . . .	29
5.2.1	Parameters . . . . .	30
5.2.2	Analysis . . . . .	32
5.3	2D experiment 2 . . . . .	33
5.3.1	Parameters . . . . .	33

5.3.2	Analysis . . . . .	33
5.4	3D experiment . . . . .	34
5.4.1	Parameters . . . . .	34
5.4.2	Analysis . . . . .	34
<b>6</b>	<b>Results</b>	<b>35</b>
6.1	2D experiment 1 . . . . .	35
6.1.1	2D experiment 1 boid behavior . . . . .	35
6.1.2	2D experiment 1 results . . . . .	38
6.1.3	2D experiment 1 analysis . . . . .	40
6.1.4	2D experiment 1 validity . . . . .	43
6.2	2D experiment 2 . . . . .	45
6.2.1	2D experiment 2 boid behavior . . . . .	45
6.3	3D experiment . . . . .	47
6.3.1	3D experiment boid behavior . . . . .	47
<b>7</b>	<b>Discussion</b>	<b>49</b>
7.1	Interpreting the results . . . . .	49
7.1.1	2D experiment 1 . . . . .	49
7.1.2	2D experiment 2 . . . . .	50
7.1.3	3D experiment . . . . .	50
7.2	Reviewing the study . . . . .	50
7.2.1	Implementation . . . . .	50
7.2.2	Process . . . . .	50
7.3	Potential changes to the methods . . . . .	51
7.4	Future research . . . . .	51
<b>8</b>	<b>Conclusion</b>	<b>53</b>
<b>A</b>	<b>Full results 2D experiment 1</b>	<b>59</b>

# Chapter 1

## Introduction

Flocking is a recurring phenomenon in nature. Examples one could think of are swarms of starlings evading diving falcons, giant schools of fish dodging attacks from hungry tuna and herds of wildebeest navigating across large distances. The pervasiveness of flocking-like behaviour suggests that it offers some concrete advantages.

Among these advantages are a dilution effect, a confusion effect and a vigilance effect [7]. By *dilution effect* it is meant that as a flock grows in size, the less likely any single member of that flock is to fall prey to a predator attacking the flock. Additionally, being in a large group facilitates movement patterns that can confuse the predator [16]. This, combined with the fact that tracking a single target among many is harder than tracking a target in isolation [19], is responsible for the *confusion effect*. Finally, being part of a group comes with the advantage that there are more of you to spot potential dangers; the *vigilance effect*.

Another possible advantage of flocking is reduced energy expenditure. Ibis flying in a V-formation gain aerodynamic benefits from flying in such an arrangement [33]. Similar effects have been observed in pelicans flying close to the water surface [42]. However, flying in a flock does not always lead to such benefits. Flying in more unorganised flocks, especially when members of the flock fly directly behind other flockmates, can actually cost more effort than flying solo [41].

Flocking or similar behavior occurring across a large range of species and environments strongly suggests that the behaviour convergently evolved in multiple independent species. Generally, this occurs in nature when there is a common problem that needs a solution. An example of such convergent evolution is the body plan of sharks, dolphins and ichthyosaurs (see Figure 1.1). Even though one is a fish, one is a mammal and one was a reptile, they evolved nearly identical body plans. Apparently, this body plan is an efficient solution to the problems faced by a relatively large aquatic predator.

This line of reasoning can be extended to flocking; what common problem is efficiently solved by flocking-



Figure 1.1: Convergent evolution in sharks, dolphins and ichthyosaurs.

like behavior? Candidates here are the search for food or the need for social contact with others. Another potential problem, as mentioned above, is predation. Many animals that flock face frequent danger from predators. This thesis attempts to determine if predatory pressure could be a driving force behind the evolution of flocking behaviour.

Earlier research has shown that in a simplified 2-dimensional model, the presence of a predator can induce flocking behaviour in agents trying to not get eaten [13]. Through the use of Deep Q-Learning, agents are trained to survive in an empty and toroidal 2-D world populated by multiple other identical agents and a single predator. This predator has hard-coded behavior. It loses hunting efficiency when multiple agents are close to it, paralleling the confusion effect observed in nature. The flocking behaviour was achieved without any reward being directly given for flocking behaviour. The only objective of the agents was to not be eaten by the predator.

This thesis first seeks to replicate results of the study mentioned above using a similar 2D model. After this, the same concept was tried in 3 dimensions using a 3D model. It stands to reason that flocking in 3D is more complex than flocking in 2D from the point of view of the flocking agents. This increase in complexity could be a step too far, making flocking no longer an effective and efficient solution to the predation problem.

Theory and research related to flocking and models of flocking are presented in Chapter 2. A concise overview of what needs to be done to accomplish the goals of this thesis is given in Chapter 3. This is followed by an explanation of the methods used in Chapter 4, with a description of the experiments performed in Chapter 5. The results are presented in Chapter 6 and further discussed in Chapter 7. Finally, the conclusion is given in Chapter 8.



# Chapter 2

## Background

This chapter is spent reviewing literature related to flocking in nature, computer models of flocking and flocking by learning agents. First, in Section 2.1, some light is shed on flocking in nature. Next, in Section 2.2, computer models of flocking featuring programmed agents are discussed. This is followed by models that feature learning agents. Finally, in Section 2.3, the method of Deep Q-Learning used for the models of this thesis is explored.

### 2.1 Flocking in nature

The exact workings of flocking have long interested experts. Very early theories proposed some sort of group mind to be governing the process [35]. This theory suggested that the nervous systems of the individuals making up a flock are somehow connected through a type of telepathy, or at least that some sort of communication is taking place. A more popular early theory assumed that there was a leader in each flock. Actually identifying the leader of any given flock proved impossible though, suggesting that this also was not the mechanism enabling flocking [17]. In 1980, J.M. Davis proposed that flocking is enabled by a process he called ‘self-generated synchrony’ [8]. He speculated that through a democracy-like process, the number of flock members displaying a specific behaviour can reach a threshold, upon which the rest of the flock follows.

A theory of how grouping in animals evolved was given in 1971 by Hamilton [14]. Through a series of rudimentary computer models, it was shown that the presence of a predator can lead to a bunching up of prey animals. A prey species was dispersed throughout a space randomly. When the model is run, each prey agent tries to minimize its own “domain of danger”. This is the area that arises from the assumption that the predator, which can suddenly appear anywhere in the environment, will always go for the closest prey agent. The domain of danger of a prey agent therefore is the region of the environment where it itself is the closest prey agent, if the predator were to appear in that area. If the model is run and each prey agent moves around in order to minimize its own domain of danger, dense groups of agents start forming. This occurs because the only way for an agents to shrink its own domain of danger is to get closer to other prey agents.

This clumping has been seen in nature as well. Fish have been observed grouping together in such tight schools that both birds on the surface of the water as well as sharks below the surface could eat mouthfulls of fish at a time [5]. Similarly, an account has been presented where a group of striped mullet swarmed in a group so dense that it appeared to be a small island, with fish being pushed up and out of the water entirely [36]. When the observer’s boat approached, the grouping dove down about a meter. They did not flee, however, and it was possible to easily catch multiple fish at a time by simply thrusting a spear into the mass of mullets.

The model of grouping has been refined over time. Through analysis of video footage of shoals of fish, evidence was found for “attraction” and “repulsion” zones around individual fish [39], with the attraction zone being larger than the repulsion zone. The observed fish seem to be attracted to other fish in their attraction zone. Once the fish get close enough to be in one another’s repulsion zone, they will move away from each other. Additionally, there appears to be a neutral zone between the attraction and repulsion threshold distances, where no attraction or repulsion reliably takes place.

This theory of attraction and repulsion seems to be confirmed by Wood and Ackland [43]. They created a model with agents that had programmed responses to other agents within their repulsion and attraction radii. The optimal sizes of these radii were determined through evolution. Fitness was determined through the addition of a predator to the environment. This resulted in a model that produced behaviour very similar to schooling fish and flocking birds. It was also able to recreate patterns observed in real animals, like milling. This suggests that real animals do in fact utilize similar repulsion and attraction thresholds in order to form schools and flocks.

### 2.1.1 Flocking and predation

As early as 1922 [27], researchers thought about why many different types of animals in many different environments display some sort of grouping, flocking or schooling behaviour. Robert Miller, during his studies of the American bushtit, observed the behavior of the tits when a hawk or another raptor approached. The tits are foraging in a tree, when one of them spots the predator. The spotter lets out a warning cry, upon which the entire group becomes motionless. In addition, they all start producing a quavering note which is hard to localize. This tends to result in the predator failing to catch one of the tits or just passing without attempting to catch one.

Added to this observation is a report of a loon repeatedly diving through a school of sardines, but somehow failing to catch one after several attempts [1]. These two cases lead to the suggestion that confusing a potential predator is one possible goal of being in a flock or school, although the evidence for this claim is not entirely clear-cut [22].

A study that looked at the predation behaviour of sticklebacks does suggest that attacking a larger group of water fleas costs more mental resources than a small group [26]. When a hungry stickleback is let into a reservoir with water fleas, it tends to attack areas with high flea density. When presented with a stimulus mimicking an avian predator before entering the reservoir, the stickleback's preferences changes to attacking less dense regions. Important to note here is that the feeding rate of the stickleback is higher when it attacks large, dense groups of water fleas than when it attacks small, less dense groups. The fact that it can be coerced by the presence of a predator stimulus into attacking lower density groups leads to the speculation that attacking a high-density area costs more of the stickleback's attention. The underlying idea here is that attacking a large group leads to lessened awareness and therefore a heightened risk of being ambushed by an unseen predator. When the presence of a predator is suggested, the stickleback chooses the less risky option of attacking a lower-density area.

In 1979, Gillett et al. [12] examined the effect of prey group size on the time it takes for a predator to catch a single locust nymph. The role of predator was played by both a lizard and a human. In all cases, the time to catch decreased as the group size of locusts increased, up to a point. Once the group size of the locusts crossed 50, the time to catch increased for both the humans and the lizard. When the group size crossed 100, the lizard failed to catch any and tried to escape the feeding environment. This suggests that at some point (around 50 here), the number of prey becomes too much for the predator to effectively process.

A relation between predator accuracy and prey flock size and density is suggested by a study where humans were tasked with hunting down a marked individual in a realistically simulated starling flock [19]. The participants were tasked with navigating a predator towards a single simulated bird, marked by a crosshair. The participants had 30 seconds to catch the target, and they were told to hit the "catch" button when they thought they were as close as possible. At that point, the distance between the predator and the prey was recorded as the targeting error. While the time until the catch button was hit decreased with increasing flock size, the targeting error grew significantly with both increasing flock size and increasing flock density. This again suggests that attacking larger and/or more dense flocks is harder for a predator.

Still, even with the above experiments showing or suggesting some benefit for the prey of being in large groups, it is hard to determine that this is definitely due to the predator getting confused. Nevertheless, there is likely a causal link between predation and flocking. A paper by Magurran and Pitcher [24] describes the behavior of minnows when attacked by a pike. Several small loose shoals of minnows are observed to come together into a tight school when a pike is introduced into their tank, if the number of minnows is high enough. The likelihood of forming such a tight school increased with increasing number of minnows. When the school was attacked, it would evade the pike through complex patterns. This suggests that there is more to predator evasion than purely the movement of individual fish, as all avoiding trajectories taken

by a member of a school can also be taken by a single fish in isolation. The element a singular fish cannot replicate is the presence of many more fish, suggesting that at least part of the advantage of a school is the overwhelming number of potential targets it affords the predator.

The evidence for a link between flocking and predation is reinforced by studies that compare flock sizes between bird populations in areas with high and low predatory pressure. It has been found that populations on islands with low predatory pressure (compared to the mainland) feature either a similar or a lower mean and maximum flock size when compared to their counterparts on the mainland [2]. This paper gathered data on 47 species of bird from a total of 22 different islands. Possible alternative explanatory factors like a reduced population density on islands failed to explain the observed effect. A similar effect was observed between two groups of starlings, each roosting in a different area [6]. The group roosting in the area featuring a lower predatory pressure featured a higher frequency of loose or low density flocking formations, with the higher predatory pressure group featuring a higher frequency of large and high density formations. Additionally, when peregrine falcons attacked each population, their success rate was significantly higher when attacking the low predatory risk group. This again suggests that predation coaxes prey into flocking behaviour, and that this flocking behaviour is an effective defense against predation.

## 2.2 Boids

One of the first attempts to simulate flocking was undertaken by Reynolds and led to a model with a group of flocking agents he called boids [34]. The idea behind the used approach was to simulate the forces acting upon a member of a real school or flock. The simulated forces are collision avoidance, velocity matching and flock centering. In order to combine these possibly competing forces into a single instruction for a boid, a weighted average of all three can be used.

Collision avoidance prevents boids from overlapping in position. In real flocks, members have to maintain a safe distance from their neighbours in order to prevent collisions. In practice, this establishes a safe minimal distance from all immediate neighbors. This distance is then maintained through the second force, velocity matching. ‘Velocity’ here consists of two components; speed and direction. By matching both the speed and direction of travel with all nearby neighbours, a uniform, ‘polarized’ flock can be formed.

The final force is flock centering. This makes each boid want to fly towards the average position of nearby boids. This force keeps the flock together, preventing boids from drifting away from their flockmates. If a boid does find itself moving away from the flock, flock centering will push it back towards the flock again. Subsequently, collision avoidance prevents the boid from crashing into the flock and velocity matching ensures it will travel in unison with its flockmates from now on.

Obstacle avoidance is also a possibility in this flocking model. This can be achieved by having each boid regularly check if they are about to run into anything. If it is determined that a surface will indeed be hit, a point one body length away from the object to be hit will be determined and the boid will be made to aim for that point. This impulse can be made to override the three fundamental forces in order to ensure that no obstacles are hit.

In 1990 Heppner and Grenander [17] published an alternative flocking model. This model relies upon four forces governing the behaviour of each individual in a flock flying over its roost. The four functions are a homing force, which attracts individuals towards the roosting area; an interaction force, which coaxes individuals towards maintaining a certain distance from their neighbours; a random force, which simulates gusts of wind and other local disturbances; and finally a velocity force, which encourages an individual to move at a set speed if one of the other forces previously accelerated or decelerated the individual.

The resulting model supports several types of behaviour that are also observed in real flocks. The artificial flock is able to mill over its roost, it is possible for the flock to break up and reform, and for chaotic flight to develop into orderly milling and the other way around.

Another possible approach to flocking is one rooted in real-world physics. An example of a model that utilizes this method is StarDisplay [18]. This model has been used to explore numerous facets of starling flocking, from patterns that confuse predators [15] to the effectiveness of falcon attacks [28]. It has also been used to demonstrate the utility of flocking against predation [19].

### 2.2.1 Trained Boids

Instead of writing rules for agents to follow in order to create flocking, it is also possible to obtain flocking behaviour through learning agents. A straightforward method, originally presented by Morihiro et al., would be to have the agents learn to follow the three rules the original boids [34] utilised using Q-Learning [30] (for more on Q-Learning see Section 2.3.1). In order to accomplish this, the agents could select any of four actions (Table 2.1).

Each step each agent  $i$  selects another agent  $j$  in order to learn, where  $i$  and  $j$  are from the set of all agents and  $i \neq j$ . Depending on the distance between  $i$  and  $j$ , a reward value is chosen.

Action	Description
$a_1$	Move closer to agent $j$
$a_2$	Turn in order to match agent $j$ 's heading
$a_3$	Turn in order to head in the opposite direction of agent $j$ 's heading
$a_4$	Move further away from agent $j$

Table 2.1: Possible actions in the model of Morihiro et al. [30]

When  $i$  is too close to  $j$ , a positive reward will only be given if the action selected by  $i$  is to move away from  $j$ . If  $i$  is too far away from  $j$ , a positive reward can only be obtained by moving closer. If the distance between the two agents is not too small and not too large, a positive reward is given if  $i$  turns to match  $j$ 's heading. In all of the previous cases, a negative reward is given if the right action has not been selected. If  $i$  and  $j$  are a large distance away from one another, no reward or punishment is given at all.

This method also supports the addition of a predator to the simulation. In the case that for any agent  $i$  the predator is chosen as agent  $j$ , moving away from the predator will net a positive reward with all other actions giving a negative reward. Similarly, if  $j$  is just another agent, if the predator is far enough away no reward or punishment will be given no matter the action chosen by  $i$ .

The resulting model is able to convincingly flock and evade a predator. Each agent, however, is still essentially following the rules set originally by Reynolds. Although the agents did learn the rules instead of being programmed to follow them, the flocking behaviour did not emerge as the solution to a problem the agents were facing.

Training a flock of boids does raise an issue: How does one train an entire flock of agents based on actions each individual takes? A solution is proposed in 2016 by Maxim Egorov [9]. The proposed method only actually trains one single agent. While this agent has its decision making process trained, all other agents have their decision-making process frozen. After a set interval, the updated decision making process of the single trained agent is shared with all other agents, after which this repeats.

This allows for all agents to decide on actions for themselves, but it also allows for a gradual learning process in which all agents utilise the same strategy. This last point is especially important when trying to induce some sort of emergent behavior, like flocking. If not all agents are participating in the emergent behavior, the few that are will likely learn it is not an effective strategy. By copying the trained-decision making process to all other agents, cooperation can be ensured.

This method of training a population of boids by only training one of them was used by Hahn et al. in order to produce emergent flocking behavior in a group of boids through the use of Deep Q-Learning [13]. During training, ten boids were put in an empty 2D toroidal environment, together with a predator. The trained boid here was given a small positive reward each timestep it managed to not be eaten by a predator, and a big negative reward if it was eaten. The predator worked by picking a target boid and chasing it down. To this was added a simulation of the confusion effect; if there were multiple boids in close proximity to the predator, it was made to needlessly switch targets frequently. This reduced the hunting efficiency of the predator when the boids were grouping together. The result was a group of boids that reliably flocked together and tried to stay together. The methods of this paper are discussed further in Chapter 4, Methods, because this thesis, among other things, attempts to recreate it.

## 2.3 Deep Q-Learning

This section explores the workings of Q-Learning and Deep Q-Learning.

### 2.3.1 Q-Learning

Q-Learning has its roots in Bellman’s 1957 book “Dynamic Programming” [3]. In this book, he describes a method for selecting actions in order to influence a system that changes state based on the chosen actions. The innovation here is that this method will ‘learn’ in order to maximise a reward that accompanies every encountered state-action pair. This allows this method to learn to function in a wide range of applications.

Given a set of system states  $S$  and a set of possible actions  $A$ , the system transitions from state to state through actions performed by some agent. This agent selects which action  $a \in A$  should be taken in a given state  $s \in S$  through a function  $Q(s, a)$  that determines the quality of each state-action combination  $(s, a)$ . Once an action is taken, the newly transitioned to state is valued by a numerical reward  $r$ . This reward can be positive or negative. If the new state is in line with what the programmer wants the system to achieve, a positive reward should be given. In the case that the taken action is detrimental and leads to an unwanted state, a negative reward should be given.

In the most basic implementation of Q-Learning, the action valuation function  $Q$  takes the shape of a table. This table is known as the *q-table* and it contains *q-values* for every possible action in every possible state. Actions are selected by picking the action  $a$  with the highest *q-value* for the current state  $s$ . The taken action is then responded to with a reward and a new state, upon which the process repeats. This leads to a sequence of triples of states, taken actions, and received rewards  $\{(s_0, a_0, r_0), (s_1, a_1, r_1), (s_2, a_2, r_2) \dots\}$ . The received reward  $r_t$  is then used to update the *q-value* of the preceding state-action  $(s_t, a_t)$  combination.

Updating the *q-table* is done through the use of the Bellman Equation.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \quad (2.1)$$

The equation works as follows. The *q-value* of the current state-action pair  $Q(s_t, a_t)$  is updated to the value it already had, plus a second term which is multiplied by the learning rate  $\alpha$ . By varying  $\alpha$ , generally between 0 and 1, the speed of the training process can be adjusted.

The second term, which is multiplied by  $\alpha$ , consist of two parts; the received reward  $r_t$ , and the second part which is multiplied by the discount factor  $\gamma$ . This second part accounts for any future rewards. This is done by adding the maximum *q-value* possible in the next state  $s_{t+1}$  to the *q-value* of the current state-action pair  $Q(s_t, a_t)$ . The maximum *q-value* possible in the next state  $s_{t+1}$  itself again includes the *q-value* possible in the next-next state  $s_{t+2}$ , and so on. This way possible future rewards are propagated backwards through the states leading up to them. By tuning the discount factor  $\gamma$ , the weight of future rewards can be modified.  $\gamma$  is generally set somewhere between 0 (future rewards are not considered at all) and 1 (future rewards are just as valuable as current rewards). Higher values for  $\gamma$  tend to result in agents more willing to invest in future rewards.

The agent must fully explore the possible state-space in order to eventually construct the most accurate *q-table* possible. If during the training process actions are always selected by their *q-value*, it is likely that certain state-regions will not be explored thoroughly enough. As soon as a specific action in a specific state gets the highest *q-value* out of all possible actions in that state, it will always be picked.

One solution to this problem is through the use of the  $\epsilon$ -greedy policy during training. This policy works by selecting a random action with probability  $\epsilon$  and otherwise selecting the action with the highest *q-value*. By choosing  $\epsilon$ , the balance between exploration and exploitation of the environment can be fine-tuned. This is explained further in Section 2.3.4.

### 2.3.2 Deep Q-Learning

While Q-Learning as described in Section 2.3.1 can be very effective, it has its limitations. One of these is the fact that the action valuation function  $Q(s, a)$  is implemented through a table. This becomes a problem when the number of states grows, as this requires a larger and larger table. The context of this thesis, flocking, would require an impractically large table in order to describe every possible observable state.

It is possible to circumvent this issue by replacing the  $q$ -table with an approximation of  $Q(s, a)$  of some sort. A deep neural network can be trained for this task. If an effective enough approximation can be constructed this way, it enables learning in far more complex environments. The trained agent will be better at generalizing what it has learned to new situations, making it more effective at learning to function within environments with many moving parts.

An early example of this method is given by Mnih et al. [29], where Deep Q-Learning is used to play a variety of Atari 2600 games. Earlier attempts to utilize Deep Q-Learning in order to train agents to perform well in tasks were generally unsuccessful or required learning setups or preprocessing that was specific to the task at hand. Mnih et al. were able to devise an architecture that was able to learn to play seven different Atari games well without any game-specific changes. The major innovation provided by Mnih et al. was the use of experience replay [23], which will be explained further in the next section.

### 2.3.3 Experience Replay

Assume we have an agent we are training through Deep Q-Learning. In order to collect the needed training data, the agent is released into an environment where it can make observations, take actions and receive rewards. A naive way to train the agent would be to use each observation-action-reward triple for updating the agent's network as they come in. This means that the triples used to train the agent are ordered temporally.

While this can work, it also has its problems. If we assume that the agent gradually makes its way through the state space, the naive method means that we repeatedly train the agent on possibly quite similar states. This can lead to overfitting to the local states.

This is a recurring problem, because once the agent traversed the state space more, we then again fit the agent very specifically to the now local states. This repeated overfitting to local states means that it is possible that beneficial changes to the network are removed later in the training process, because the states where those changes are beneficial have not been encountered in a long time.

To illustrate this concept further, consider the following: When training a network to be able to classify images of dogs and cats, one does not first feed the network all the dogs and then the cats. The way to train such a network is to present stimuli randomly selected from the entire training set, ensuring a good mix between the two classes. Randomly selecting from the entire training set is hard to do in a reinforcement learning case, because the training set is gradually “generated” as the agent picks actions and transitions to new states.

A way to approximate this random selection is to make use of experience replay [23]. This works by keeping track of a memory for the agent. Each time a reward is given for a particular action from a particular state, this combination is stored in memory. Then, when it is time to update the network, the network is not updated with the most recent action-state-reward combinations, but instead with a random selection from memory. This memory has a fixed size, meaning that if it is filled up, new action-state-reward combinations overwrite old ones.

By randomly selecting action-state-reward combinations in this manner, we prevent the repeated overfitting to local state spaces of the naive approach, by breaking any relation (temporal, causal or otherwise) between consecutive network updates. This ensures a more gradual training across the entire state space, leading to better performance.

### 2.3.4 Exploration strategy

The described selection of actions in Deep Q-Learning has a flaw in practice [40]; in training for tasks where it is not immediately obvious how to get positive rewards efficiently, the agent might get stuck on a bad or non-optimal solution. Once a first successful strategy is found and incorporated into the Q-network, that solution will always be picked because it has high  $q$ -values associated with it. This can prevent an agent from fully exploring the environment space and from finding (the most) efficient solutions to problems in the environment.

In order to facilitate a more thorough exploration of the environment and of the strategies available to the agent, an action selection policy can be used. A simple example of this is the  $\epsilon$ -greedy strategy [38]. Here the action with the highest  $q$ -value is chosen with probability  $1 - \epsilon$ , and a random action is picked with

probability  $\epsilon$ . By sometimes picking a random action, the agent can be maneuvered into unknown states which may prove to be beneficial.

Picking a suitable  $\epsilon$  for the entire training process is difficult. At the beginning of training more exploration is generally a good idea, suggesting a high  $\epsilon$ . Towards the end of training, one might want the learning agent to perfect and reinforce its strategies, which is easier with a lower  $\epsilon$ . A solution to this is annealing. When using annealing,  $\epsilon$  starts out with a relatively high value which slowly decays during the training process. This means that the agent can quickly explore the possible states at the beginning of training, while still benefiting from everything it has learned towards the end of training.





## Chapter 3

# Problem statement and overview

The research question this thesis aims to answer is “Can predatory pressure induce flocking behavior in Deep Q-Learning agents”? This was tested in both 2D and 3D environments. The predator’s behavior has been hand-coded, and the behavior of the agents was purely determined by Deep Q-Learning. The only objective of the agents was to not be eaten by the predator.

Through this research question, this thesis attempts to verify the claim made by Hahn et al. [13] that predatory pressure can induce emergent flocking in Deep Q-Learning equipped prey animals, in a simple 2D environment. It also attempts to extend this claim to a 3D environment. The underlying reasoning behind this is that predatory pressure is a prime candidate for the reason many diverse animals in many diverse environments display flocking or flocking-adjacent behavior. If flocking behavior can indeed be induced purely through predatory pressure, this strengthens the claim that predatory pressure can explain why flocking is so widespread in the animal kingdom. Before being able to come to a conclusion on this, several things had to be done.

First, models had to be built and trained for both the 2D and 3D cases. The 2D version was closely modelled after the model used by Hahn et al. The 3D model is a relatively straightforward extension of the 2D model. The 2D model was trained under a variety of parameter configurations, in order to maximise the chance of finding a configuration in which flocking occurs. Another advantage to this approach is that it might give insight into which parameters are important for the formation of flocking behavior. The exact specifications of the models and reasoning for made choices can be found in Chapter 4, Methods.

Secondly, the trained models needed to be evaluated on the extent to which the agents in them displayed flocking behavior. This meant that the concept of flocking needed to be quantified, ideally in a fashion that allows for comparisons between different models. This was done by making snapshots of agent positions while a model ran, and processing these snapshots in order to come to several measures indicative of flocking. Added to this was a visual sanity check, making sure that measurements indicating flocking were not instead produced by other behaviour that happened to have similar characteristics in the measured factors. The manner in which all of this was done is motivated and described in Section 4.5, Statistical Analysis.

Finally, the statistics were compiled in order to draw conclusions. Here, attention was paid to which parameters had what effect on the measurements of factors indicative of flocking. The compiled statistics are presented in Chapter 6, Results. A discussion of the methods used and experiments performed is given in Chapter 7, Discussion. Finally, concrete conclusions are drawn and laid out in Chapter 8, Conclusion.



# Chapter 4

## Methods

In this chapter the used methods will be explained and justified. First a high level view of the used models is presented in Section 4.1. Next, Section 4.2 will go in depth about the behaviour of the boids and Section 4.3 discusses behaviour of the predator. The manner in which the boids were trained is presented in Section 4.4. Finally, which statistical analyses were performed and in what manner is laid out in Section 4.5.

### 4.1 The general flocking model

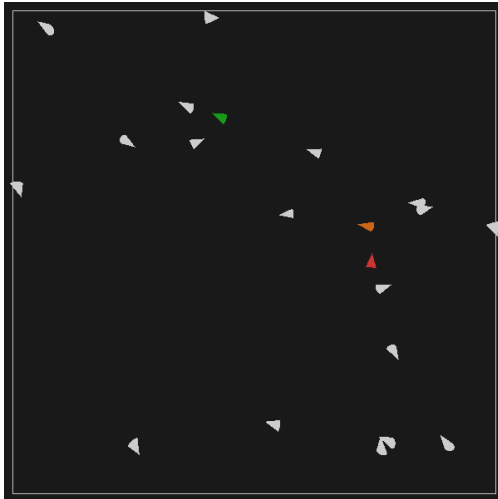
The computational 2D and 3D models developed for this research project largely share a common design. Both feature toroidal empty environments, populated by a number of boids and a predator. The models feature discrete time steps. Every step in the environment has the boids and the predator move forward and possibly rotate. The time steps are relatively small, allowing for smooth movement. Screenshots of both the 2D model and the 3D model are shown in Figure 4.1.

The boids select the action taken each step using a policy trained through Deep Q-Learning. The used reward function encourages the boids to stay alive by giving a small positive reward for staying alive and a large negative reward for dying. This means that any flocking behavior that occurs is a direct result of the boids' strategy to stay alive, because no reward is directly given for flocking-like behaviour. Each step, each boid feeds its current observation through its policy network. The action with the highest activation is then picked. Section 4.2 goes further into detail about how the boids work.

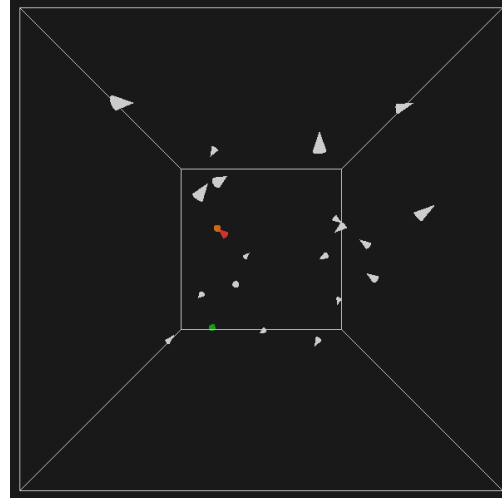
In contrast to the trained boids, the predator has hand-coded behaviour. It will select the closest boid as a target and chase it for a set amount of time. If the predator gets sufficiently close to the boid it is chasing, it will eat it. If there are multiple boids close to the predator, it will select one at random, simulating the predator getting confused by a flock of prey and having a hard time tracking any single one of them for an extended period of time. The resulting frequent target switching will effectively slow the predator down as it keeps changing direction. Additionally, the predator can speed up for a short amount of time to simulate a short sprint in order to catch the prey. In Section 4.3, the precise behaviour of the predator is laid out.

The training of the boids is done by only actually training a single boid and copying that boid's policy to all other boids. This both speeds up training and ensures a common strategy shared between all boids, which is essential for flocking to occur. A large flock cannot form unless a large number of boids are trying to flock together. The training occurs in episodes that end after a set number of steps or when the one trained boid is eaten. The exact method of training is specified in Section 4.4.2.

The models were created using python. The Deep Q-Learning implementation in Keras-RL was used [32] for the learning boids. This implementation is meant to be used with an OpenAI Gym [4] like environment, where the environment's step function is repeatedly called by the learning agent in order to advance the model. The written python code follows this convention. Further specifics of the training logic is found in Section 4.4.3.



(a) Screenshot of the 2D model. The white square is the edge of the environment



(b) Screenshot of the 3D model. The white lines are the edges of the cube-shaped environment.

Figure 4.1: Screenshots of the two models. The predator is red, the targeted prey boid is orange and the one trained boid is green. If viewed in black and white, the predator is the darkest grey, the current target is the lightest grey and the one trained boid is in between.

## 4.2 Boids

This section explains how the boids are implemented. First an overview is given to help in understanding the rest of the explanation. Then, the decision making of the boids and their possible actions are elaborated upon, followed by what happens if a boid is eaten.

### 4.2.1 General characteristics

The boids in the models move around freely in the environment. They move with a set speed and the only action they can take is changing their direction of travel. Generally, the boids are at most as fast as the predator in their respective model. Combined with slightly higher maneuverability than the predator, this creates a standard predator-prey dynamic. Since the predator is at least as fast as the boids and is possibly able to speed up for a short amount of time, simply running away from the predator is not a viable survival strategy. This leaves two possible successful strategies for a boid; it can either try and outmanoeuvre the predator, or fly close to other boids in order to form a flock that will confuse the predator.

### 4.2.2 Decision making

Each step of the model, each boid gets a state representing the current environment from that boid's point of view. This state contains information about the boid itself, the predator, and the  $n$  closest other boids. This simulates the boids not being able to keep track of all other boids, but only paying attention to the closest few. The state is passed to the boid as a 2D array with  $n + 2$  rows, featuring information about the boid itself at the top, then the predator, and finally the  $n$  closest boids.

The information in each row represents a snapshot of the current situation; each row contains the current distance and vector to the subject of the row as well as the current heading of the subject of the row. In the row about the boid itself, the current absolute position and current heading are given, with distance being reported as 0.

The state is then fed into each boid's neural network. This network has a neuron for each possible action in the output layer. The action with the highest activation is chosen and executed. Since this method can only select discrete actions, the boid cannot simply select the exact direction it wants to head in. The way

the boid picks a direction differs slightly between the 3D and 2D models. In 2D, the boid gets to choose between nine different options. If  $r$  is the maximum number of radians the boid can turn in one step, the nine options are equally space apart from  $r$  to  $-r$  (see table 4.1). The selected option is then used to rotate the boid the desired amount.

Action index	Rotation
0	$r$
1	$r \cdot 0.75$
2	$r \cdot 0.5$
3	$r \cdot 0.25$
4	0
5	$-r \cdot 0.25$
6	$-r \cdot 0.5$
7	$-r \cdot 0.75$
8	$-r$

Table 4.1: Possible actions for boids in the 2-Dimensional models where  $r$  is the maximum turning radius.

In 3D this is trickier, because multiplying a single scalar with a single rotation does not work anymore. A possible workaround is the utilisation of not only a left/right rotation, but also an up/down rotation, with separate scalars for each. While this does allow for free movement, it would require the boid to have not only a direction of travel but also an orientation. This in turn necessitates the boid having control over this orientation by being able to roll. In order to avoid this added complexity, the 3D models make use of a set number of goal directions. The boid is then able to pick between these goal directions, and it will turn towards the selected direction. The turning speed is limited by the boid's maximum turning speed. In total the boids can pick between fourteen directions (see Table 4.2). These directions correspond to the vectors from the center of a cube to the middle of its six faces and to all eight vertices. This creates a fairly uniform distribution of goal directions, ensuring that the boid can navigate as freely as is needed.

Action index	Direction vector
0	[1, 0, 0]
1	[0, 1, 0]
2	[0, 0, 1]
3	[-1, 0, 0]
4	[0, -1, 0]
5	[0, 0, -1]
6	[1, -1, -1]
7	[1, -1, 1]
8	[-1, -1, 1]
9	[-1, -1, -1]
10	[1, 1, -1]
11	[1, 1, 1]
12	[-1, 1, 1]
13	[-1, 1, -1]

Table 4.2: Possible goal directions for boids in the 3-Dimensional models.

### 4.2.3 Being eaten

If a boid is being chased by a predator and the predator gets sufficiently close, the predator will eat the boid. This results in the boid being removed from the environment. If the single trained boid is eaten, a negative reward will be given to the learning algorithm. This is described in more detail in Section 4.4.2.

The disadvantage of boids being eaten by the predator is that fewer and fewer boids remain. With fewer boids, confusing the predator by flocking becomes harder and harder. This shrinking of the boid population continues until the one boid that is actually being trained is eaten, at which point the world will reset and the population is briefly at maximum size, after which the boids are eaten once again and the population shrinks once again. The result of this is that the optimal conditions for flocking are only present a fraction of the time. In order to circumvent this problem, each boid that is eaten will respawn at a random position instantly. This is not the case if the one trained boid is eaten, as the model will simply restart then.

## 4.3 Predator

This section specifies the behavior of the predator in the models. First a quick overview of the behavior is given, accompanied by the algorithm of the predator in pseudocode. Afterwards, a more detailed explanation of the algorithm is presented.

### 4.3.1 General characteristics

The predator's behavior is determined by a hand-coded algorithm. This algorithm makes the predator attempt to catch and eat the boids. In short, it will select a target boid and then start moving towards it. Just like the predator in the flocking model from Hahn et al. [13], it briefly speeds up at a set interval. This is to simulate a leap or short sprint by the predator. In order to eat a boid, the predator needs to get sufficiently close to it. One of the experiments adds another behavior to the predator. In that case, the predator halts for a set amount of time after eating a boid. In nature, this is similar to the time spent eating the caught prey. This is implemented to make flocking a more viable strategy, because the boids have the time to reform any flocks that were broken up by the attack.

For the purpose of illustration, the version of the algorithm with pausing after eating is listed in Figure 4.2.

### 4.3.2 Target selection

Whenever the predator is without a target, a new one must be selected. This is done by first creating a list of all boids, ordered by the distance of the predator to each boid. If multiple boids are within the preset confusion distance, a random boid of those within that distance is selected. If no boids are within the confusion distance, the closest boid is selected as the next target. This is meant to simulate the predator being unable to focus on one boid out of many.

### 4.3.3 Chasing the target

Once a target is selected, the predator starts chasing it. This is done by turning in the direction of the target boid. The rate of turning is limited by the predator's maximum turning speed. The length of time the predator keeps chasing the same target before selecting a new one is limited by a chase timer, set as soon as a target is selected. The reasoning behind this is that a more convenient prey boid may have presented itself while the predator was chasing the current target. If the predator were to instantly switch to this new, now closest target, it would have a hard time ever catching a single boid because it would switch between targets and thus directions so frequently that it would not be able to get close enough to any boid. By locking on to a single target for a while, the predator can actually approach and eat it. The intensity of this confusion-like effect of repeatedly switching targets can be tuned by changing the length of the chase timer; lower values simulate a higher amount of confusion.

If the chase timer runs out, the current target is forgotten and a new one is selected. This is where the confusion effect becomes most clear: if the predator is close to multiple boids, it will likely switch to another target once the chase timer runs out. The result of this is that the predator needs to change direction in order to go after the current target. Ideally, for the boids, once the predator is closing in on its new target, it will switch again to yet another target.

During the chase, the predator speeds up occasionally. The duration of this sprint and the interval at which it occurs are set parameters (*sprint duration* and *sprint interval* respectively). The occasional speedup

```

set target to None
set eating_timer to 0
set chase_timer to 0
set current_tick to 0
LOOP
  IF chase_timer is 0 THEN
    set target to None

  IF target is None THEN
    create list of boids ordered by closeness to predator
    IF multiple boids are closer than confusion_distance THEN
      set target to random boid within confusion_distance
    ELSE
      set target to closest boid
    SET chase_timer to chase_time

  change predator direction towards target by at most the turning_speed

  IF eating_timer is greater than 0 THEN
    do not move the predator
    decrement eating_timer by 1
  ELSE IF current_tick modulo sprint_interval is less than sprint_length:
    move predator forward by sprint_speed
  ELSE
    move predator forward by speed

  IF distance to target is less than eating_distance THEN
    eat the target
    set eating_timer to eating_time
    set target to None

  decrement chase_timer by 1
  increment current_tick by 1

```

Figure 4.2: The algorithm of the predator including pausing after eating. Variables are styled in *italic* and customizable parameters are styled in **bold**. The values used for these parameters are given in Section 5.2.1 for the 2D models and in Section 5.4.1 for the 3D models.

is implemented by taking the current tick number and performing a modulo on it with the *sprint interval*; if the result of this modulo is less than the *sprint duration*, the predator moves at sprinting speed. If the modulo is greater than the *sprint duration*, the predator is moved at the normal speed. This way a sprint starts every *sprint interval* ticks and lasts for *sprint duration* ticks each time.

#### 4.3.4 Eating the target

Each step, the distance to the target is compared to the distance at which the predator can eat the target. If the target is close enough, the predator will eat it and the current target will be cleared. What happens to a boid if it is eaten is specified further in Section 4.2.3. If pausing after eating is enabled, the eating timer is also set at this point. While this timer is counting down, the predator will not move, simulating the time spent eating the prey. This gives the remaining boids a chance to reform any flock or formation they may have been moving in. It also prevents the predator from quickly eating its way through a denser group of boids, which would likely discourage flocking.

## 4.4 Learning

This section is dedicated to the manner of learning used by the boids in both the 2D and the 3D model. It will discuss which learning method was used and why it was chosen. After this, the way the boids were trained will be explained.

### 4.4.1 Why Deep Q-learning?

There are many different ways of making an agent in an environment learn something. For the purposes of this Master’s thesis, some type of Deep Learning method is preferable, as it allows the agent to generalize past experiences to similar, but new situations. This is important, because the models feature a high-dimensional continuous state-space, which can not be fully explored during the learning process. This, combined with Deep Q-Learning’s past success in the learning of flocking [13] by Hahn et al., made Deep Q-Learning a promising option for this thesis.

An alternative that was also looked at was Deep Deterministic Policy Gradient (DDPG). In addition to Deep Q-Learning, Hahn et al. [13] also had success using this in order to teach boids flocking. The major upside of DDPG over Deep Q-Learning is DDPG’s ability to utilize a continuous action space. This allows the boids to be able to move around unrestricted by the discrete turning-steps necessary for Deep Q-Learning. A downside to DDPG is that it can be unstable even in relatively simple environments [25, 21, 44].

In an early prototype model where a boid was tasked with learning to stay within a certain area, a DDPG-equipped boid failed to learn this consistently. A boid with Deep Q-Learning performed better at this task, quickly and consistently learning to stay within the set area. This led to the choice for Deep Q-Learning. While a continuous action space would be preferable to a discrete one, flocking has been achieved before using a discrete action space [13]. Additionally, since the amount of movement of boids between model steps is small, alternating between discrete options can still lead to directions of movement that are not included in the discrete action space.

### 4.4.2 Training the boids

The method used for training the boids is important. There should be room and opportunity for the boids to display the goal behavior of flocking without the boids being encouraged directly to group and flock. In order to achieve this, the reward function does not look at the behavior of the boids; it instead purely looks at the period they manage to stay alive. Boids get a small positive reward for each step they are alive. If a boid does get eaten by the predator, it gets a large negative reward. When this function is maximised, it will lead to evasion of the predator. If flocking is indeed, as suspected, an efficient solution for evading capture by the predator, it should be able to arise from these circumstances.

Training multiple boids at once can be done in various ways. Training each boid individually is an intuitive solution. There are, however, some downsides to this. The first is the amount of time needed to train each boid. Each boid needs to have its network updated each step, which involves a lot of computation time. A second difficulty is that for flocking to occur, all or a majority of boids need to have it as a strategy. When all boids are trained independently, this is hard to achieve. Even if one boid stumbles upon a flocking-like strategy, without the cooperation of the others it is unlikely to succeed.

The used method instead only trains one single boid. Each step this trained boid’s network is also used to determine the actions of all other boids. After each action taken by the one trained boid, the network is updated using the reward value that the taken action resulted in. While the other boids use this same network, it is not updated after these other boids have taken their actions. The result of this method is exactly what we are looking for; the boids all learn to avoid the predator and they use the same strategy to do so. This means that the scenario where a boid tries to flock but lacks the support from other boids to do so cannot occur.

### 4.4.3 The flow of training

The total training time is defined as a number of steps. These steps form episodes; an episode ends when either a set maximum episode length has been reached or the one trained boid is eaten. After an episode ends, the environment is reset and a new episode begins.



Each episode starts with the construction of an initial observation for the one trained boid. This observation is used to kick-start the action-reward-observation loop that is responsible for the actual training. The initial observation is fed through the trained boid's network in order to obtain an action that the trained boid should take. This action is executed, after which a reward value for the new state is determined and communicated to the trained boid, together with a new observation. The reward is used to update the network, and the observation is used to obtain a new action. This new action is then executed, which produces a new reward and a new observation, and so forth.

In this description of the training loop, the actions taken by the other boids and the predator have not been incorporated. With these added, the order in which the model updates each step is as follows. The action of the trained boid is executed first. As this action is purely a rotation, the trained boid has not moved yet. This action is determined using the observation from the previous tick. Next, the non-learning boids determine their actions through the use of the trained boid's network. Each non-learning boid feeds its own observation, constructed in an identical manner to the trained boid's observation, through the trained boid's network. The determined best actions are then executed. This does not move the other boids yet, only rotates them. Then the predator acts. It changes direction to its target if needed, and then moves. This is also the point at which the predator checks whether it is close enough to a boid to eat it (and does so, if it is close enough). Finally, all boids, including the trained boid, move in the direction they are pointing. Now that everyone has moved, the trained boid's new observation is made and the reward determined. In addition, it is also checked whether the episode needs to end prematurely, which happens if the trained boid is eaten.

The process then repeats, with the new action of the trained boid being executed, followed by all non-learning boids determining their action and executing it, followed by the predator's action and movement, followed by all boids including the trained boid moving. A new observation is constructed, and so forth.

The complete step function of the model is listed in Figure 4.3.

```

FUNCTION Step(action)
{
  execute action for the trained boid
  FOR boid in all non-learning boids
  {
    construct observation for boid
    determine action for observation
    execute action for boid
  }

  execute predator actions
  move all boids forward

  construct new observation for the trained boid
  determine reward for the state of the trained boid
  determine if episode is done

  return observation, reward, done
}

```

Figure 4.3: The Step method used for training. The framework from which it is called is listed in Figure 4.4.

This step function is called by the trained agent in order to execute the selected action, progress time in the model and obtain the new observation. The algorithm of the one trained agent is listed in Figure 4.4.

Once the model has been trained, it is used for an experiment. In this experiment, it is run for a large number of steps, with a snapshot of the world being taken regularly. This is described in more detail in Chapter 5, Experiments. The snapshots are used for statistical analysis which will be explained further in the next section.

```

set total_steps to 0

WHILE total_steps is less than n_steps
{
  # start of an episode
  initialize model in order to obtain initial observation
  set episode_steps to 0

  WHILE episode_steps is less than max_episode_steps
  {
    pass observation through network to obtain action
    set observation, reward, done to Step(action)
    update network using reward

    increment episode_steps by 1
    increment total_steps by 1

    IF done
    {
      break
    }
  }
}

```

Figure 4.4: The training algorithm. The called Step function is listed in Figure 4.3.

## 4.5 Statistical analysis

The data generated by an experiment is a list of snapshots of the world. These snapshots contain the position and rotation of each agent in the world as well as the number of boids eaten since the last snapshot was taken. This section will go into how and why these snapshots are processed. It will also discuss the statistical tests used.

### 4.5.1 Objective of the analysis

The analysis of the experiment results should accomplish two main tasks. Firstly, it should operationalise the concept of flocking. This is necessary because it is not possible to directly measure a degree of flocking. While it is somewhat possible to judge flocking visually, this is not reliable enough to base solid conclusions on.

Secondly, the analysis should allow for statistical comparisons between differing models. This helps in determining to what degree differing experimental results between two models are due to differences between the models or due to chance. By then comparing a range of models against each other, parameters which encourage or discourage flocking could be identified.

### 4.5.2 Measurements

In order to flock, the members of the flock must head in about the same direction as the flock as a whole and they must stay together as a relatively coherent unit. These two factors, deviation from average heading and average distance between flock members, have been used as a basis for the analysis that has been performed.

Computing these two measurements over the entire population of boids in each snapshot does not make much sense; they should be done per flock. In order to approximate this, the DBSCAN clustering algorithm is used [10]. This algorithm works using two parameters, an  $\epsilon$  deciding how large each point's neighbourhood is, and a `min_neighbours`, determining how many points must be within a point's neighbourhood (including

itself) for that point to become a *core point*. Core points in one another's neighbourhood belong to the same cluster. Any points that do not meet the `min_neighbours` requirement but are within a core point's neighbourhood are still considered belonging to the core point's cluster. Any points that are not a core point themselves and are not in the neighbourhood of a core point are considered outliers and are not included in any cluster.

For each of these clusters, the average pairwise distance between the boids is computed, as well as the average deviation from the average heading of the cluster. The size of the cluster is recorded as well. The result of this is that the collection of snapshots from an experiment is reduced down to several important elements. These are:

- **The total number of clusters:** A larger number of clusters, all else being equal, indicates a stronger tendency to flock by the boids.
- **A table with for each cluster:**
  - **Number of boids in the cluster:** All else being equal, a larger average number of boids within each cluster indicates a higher amount of flocking.
  - **Average pairwise distance:** A higher average pairwise distance means that a flock is looser. A lower average pairwise distance means a flock is denser.
  - **Average deviation from the average heading:** A higher value here means that a flock is messy, with individual boids not necessarily heading in the same direction. A low value indicates a common heading among boids making up the flock.
- **The total number of boids eaten:** A measure of model performance. The lower this is, the more successful the boids were in evading the predator.

For an example of the average pairwise distance and the average deviation from the average heading measurements in action, consider the configuration of boids in Figure 4.5. In this configuration, two clusters have been identified: an orange cluster and a blue cluster (light and dark grey respectively if viewed in black-and-white). The orange cluster is more spread out and is quite polarised. The blue cluster on the other hand is very tightly packed, and features boids heading in many different directions. This is mirrored in the measurements. The orange cluster has an average pairwise distance of 0.180 units and an average deviation from the average heading of 0.050 radians. The blue cluster has an average pairwise distance of 0.073 units and an average deviation from the average heading of 0.941 radians. This correlates with the observations that the orange cluster is spread out and polarised, and the blue cluster is tight and not polarized.

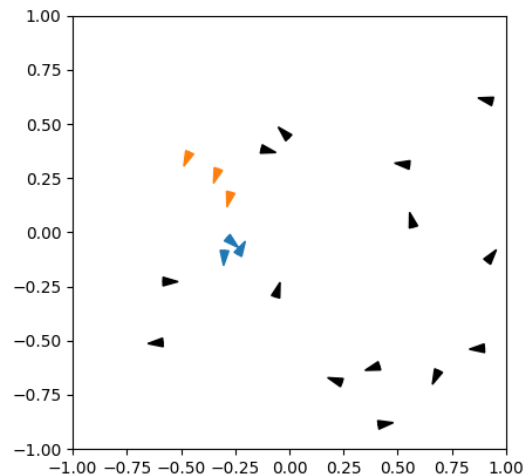


Figure 4.5: Configuration of boids with two identified clusters.

### 4.5.3 The performed analyses

This section will go over the analyses that were performed.

#### Experiment reliability

Before drawing conclusions based on the done experiments, the reliability of the experiment must be demonstrated. If the experiment is too short and does not capture enough data about the model being measured, it is of no use. In other words, experiments of the same model should not differ significantly in their results.

This is tested by testing a model for each possible parameter permutation twice. The resulting pairs of experiments are then tested for any significant differences using a Hotelling's T-squared test [36] over the size of each cluster, the average pairwise distance within each cluster and the average deviation from the average heading within each cluster. In order to reduce this set of 32 tests to a single p-value, Fisher's method is applied [11]. The results of all this are presented in Section 6.1.4.

#### Regression model

In order to find out more about what factors cause flocking-like behavior, linear regression models were fitted to the measured flocking indicators. These models, given the parameter configuration of a model, predict the expected measurements for the clusters in that model. The best fitting model for each of the measurements was determined using the Akaike Information Criterion, which is an estimator for model quality. The best fitting models and their coefficients are presented in Section 6.1.3.

## 4.6 Code

The source code of all models, as well as the tools used for analysing the experiments, is available as a GitHub repository, and can be found on <https://github.com/JRdeLange/MasterThesis>. The trained models can also be found there.

# Chapter 5

## Experiments

In this chapter, the experiments that were performed are explained. First, the general experimental design is laid out in Section 5.1. Next, the three experiments that have been performed are presented in their own sections. These are the main 2D experiment in Section 5.2, a second smaller 2D experiment in Section 5.3, and finally a 3D experiment in Section 5.4.

The main 2D experiment trains the boids under varying parameter circumstances, among which those used by Hahn et al. [13]. The second 2D experiment adds a pause after eating for the predator. The 3D experiment is exploratory, to see if any flocking observed in the 2D experiments also emerges in a 3D environment.

### 5.1 General experiment design

All experiments that were performed followed the same structure. The goal of the experiments was to establish whether there was any flocking-like behaviour exhibited by the boids after they have been trained. If this was the case, then the predator was likely the cause of this behavior, as no other hazards were present.

In each experiment, the model was trained for a number of steps, after which the model was tested for a set number of steps. During this test, the model was no longer updated and always selected what it deemed to be the most optimal action. While the test was running, data on the positions and rotations of the boids was collected (see Section 4.5.2).

The 2D model was trained for 500.000 steps before being subjected to the experiment. This follows Hahn and colleagues, who also trained their 2D Deep Q-Learning flocking model for 500.000 steps [13].

The 3D model was trained for 1.000.000 steps before being tested in the experiment. The model was trained longer than the 2D model since a 3D model is inherently more complex.

From the data collected from each experiment, measurements indicative of flocking behavior will be calculated. These measurements are the number of eaten boids, the total number of boid clusters, the number of boids in each cluster, the average pairwise distance within each cluster and finally the average deviation from the average heading within each cluster. More detail on this is presented in Section 4.5.2.

The next sections will go further in depth on the experiment specific details.

### 5.2 2D experiment 1

After having been trained for 500.000 steps, the model is run for an additional 100.000 steps, without any training, in order to collect data about the behavior of the boids. In order to prevent any close (causal or temporal) relation between data points, a collection interval of 100 steps is used. This ensures that each snapshot taken of the model does not relate to any other snapshot.

Since the training of a randomly initialized Deep Q-Network is not deterministic and not guaranteed to achieve maximum performance, each configuration of parameters has been trained and tested a total of three times. These three repetitions of each parameter configuration will henceforth be referred to as three “sets” of trained models. For the statistical analysis, only the best performing model out of the three sets will be

taken into account. Which result is best is determined through the number of boids that have been eaten over the course of the test, because not being eaten is the only objective with which the boids have been provided. A summary of the results of the three sets of models can be found in Appendix A.

In order to determine the reliability of the experiment itself, the experiment was run twice on one of the sets. By comparing the similarity of the results between the two experiments that used the same underlying model, experiment reliability can be judged. Very similar measurements indicate that the experiment generates enough data points in order to accurately judge model performance. The results of this comparison can be found in Section 6.1.4.

### 5.2.1 Parameters

Whether or not flocking behavior emerges is dependent on parameters in the model. For example, if the boids can easily outrun the predator, there is no reason for them to form a flock in order to confuse it. In order to maximize the chance that the model produces flocking behaviour, several different configurations of model parameters have been trained and tested. The parameters that were varied were chosen because they influence key features in the model. The varied parameters will now be named and explained.

- **Boid speed:** By varying the movement speed of the boids, the relative speed to the predator changes. It stands to reason that this will affect their behavior. Additionally, by the boids moving a smaller or larger distance each step, their relative maneuverability to the predator is varied as well. This is most easily imagined through the size of the smallest circle they can circumvent; Even if the turning speed stays the same, if a boid moves a smaller distance each step, the resulting circle will be smaller.

The values used were 0.0333 (approximately 1/60th of the world each step) and 0.0283 (approximately 1/70th of the world each step).

- **Boid count:** By varying the number of present boids, we are effectively varying the boid density. Flocking with a larger number of boids may be more effective, because the predator has more potential prey agents to randomly pick between. On the other hand, since there is a larger density of boids, maybe flocking is not as necessary for survival, since the predator encounters a new closer boid more frequently than in lower boid densities.

The values used were 10 total boids and 20 total boids.

- **Predator chase time:** This varies the number of steps the predator is locked on to a single target. When this is high, the predator will stay locked on to its target for longer. If this is low, the predator will switch targets frequently, mimicking the confusion effect. This setting therefore fairly directly controls how viable flocking is as a strategy.

The values used were 20 steps of continuous lock-on and 10 steps of continuous lock-on.

- **Number of observable neighbours:** Varying this has two main goals. First, it is a good control trial in order to see if any observed flocking can be interpreted as intentional grouping. After all, boids in similar positions might simply act similarly, therefore looking like they are flocking. If all flocking observed in tests where boids can see multiple neighbours also occurs in tests where boids cannot see any neighbours, then the observed flocking is likely a side-effect of boids just responding similarly to similar circumstances.

Secondly, a smaller number of observable neighbours means that there is less information to pay attention to, which could lead to different behavior in the trained boids.

The values used were 5, 2, 1 and 0 observable neighbours.

All possible permutations of the above parameters (see Table 5.1) were run. As already mentioned, all permutations were trained and tested three times, with the best performing version of each permutation being used for the final statistical analysis.

Parameters other than the ones above were not varied at all. These parameters and their values are:

Configuration name	Observable neighbours	Boid count	Boid speed	Predator chase time
5_obs_10_boids	5	10	0.0333	20
5_obs_10_boids_short_chase	5	10	0.0333	10
5_obs_10_boids_slower	5	10	0.0283	20
5_obs_10_boids_slower_short_chase	5	10	0.0283	10
5_obs_20_boids	5	20	0.0333	20
5_obs_20_boids_short_chase	5	20	0.0333	10
5_obs_20_boids_slower	5	20	0.0283	20
5_obs_20_boids_slower_short_chase	5	20	0.0283	10
2_obs_10_boids	2	10	0.0333	20
2_obs_10_boids_short_chase	2	10	0.0333	10
2_obs_10_boids_slower	2	10	0.0283	20
2_obs_10_boids_slower_short_chase	2	10	0.0283	10
2_obs_20_boids	2	20	0.0333	20
2_obs_20_boids_short_chase	2	20	0.0333	10
2_obs_20_boids_slower	2	20	0.0283	20
2_obs_20_boids_slower_short_chase	2	20	0.0283	10
1_obs_10_boids	1	10	0.0333	20
1_obs_10_boids_short_chase	1	10	0.0333	10
1_obs_10_boids_slower	1	10	0.0283	20
1_obs_10_boids_slower_short_chase	1	10	0.0283	10
1_obs_20_boids	1	20	0.0333	20
1_obs_20_boids_short_chase	1	20	0.0333	10
1_obs_20_boids_slower	1	20	0.0283	20
1_obs_20_boids_slower_short_chase	1	20	0.0283	10
0_obs_10_boids	0	10	0.0333	20
0_obs_10_boids_short_chase	0	10	0.0333	10
0_obs_10_boids_slower	0	10	0.0283	20
0_obs_10_boids_slower_short_chase	0	10	0.0283	10
0_obs_20_boids	0	20	0.0333	20
0_obs_20_boids_short_chase	0	20	0.0333	10
0_obs_20_boids_slower	0	20	0.0283	20
0_obs_20_boids_slower_short_chase	0	20	0.0283	10

Table 5.1: All ran 2D model permutations.

- **World dimensions:** The width and height of the world. While not directly influencing the behavior of the boids or the predator, this parameter is important to contextualize the other distances mentioned. The used values here are -1 to 1 on both axes.
- **Boid turning speed:** The maximum number of radians a boid can turn each step. The used value here is  $0.5\pi$ .
- **Predator speed:** The distance the predator moves each step. The used value here is 0.0333.
- **Predator sprint speed:** The distance the predator moves each step during the periodic speed-up lunge. The used value here is 0.0533.
- **Predator sprint interval:** The step interval at which the predator starts sprinting, increasing its speed. The used value here is 100 steps.
- **Predator sprint length:** The number of steps the predator stays at sprint speed. The value used here is 20 steps.

- **Predator turning speed:** The maximum number of radians the predator turn each step. The used value here is  $0.25\pi$ .
- **Predator confusion distance:** Boids closer than this distance to the predator will be considered close enough to cause confusion. If more than one boid is within this distance, the predator will pick a target at random from all boids within this distance. The used value here is 0.4.
- **Predator eating distance:** The maximum distance at which the predator is considered close enough to a boid in order to eat it. Since agents in the model do not have a diameter, this is measured between the actual positions of the boid and the predator. The used value here is 0.0666.

During training, an action selection policy is used to help the agents in exploring the state-space. The used policy is an annealing epsilon-greedy policy [38], which was described in detail in Section 2.3.4. At the beginning of training we set  $\epsilon = 0.2$ , which linearly scales to  $\epsilon = 0.002$  at the end of training.

## 5.2.2 Analysis

This section details the analyses performed on the results of the experiments. It first discusses the clustering method used for determining potential flocks. Next, it discusses the statistical analysis that will be performed.

### Clustering

For the analysis of each of the trained model permutations, the DBSCAN clustering algorithm [10] is used to determine clusters, within which flocking-indicating values are then determined. The specific implementation used [31] allows for the specification of several parameters to customize the manner of clustering. As described in Section 4.5.2, this requires the setting of an  $\epsilon$  to determine the size of the neighbourhood around a point and a `min_neighbours` to determine how many points need to be within a point's neighbourhood for that point to be a core point of a cluster.

After some experimentation, an  $\epsilon$  value of 0.2 with a minimum cluster size of 3 provided reasonable results. A lower  $\epsilon$  quickly made the formation of clusters almost impossible, as shown in Figures 5.1 below.

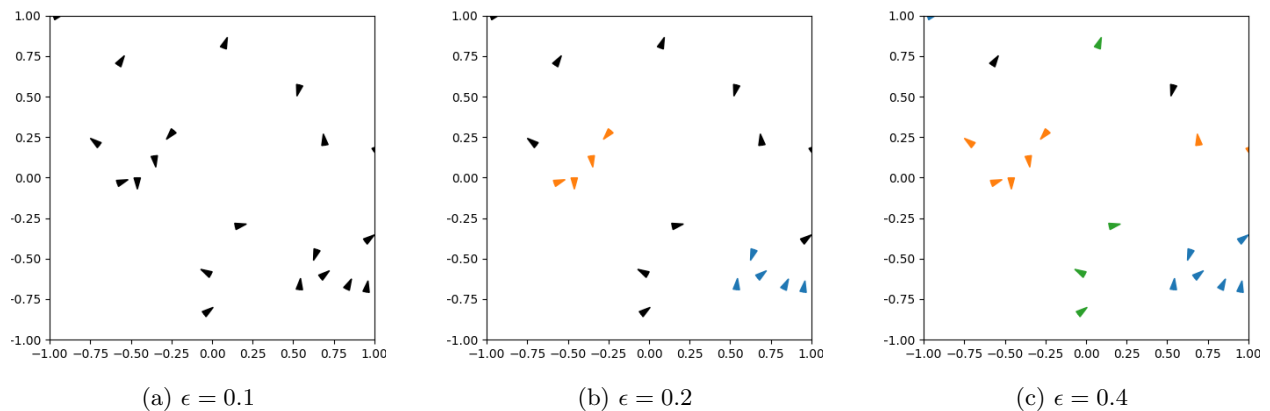


Figure 5.1: Clusters with different values of  $\epsilon$  and `min_neighbours`= 3. Each color constitutes a cluster.

In Subfigure 5.1(a), no clusters are formed at all, while in Subfigure 5.1(c) some agents that are quite far apart are still classified as a cluster. The orange cluster (light grey in black and white) in Subfigure 5.1(c) is a good example of this. Instead of only encompassing the four boids close together, like in Subfigure 5.1(b), it now loops around the world to the right side, incorporating boids that are quite distant from the original cluster.

Similar experimentation with the value of `min_neighbours` can be seen in Figure 5.2.

A `min_neighbours` of 2, like in Subfigure 5.2(a), leads to pairs of agents being classified as a cluster. This bar is too low. With a value of 4 however, like in Subfigure 5.2(c), clusters are only made very small. If we look at the blue clusters (dark grey in black and white) in Subfigures 5.2(b) and 5.2(c), we can see that in



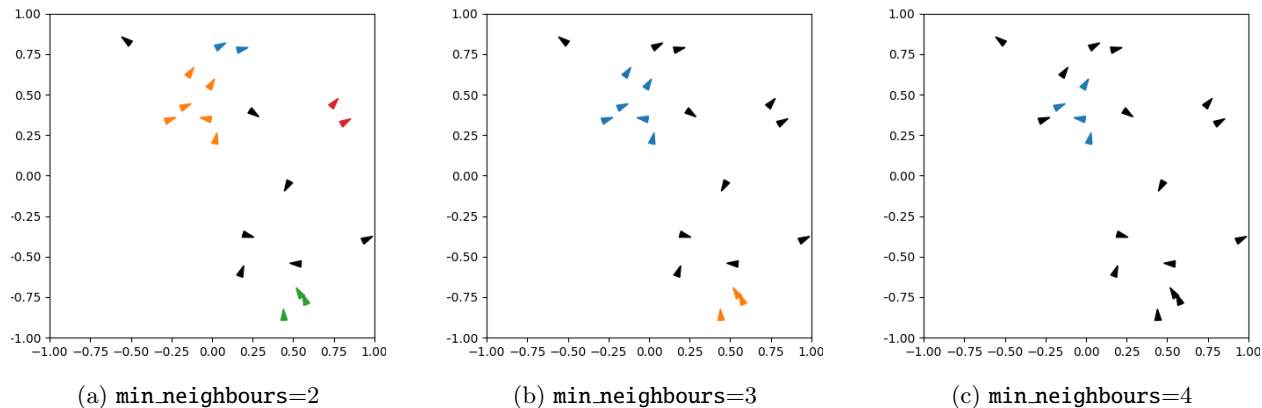


Figure 5.2: Clusters with different values of `min_neighbours` and  $\epsilon = 0.2$ . Each color constitutes a cluster.

(c) two boids are culled from the cluster even though they are not noticeably separate from the four boids that are still in the cluster.

The exact values here should not have a large influence on the measurements as long as they are reasonable. If flocking is actually present, it should show in the results even if clusters sometimes include an extra boid or do not include a boid that should have been included.

### Statistical analysis

In order to understand the impact that the different parameter configurations have on the measured flocking indication variables, linear regression will be used. Separate regression models will be fitted for each measured flocking indicating variable, giving insight into correlations between the parameters and boid behavior.

## 5.3 2D experiment 2

After 2D experiment 1 was performed and no clear flocking behavior emerged (see Section 6.1), one of the parameter configurations was trained and tested again with a change in predator behavior. In order to lessen the predatory pressure on the boids, a pause after eating was implemented into the predator algorithm (see Figure 4.2). The idea behind this was that this would prevent the predator from eating a whole cluster of boids quickly. By pausing the predator for a set amount of time after each time it eats a boid, all other boids get a chance to distance themselves from the predator and, possibly, to form a flock. Three separate models were trained and tested in this experiment in order to create three opportunities where flocking-like behavior could emerge.

### 5.3.1 Parameters

This experiment was run using configuration `2_obs_20_boids_short_chase` in Table 5.1, with the addition of predator pausing. A configuration with 20 boids was chosen because this increased amount of boids allows for more and larger flocks. A short chase condition was chosen as it further favors the boids, allowing for more flocking opportunities. Finally, an observed neighbour boids condition was picked as this is required for real flocking behavior. As the number of observed neighbours did not appear to make a large difference in the results (see Section 6.1.2), the middle option of two observed neighbours was used.

### 5.3.2 Analysis

The behavior of the boids in the finished models was analysed and examined for evidence of flocking-like behavior. Since it was clear from this analysis that no flocking-like behavior was present (see Section 6.2.1), no further analysis was done.

The results were not directly compared to the same configuration without predator pausing from experiment 1, as the small dataset (only three models) and the inconsistency of trained models meant that a direct comparison would not be meaningful.

## 5.4 3D experiment

The final experiment that was conducted is one with a 3D environment. Apart from the environment being 3D and the actions that the boids choose to move around (see Section 4.2.2), the model is identical to the 2D models.

This experiment was meant to see whether any flocking behavior observed in the earlier 2D experiments would transfer over to a 3D environment. Although eventually no flocking-like behavior was observed in any of the 2D experiments, the 3D experiment was still run.

### 5.4.1 Parameters

Just like the second 2D experiment, this experiment's parameters are taken from the `2_obs_20_boids_short_chase` configuration (in Table 5.1). A number of 20 total present boids were used since this allows for more flocking opportunities. The predator's chase was short, since this gives an advantage to the boids. The two observable neighbours condition was picked since the number of observable neighbours did not appear to strongly influence the behavior the boids displayed in the 2D experiments (see Section 6.2.1). Picking a value here where neighbour boids were observable, but the observation did not contain too much information, seemed like a good balance, so a number of two observable neighbours was chosen.

### 5.4.2 Analysis

The results of this experiment were established through looking at the behavior of the boids. While measurements were taken, these did not show any interesting trends nor did they hint at anything not observable through simply looking at the behavior of the boids.

# Chapter 6

## Results

This chapter presents the results of all three performed experiments; the first and main 2D experiment (see Section 6.1), the second 2D experiment with predator pausing (see Section 6.2), and the 3D experiment (see Section 6.3). In each case, the results are reported in several subsections. First, a look is taken at the behavior the boids displayed during the experiment. Next, the measurements are presented through tables and various graphs. Finally, a more thorough analysis of the measurements is performed.

### 6.1 2D experiment 1

This section is dedicated to the presentation of the results of the first 2D experiment. In summary, this experiment trained a group of boids through Deep Q-Learning. The goal of the boids was to not be eaten by a predator. In total, the boids were trained under 32 different parameter configurations (see Section 5.2.1). After training, the boids were tested in order to collect measurements of flocking-indicating factors in the behavior of the boids (see Section 4.5.2). These measurements were then processed (see Section 5.2.2) into the results presented in this section.

Three separate models were trained and tested for each parameter configuration. Then, in each configuration only the model with the lowest number of eaten boids during the experiment was used for the data and analysis presented in this section. This was done in order to reduce the inherent variance in performance of trained neural networks, like the q-networks used by the boids. A summary of all trained and tested models, including the ones that are not presented or used for analysis in this section, can be found in Appendix A.

If there is indeed emergent flocking-like behavior, this should become clear in the data presented in this section. This would become evident in several ways; behavior which visually resembles flocking should be observable. There should also be differences between the zero observed neighbour conditions and those where boids did observe their neighbours. The expected differences are more and larger clusters in observable neighbour conditions, as well as more aligned clusters in those conditions. This is because any boids displaying real flocking behavior must utilise the observations of nearby boids in order to stay close to them and to align with them. This is impossible in the conditions where no neighbouring boids are observed. Flocking-like behavior therefore can only be present in conditions with observable neighbours.

This section is structured as follows: in Section 6.1.1 the types of behavior the trained boids display are shown and compared to flocking. Next, Section 6.1.2 gives an impression of the taken measurements through tables and graphs, in addition to checking if the measurements indicate flocking-like behavior. Section 6.1.3 then presents statistical analyses of the taken measurements. Finally, Section 6.1.4 discusses the validity of the experimental design.

#### 6.1.1 2D experiment 1 boid behavior

In this section, the types of behavior observed in the experiments will be shown. For each of the described types of behavior it will be judged how much they resemble flocking behavior. Flocking behavior here is defined as highly polarized groups of boids that move in unison by reacting to other boids in the same flock. The behavior the boids displayed over all of the experiments roughly falls within one of four identified

categories; random spinning, random spinning except when the predator is near, uniform fleeing, and linear fleeing. These are now discussed one by one.

Each type of behavior is represented by a sequence of three images. The leftmost image is the first of the sequence, with the middle image being taken a few steps after the leftmost image, and the rightmost image being taken a few steps after the middle image. This gives a rudimentary sense of the movement of the boids.

### Random spinning

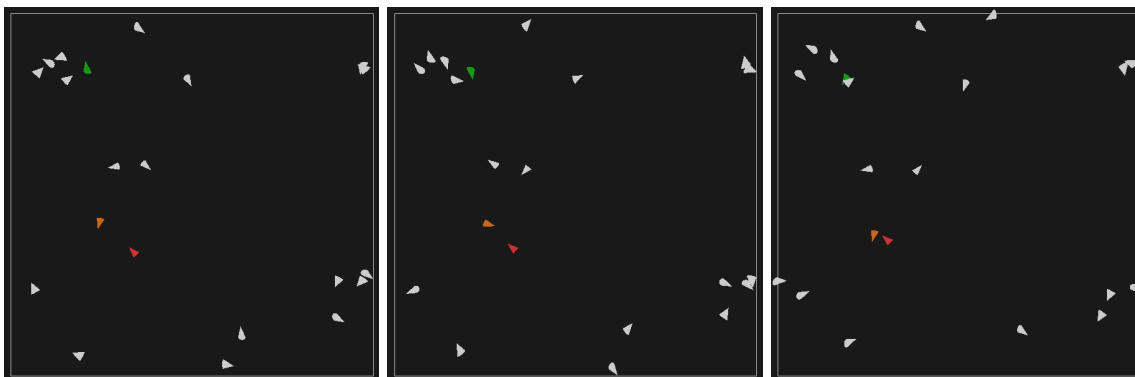


Figure 6.1: Boids showing exclusively spinning behavior. The images are ordered temporally from left to right. The red agent is the predator, the orange boid is the current target and the green agent is the one trained boid. If viewed in black and white, the predator is the darkest grey, the current target is the lightest grey and the one trained boid is in between.

Figure 6.1 shows some of the least effective behavior the boids displayed. No reaction to the predator was observed, and boids simply spun in circles or moved around randomly. This specific example was taken from configuration 2\_obs\_20\_boids in set one (see Table A.4). This behavior does not qualify as flocking-like. The boids do not group together at all, let alone move together as a polarized group.

### Random spinning except when the predator is near

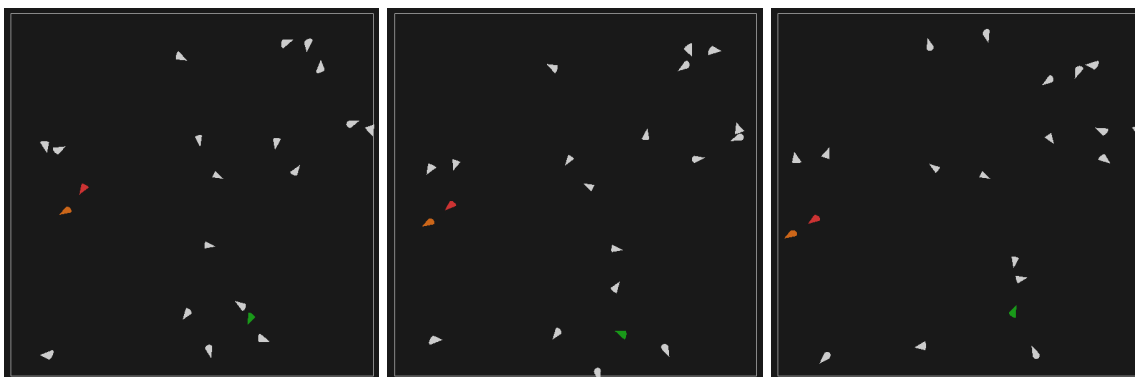


Figure 6.2: Boids showing spinning behavior except when the predator is near.

The behavior depicted in Figure 6.2 consists of boids moving randomly or spinning around, like in Figure 6.1, except that when the predator gets close, they do react to it by fleeing. As can be expected, this is more effective than not reacting to the predator. This is also evident in the number of eaten boids of the two configurations these images were captured from; the purely spinning boids were eaten a total of 14534 times,

whereas the boids in the configuration from which this example was captured (configuration 2\_obs\_20\_boids in set two. See Table A.5) had 8486 boids eaten.

Once again, this behavior does not qualify as flocking-like. The boids do not cluster together, which is necessary for flocking-like behavior.

### Uniform fleeing

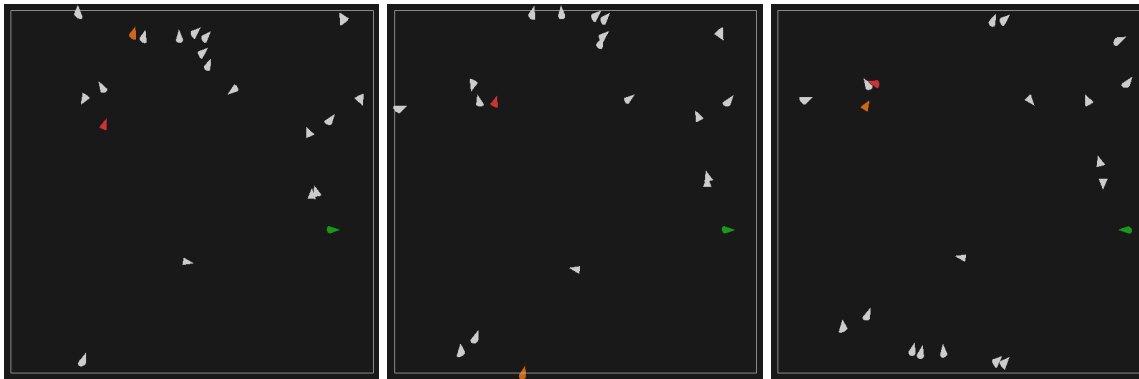


Figure 6.3: Boids showing uniform fleeing behavior.

The most flocking-like behavior observed is depicted in Figure 6.3. Here, boids fairly uniformly flee from the predator, forming what appears to be a flock. This potential flock starts at the top of the environment and travels diagonally upward, reappearing at the bottom of the environment. This behavior appears quite flocking-like; the boids here are grouped together and they are moving uniformly.

This specific example, though, is captured from a condition where no neighbouring boids were observed ((configuration 0\_obs\_20\_boids in set two. See Table A.11). Because this behavior can also occur without reacting to any neighbouring boids, this also is not flocking behavior. These are simply lone boids, fleeing a predator, close to one another.

### Linear fleeing

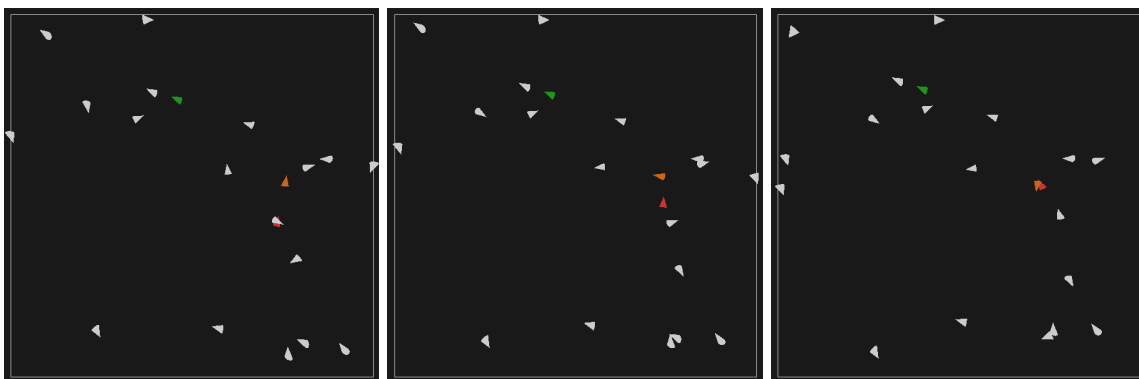


Figure 6.4: Boids showing linear fleeing behavior.

The final type of observed behavior is shown in Figure 6.4. This behavior consists of boids going mostly straight, although less uniformly in one direction than the uniform fleeing behavior seen in Figure 6.3.

This type of behavior does not feature boids grouping together, and therefore cannot be classified as flocking-like behavior.

### 6.1.2 2D experiment 1 results

This section will further explore to what extent the behavior displayed by the trained boids is flocking-like. If there is flocking-like behavior, the following observations are expected:

- **Number of clusters:** If there is flocking-like behavior, that means that boids are actively going towards one another. This should result in more clusters in conditions where neighbouring boids are observed than in conditions where no neighbours are observed.
- **Cluster size:** If there is flocking-like behavior, clusters will be larger in conditions where neighbours are observable. This follows from the reasonable assumption that clusters made up of boids that actively seek out other boids in order to form a flock would generally grow larger than clusters made up of boids that coincidentally ended up close to one another.
- **Mean deviation from average heading:** Two things are expected here. First, any flocking behavior should result in a mean deviation from the average heading of less than that of a completely random cluster, which is  $0.5\pi$  radians. This might also be the case in conditions where there is no flocking, as boids might randomly cluster together while fleeing from the predator, therefore heading in a similar direction as one another.

Second, if there is flocking-like behavior, clusters in observable neighbour conditions should be more aligned than those in invisible neighbour conditions. This is because flocking-like behavior, which can only arise in visible neighbour conditions, requires a polarized flock.

#### Number of clusters

Figure 6.5 shows the total number of observed clusters for all numbers of observed neighbours. A distinction is made between the 10 and 20 present boids conditions, as the number of present boids has a large influence on the number of formed clusters.

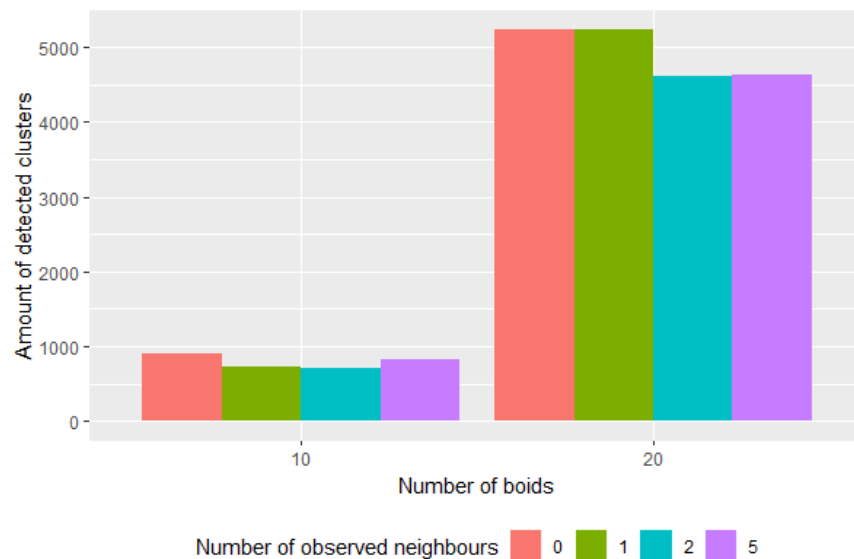


Figure 6.5: The number of observed clusters for each number of observed neighbours and each number of total present boids.

As can be seen in Figure 6.5, the expected difference between invisible and observed neighbour conditions if there was indeed flocking-like behavior is not present. In fact, the largest number of clusters observed in both the 10 and 20 present boids conditions was observed in the zero observed neighbours condition.

### Cluster size

Figure 6.6 and Figure 6.7 show the distribution of cluster sizes in the 10 and 20 present boids conditions, respectively.

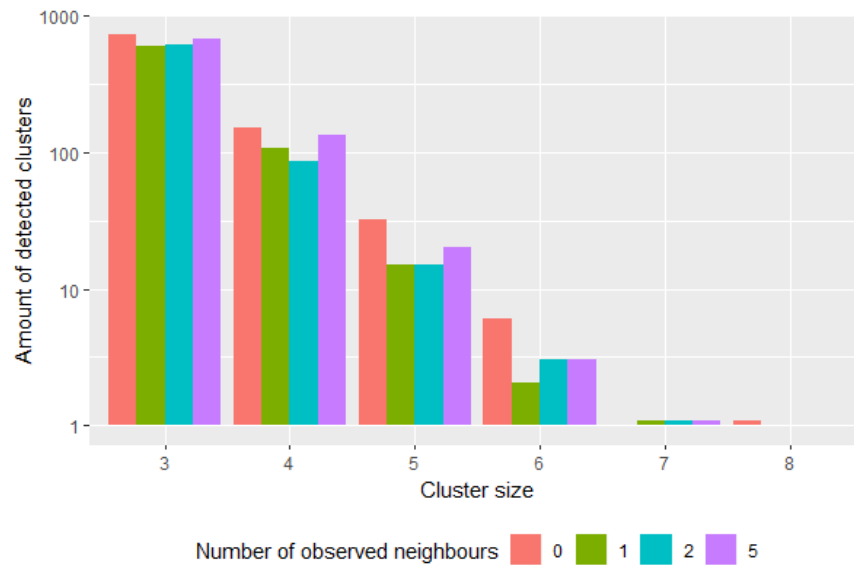


Figure 6.6: Cluster size for each number of observed neighbours with 10 total present boids (note the log scale).

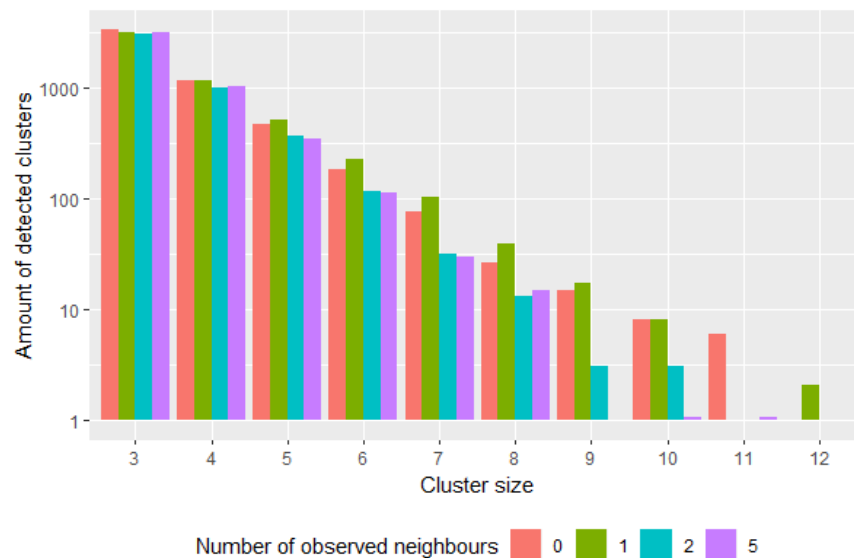


Figure 6.7: Cluster size for each number of observed neighbours with 20 total present boids (note the log scale).

If there is flocking-like behavior, this should manifest itself into larger clusters in conditions where neighbours are observable. This is not the case, as can be seen in Figures 6.6 and 6.7. The differences between the different numbers of observed neighbours are small, suggesting that whether the boids can see their neighbours does not play a role in the size of the clusters they form. This in turn suggests that there is no

flocking-like behavior.

### Mean deviation from average alignment

The distribution of the average deviation from the mean heading within clusters for the different numbers of observed neighbours is plotted in Figure 6.8.

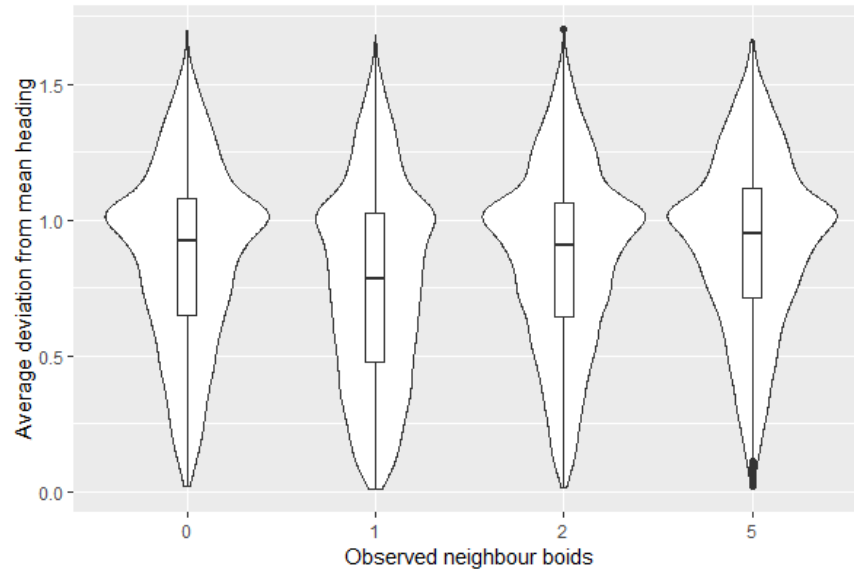


Figure 6.8: The average deviation from the mean heading within clusters for all numbers of observed neighbours. The width of each shape represents the distribution of the average deviation from the mean heading values; a wide area means more values around that area. Within each shape a boxplot is displayed. The box starts and ends at the first and third quartile, with the short horizontal line within the box marking the median value.

Flocking-like behavior should result in a lower average mean deviation from the average heading in observable neighbour conditions than in non-observable neighbour conditions. This means that the one, two and five observable neighbour conditions should have averages less than the average of 0.87 of the no observable neighbours condition. For the one and two observable neighbour conditions, which have averages of 0.76 and 0.86 respectively, this is indeed the case. The difference between the conditions is significant only for the one observed neighbour condition ( $t = 17.356$ ,  $df = 11939$ ,  $p < 2.2e-16$ ).

### 6.1.3 2D experiment 1 analysis

None of the analyses so far have provided compelling evidence for flocking. In order to completely rule out any flocking-behavior, this section will look at the measurements from a third perspective: linear regression models. Any strong correlations between flocking behavior and one or more of the parameters should become clear through this lens.

In order to determine which model parameters, if any, influence flocking-like behaviour, multiple linear regression was performed for each of the three measured factors indicative of flocking (cluster size, average pairwise distance within clusters and average deviation from the mean heading within a cluster). This section discusses the model selection process and, afterwards, presents the best model for each of the measurements indicative of flocking.

#### Regression model selection

In order to select the best possible regression model for each of the three flocking indicators, models with all possible combinations of independent variables (these are the number of boids, the predator chase time,



the boid speed and the number of neighbours observed) were made. These regression models were then compared to one another using the Akaike Information Criterion (AIC) in order to select the best model for each measurement. The AIC scores of all regression models can be found in Table 6.1 (cluster size), Table 6.2 (average pairwise distance) and Table 6.3 (average deviation from mean heading). For each variable, the regression model with the lowest AIC was picked.

Model	AIC	AIC $\Delta$
Cluster.NrOfBoids_ChaseTime_BoidSpeed_NeighboursObserved	61775.51	
Cluster.NrOfBoids_BoidSpeed_NeighboursObserved	61780.11	4.60
Cluster.NrOfBoids_ChaseTime_NeighboursObserved	61781.49	5.98
Cluster.NrOfBoids_NeighboursObserved	61786.00	10.49
Cluster.NrOfBoids_ChaseTime_BoidSpeed	61850.67	75.16
Cluster.NrOfBoids_BoidSpeed	61856.87	81.36
Cluster.NrOfBoids_ChaseTime	61857.31	81.80
Cluster.NrOfBoids	61863.41	87.90
Cluster.ChaseTime_BoidSpeed_NeighboursObserved	62174.43	398.92
Cluster.BoidSpeed_NeighboursObserved	62178.90	398.79
Cluster.ChaseTime_NeighboursObserved	62181.29	405.78
Cluster.NeighboursObserved	62185.67	410.16
Cluster.ChaseTime_BoidSpeed	62253.10	477.59
Cluster.BoidSpeed	62259.18	483.67
Cluster.ChaseTime	62260.69	485.18

Table 6.1: The AIC of all regression models predicting the size of clusters. The table is ordered based on the AIC scores of the models, with the lowest at the top. The AIC  $\Delta$  column indicates the difference between each regression model's AIC and the lowest AIC of all regression models.

Model	AIC	AIC $\Delta$
Distance.NrOfBoids_BoidSpeed	-73390.31	
Distance.NrOfBoids_BoidSpeed_NeighboursObserved	-73389.77	0.54
Distance.NrOfBoids_ChaseTime_BoidSpeed	-73388.47	1.84
Distance.NrOfBoids	-73388.39	1.92
Distance.NrOfBoids_NeighboursObserved	-73387.92	2.39
Distance.NrOfBoids_ChaseTime_BoidSpeed_NeighboursObserved	-73387.90	2.41
Distance.NrOfBoids_ChaseTime	-73386.54	3.77
Distance.NrOfBoids_ChaseTime_NeighboursObserved	-73386.04	4.27
Distance.BoidSpeed	-73320.76	69.55
Distance.BoidSpeed_NeighboursObserved	-73319.95	69.82
Distance.ChaseTime_BoidSpeed	-73318.91	71.40
Distance.NeighboursObserved	-73318.40	71.91
Distance.ChaseTime_BoidSpeed_NeighboursObserved	-73318.08	72.23
Distance.ChaseTime	-73317.30	73.01
Distance.ChaseTime_NeighboursObserved	-73316.52	73.79

Table 6.2: The AIC of all models predicting average pairwise distance within a cluster.

Model	AIC	AIC $\Delta$
Heading.BoidSpeed	15128.28	
Heading.NrOfBoids_BoidSpeed	15128.50	0.22
Heading.ChaseTime_BoidSpeed_NeighboursObserved	15131.66	3.38
Heading.NrOfBoids_NeighboursObserved	15131.88	3.60
Heading.ChaseTime_BoidSpeed	15181.01	52.73
Heading.NrOfBoids_ChaseTime_BoidSpeed	15181.06	52.78
Heading.ChaseTime_NeighboursObserved	15184.58	56.30
Heading.NrOfBoids_ChaseTime	15184.63	56.35
Heading.NrOfBoids_BoidSpeed_NeighboursObserved	15308.35	180.07
Heading.BoidSpeed_NeighboursObserved	15308.64	180.14
Heading.NrOfBoids_ChaseTime_BoidSpeed_NeighboursObserved	15309.79	181.51
Heading.ChaseTime	15310.08	181.80
Heading.NrOfBoids	15363.23	234.95
Heading.NeighboursObserved	15363.73	235.45
Heading.NrOfBoids_ChaseTime_NeighboursObserved	15364.83	236.55

Table 6.3: The AIC of all models predicting average deviation from the average heading within a cluster.

### Cluster size model

The best model predicting cluster size included all possible predictors. The coefficients of this model are listed in Table 6.4. All predictors significantly aid in the predictions. As expected, a higher number of boids appears to contribute to larger clusters ( $p < 2e^{-16}$ ). When going from 10 to 20 boids, the expected cluster size increases by approximately 0.36.

If, as expected, a more easily confused predator leads to more flocking and therefore to larger cluster sizes, the effect of predator chase time on cluster size should be negative. This means that as the predator becomes more capable of tracking a single boid without being confused by other small boids, the average cluster size shrinks. This is indeed the case in this regression model, although the effect is rather small and the p value is not as convincing as those of other predictors ( $p = 0.01020$ ). When the predator chase time decreases from 20 steps to 10 steps, the predicted cluster size increases by only 0.032.

The third significant predictor is the speed of the boids. As this varies between 0.0333 and 0.0283, the predicted difference value here is 0.035 higher for fast boids than for slow boids ( $p = 0.00475$ ). This suggests that faster boids form larger groups. This could be because they are harder to catch for the predator, which leads to the boids having cluster members eaten less often, leading to larger overall clusters.

The final predictor is the number of observed neighbours. The effect and significance of this predictor is important. Purposeful flocking can only occur if the boids can actually perceive one another. Therefore, if there actually is any purposeful flocking, it is reasonable to assume that the expected cluster size rises as more neighbours are observed. The opposite of this is actually observable in the regression model. The predicted cluster size actually goes down by 0.029 for each additional observed neighbour. This change significantly contributes to accurate predictions ( $p < 2e^{-16}$ ).

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.751078	0.085209	32.286	<2e-16
nr_of_boids	0.035881	0.001784	20.109	<2e-16
chase_time	-0.003172	0.001235	-2.569	0.01020
boid_speed	6.968226	2.467947	2.823	0.00475
nr_neighbors_observed	-0.029116	0.003312	-8.790	<2e-16

Table 6.4: Multiple linear regression coefficients on the size of clusters.

### Average pairwise distance model

The regression model predicting the average pairwise distance within clusters with the lowest AIC is given in Table 6.5. Only two predictors are used here; the number of present boids and the speed of the boids. Notably absent here is the number of observed neighbours, suggesting that boids seeing where their neighbours are does not significantly influence the distance they keep to their neighbours.

The number of boids present does have significant value in predicting the average pairwise distance between boids in a cluster ( $p < 2e^{-10}$ ). When going from 10 to 20 present boids, the expected mean pairwise distance rises by 0.0079. This is a very small amount, given that the world size in the same units is 1 by 1 and the boids move a distance of 0.0333 or 0.0283 per tick.

The second predictor in the model is the speed the boids travel at. The p value here is only just significant, at 0.0478. The predicted value of slow boids is 0,0013 higher than that of fast boids.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.1679	4.329e-03	38.788	<2e-16
nr_of_boids	7.878e-04	9.307e-05	8.465	<2e-16
boid_speed	-0.2547	0.1287	-1.979	0.0478

Table 6.5: Multiple linear regression coefficients on the average pairwise distance within clusters.

### Average deviation from mean heading model

The model with the lowest AIC predicting the average deviation from the mean heading only has 1 significant predictive factor; the speed of the boids ( $p = 0.87e^{-14}$ ). The predicted value is 0,034 higher for slower boids than for faster boids. Although the difference is quite small when compared to the intercept of 1.06, the low  $p$  value does suggest it is a real difference. A possible reason for this could be that faster boids are harder to catch for the predator. This then leads to fewer boids being eaten, meaning fewer boids are respawning at a random position. When this newly spawned boid then joins a cluster, the joining process will likely lead to a higher overall deviation from the mean heading, because a boid can only join a cluster by moving into it in another direction than the one the cluster is heading in. Faster boids that are eaten less frequently therefore lead to fewer cluster joining events, possibly keeping the average deviation from the mean heading down.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.05553	0.02766	38.16	<2e-16
boid_speed	-6.76807	0.89406	-7.57	.87e-14

Table 6.6: Multiple linear regression coefficients on the average deviation from the average heading within clusters.

#### 6.1.4 2D experiment 1 validity

This section compares two experiments run using the same model for each parameter configuration. By looking at the magnitude of the difference between the two experiments, it is possible to get a sense of how reliable the experimental procedure is. If the performance of a model and the measurements indicative of flocking differ only slightly between the two experiments, this indicates that the experiments are reliable and a good judgement of the models.

The way this has been measured is by seeing whether there are any significant differences between the results of two experiments of the same model. For this purpose, a model for each parameter configuration has been tested with an experiment twice. On each experiment pair a Hotelling's T-squared test [20] was then performed. This way the results of the two experiments are compared. The variables used to compare the experiments are the size of all found clusters, the average positional deviation within each cluster and the average deviation from the average heading within each cluster. The results of these tests can be found in Table 6.7.

Out of all experiment pairs, only two have a p-value below 0.05, indicating a significant difference. In order to determine whether this is within reason (the two  $p < 0.05$  tests could be significantly different purely

Configuration	p value	Degrees of freedom	Test statistic
0_obs_10_total_boids	0.4437581	3, 477	2.6952658640934
0_obs_10_total_boids_short_chase	0.2248029	3, 711	4.38604000609944
0_obs_10_total_boids_slower	0.02951269	3, 477	9.09692566844992
0_obs_10_total_boids_slower_short_chase	0.5964847	3, 522	1.89444170198225
0_obs_20_total_boids	0.8466731	3, 3433	0.812123137477304
0_obs_20_total_boids_short_chase	0.8208514	3, 2968	0.91956144670717
0_obs_20_total_boids_slower	0.8014116	3, 2237	1.0002337869975
0_obs_20_total_boids_slower_short_chase	0.8110781	3, 2731	0.960080962723253
1_obs_10_total_boids	0.6827462	3, 411	1.50614091843049
1_obs_10_total_boids_short_chase	0.9834854	3, 327	0.16272696030266
1_obs_10_total_boids_slower	0.7667526	3, 546	1.14713972629319
1_obs_10_total_boids_slower_short_chase	0.2544424	3, 357	4.10621161869531
1_obs_20_total_boids	0.7194327	3, 2725	1.34201867278632
1_obs_20_total_boids_short_chase	0.6291817	3, 2318	1.73676779996546
1_obs_20_total_boids_slower	0.5733845	3, 2274	1.99748355318666
1_obs_20_total_boids_slower_short_chase	0.3311882	3, 2665	3.42486280223181
2_obs_10_total_boids	0.9385077	3, 322	0.410531732195444
2_obs_10_total_boids_short_chase	0.7427536	3, 334	1.25071836275259
2_obs_10_total_boids_slower	0.1047603	3, 531	6.19821449066535
2_obs_10_total_boids_slower_short_chase	0.8632219	3, 524	0.744909666345674
2_obs_20_total_boids	0.08652763	3, 2922	6.59165494789984
2_obs_20_total_boids_short_chase	0.1160228	3, 2439	5.92178102330716
2_obs_20_total_boids_slower	0.6209664	3, 2413	1.77412094255429
2_obs_20_total_boids_slower_short_chase	0.8647807	3, 2394	0.736256561238979
5_obs_10_total_boids	0.4567889	3, 543	2.61745412177831
5_obs_10_total_boids_short_chase	0.5097093	3, 363	2.33168020697456
5_obs_10_total_boids_slower	0.09111449	3, 495	6.52551339206644
5_obs_10_total_boids_slower_short_chase	0.9976336	3, 372	0.0434881689971603
5_obs_20_total_boids	0.7167636	3, 2545	1.35341115665669
5_obs_20_total_boids_short_chase	0.1382797	3, 2811	5.51434426276737
5_obs_20_total_boids_slower	0.3372915	3, 2645	3.37942539112911
5_obs_20_total_boids_slower_short_chase	0.03396578	3, 2809	8.6913010836842

Table 6.7: Hotelling's T-squared test results for all pairs of experiments for all models in set 1.

by chance), Fisher's method [11] is used to combine all p values. This gives a combined p-value of 0.60, suggesting that there is no significant difference between the results of the first and the second experiment. This in turn indicates that the experiment captures a sufficient amount of data.

## 6.2 2D experiment 2

This section is dedicated to the results of the 2D experiment with predator pausing (for more details see Section 5.3). In order to increase the chances that flocking behavior emerges, three separate models have been trained under the same circumstances. In this section a look is taken at the behavior displayed by the boids in these experiments. This is done in order to see whether the predator pausing caused the emergence of flocking-like behavior.

### 6.2.1 2D experiment 2 boid behavior

Here the behavior observed in each of the three trained models is discussed and judged on their likeness to flocking behavior.

#### Model 1

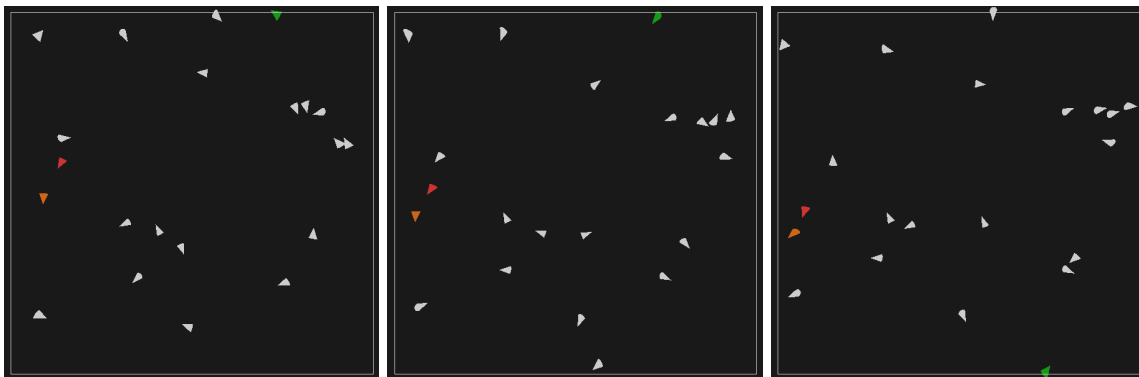


Figure 6.9: Boids showing spinning behavior except when the predator is near. The images are ordered temporally from left to right. The red agent is the predator, the orange boid is the current target and the green agent is the one trained boid. If viewed in black and white, the predator is the darkest grey, the current target is the lightest grey and the one trained boid is in between.

The behavior displayed by the first model is depicted in Figure 6.9. This behavior closely resembles the “random spinning except when the predator is near” behavior observed in the main 2D experiment and described in Section 6.1.1. The boids spin around or move randomly, but do usually react when the predator gets close. In that case, they tend to move away from the predator, as the orange boid (or, in black and white, light grey) in Figure 6.9 can be seen doing.

Just like the “random spinning except when the predator is near” behavior in Section 6.1.1 does not qualify as flocking-like behavior, this also does not. The boids do not group together, which is a requirement for flocking-like behavior.

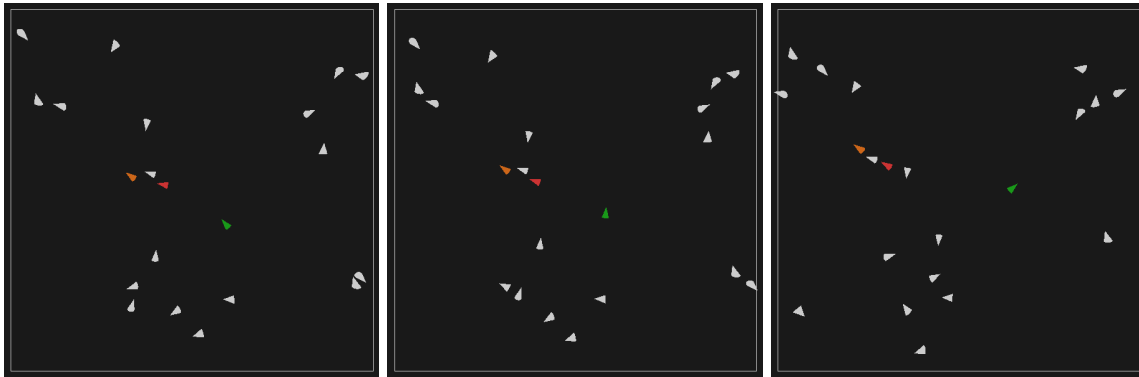
**Model 2**

Figure 6.10: Boids showing linear fleeing behavior.

Boids in the second model featuring predator pausing displayed behavior very similar to the “linear fleeing” behavior discussed in Section 6.1.1. The behavior is presented in Figure 6.10. In this experiment, boids generally did not turn, instead simply going in a straight line. When the predator got close, they often did take evasive action, which is also visible in Figure 6.10.

This type of behavior does not resemble flocking. The boids here do not group together, and seem to prefer heading in their own direction instead of aligning with other boids. Both of these things are requirements for flocking behavior.

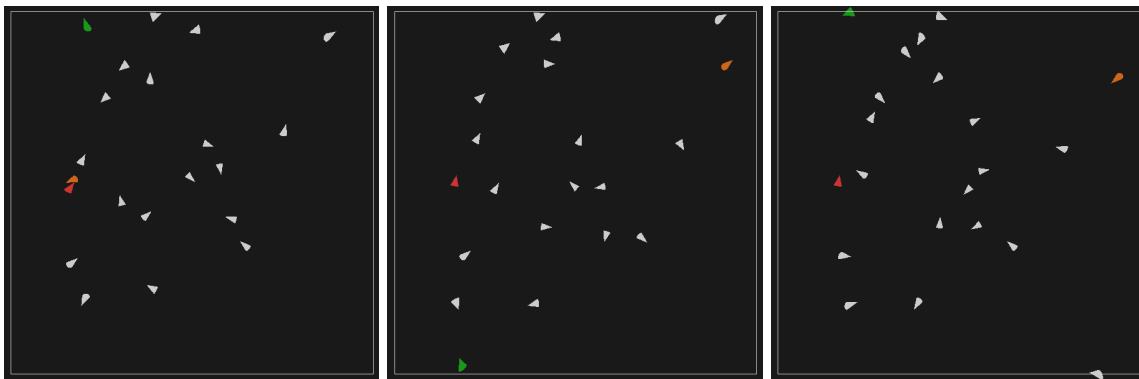
**Model 3**

Figure 6.11: Boids showing random spinning behavior.

The last model produced boids that randomly spin in place, not reacting to the predator. This resembles the behavior described in Section 6.1.1. As the boids do not appear to react to other boids (nor to the predator for that matter), this does not qualify as flocking-like behavior. No groups are actively formed and no boids are aligning with one another.

## 6.3 3D experiment

This Section discusses the results of the 3D experiment described in detail in Section 5.4. Three separate models have been trained under the same parameter conditions. An experiment was then run with each of these models. The goal of this experiment was to see whether any flocking-like behavior observed in the 2D experiments would carry over to a 3D environment.

No flocking-like behavior is expected to emerge in these experiments, because no real flocking-like behavior presented itself in the 2D conditions. Nevertheless, this experiment was run, since there was no good reason not to. The behavior which the boids did display will now be discussed. No additional statistics or figures have been provided, because these did not show anything of interest.

### 6.3.1 3D experiment boid behavior

Boids in all three trained models exhibited very similar behavior. In all cases they moved around randomly and they did not effectively react to the predator. This can also be seen in each of the examples (Figures 6.12, 6.13 and 6.14). Each of these examples features the chased boid either not moving away from the predator or being eaten by the predator.

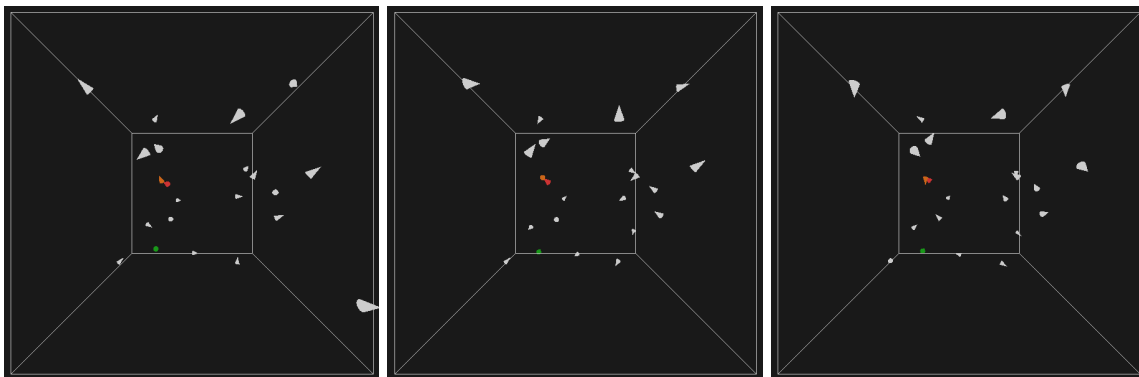


Figure 6.12: Boids showing random behavior. The images are ordered temporally from left to right. The red agent is the predator, the orange boid is the current target and the green agent is the one trained boid. If viewed in black and white, the predator is the darkest grey, the current target is the lightest grey and the one trained boid is in between.

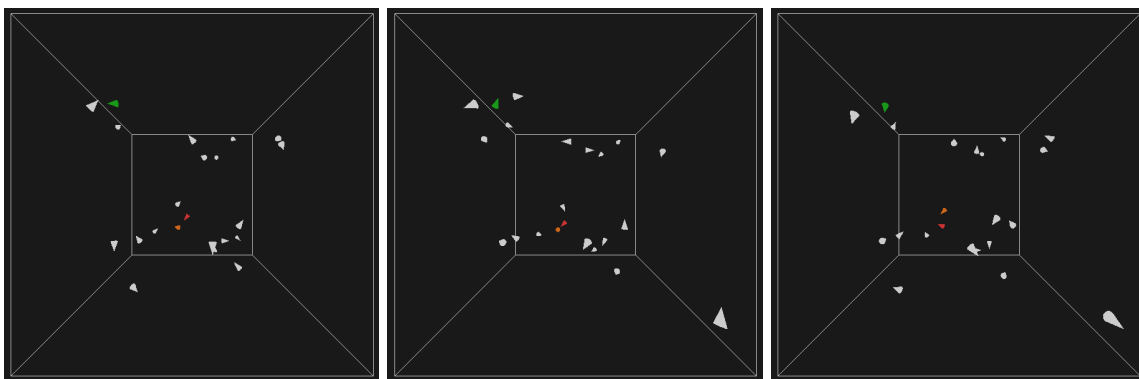


Figure 6.13: Boids showing random behavior.

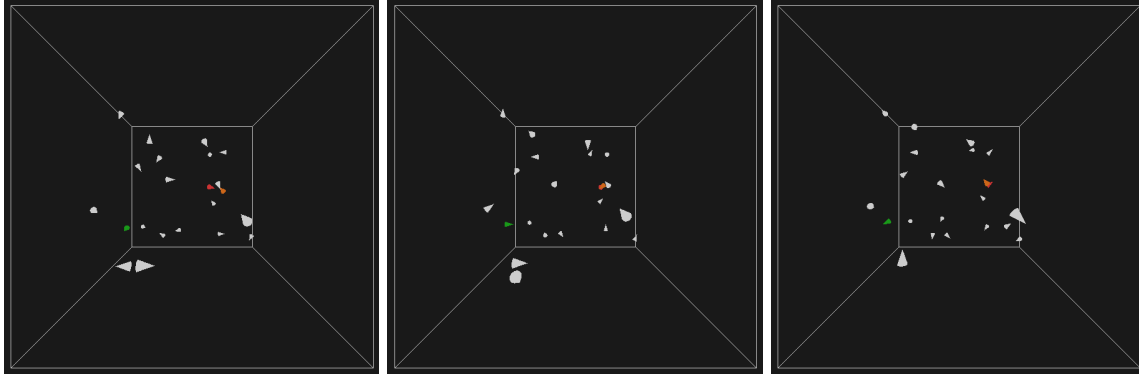


Figure 6.14: Boids showing random behavior.

No intentional grouping of boids could be observed, nor could any aligning with neighbouring boids. As expected, no flocking-like behavior occurred in any of the three models. Because the behavior of the boids is practically random, nothing useful could be learned from additional analysis.



# Chapter 7

## Discussion

This chapter further interprets the results and discusses strong points and shortcomings of the methods that produced them. It also speculates on the possible impact of several hypothetical changes to the methods, as well as possible directions for future research.

### 7.1 Interpreting the results

This section summarizes the results of each of the experiments and judges them for any evidence of flocking-like behavior.

#### 7.1.1 2D experiment 1

No convincing evidence for emergent flocking-like behavior in any of the trained models has been found. The behavior displayed by the boids generally did not resemble flocking, and the behavior which most resembled flocking (see Section 6.1.1) was also present in no observable conditions, where flocking is impossible. The closer look at the measurements taken during the experiment in Section 6.1.2 also was not convincing.

The only part in the analysis of the first 2D experiment that did hint at flocking-like behavior, the alignment between boids being significantly stronger in the one visible neighbour condition than in the no visible neighbours condition (see Section 6.1.2), should not be taken at face value. This is because the amount of data generated by agent-based modelling is large, which causes small differences in the data between two conditions to quickly be a significant difference. In the context of flocking, what we are looking for is a shift in behavior between no visible neighbours (not flocking) and visible neighbours (flocking). This shift should be accompanied by a difference in measured alignment. While there is indeed a difference in measured alignment, this difference is quite small (0.87 vs 0.76). It is significant, with a very low p value ( $p < 2.2e-16$ ), but this p value is only this low because of the large amount of data generated by the experiment. If there were indeed a shift in behavior, the difference itself would likely be larger. It is also expected that this shift in behavior would influence other measurements, like cluster size and the number of observed clusters. This is not the case.

The linear regression models fitted in Section 6.1.3 do not provide evidence for flocking-like behavior. Effect sizes are generally small, and the large significance some of them have is explainable without there being flocking-like behavior present. It stands to reason that if parameters used in the generation of data are altered, the generated data itself could be slightly different. Due to the large amount of data generated by agent-based modelling, these small differences can quickly become significant. This is no evidence for flocking-like behavior, however, as it merely means there is a slight change in behavior when certain parameters are varied. If flocking were present, there would not be a small change in behavior, but a major shift. This shift is not visible in any of the regression models.

All three explorations of the behavior displayed by the boids in the first 2D experiment did not result in any evidence for the emergence of flocking-like behavior in agents under predatory pressure.

### 7.1.2 2D experiment 2

The idea behind having the predator pause after eating was to give the boids a chance to reform any clusters broken up by a predator attack. This could help the boids flock, as they now had more opportunities to cluster together. The trained models, however, did not display any flocking-like behavior. The behavior which was observed, was also seen in the first 2D experiment. No evidence for flocking was therefore found here.

### 7.1.3 3D experiment

Although no real flocking-like behavior was observed in any of the 2D models, the 3D experiment was still performed. There was no reason not to see what behavior the boids would display. Once trained, the boids still seemed to act almost randomly. No consistent reaction to the predator was observable, and no clusters were actively being formed by the boids.

## 7.2 Reviewing the study

In this section, various encountered issues and potential limiting factors will be discussed. First, potential problems to do with the implementation of the methods are presented, and afterwards the research process itself is reflected upon.

### 7.2.1 Implementation

When replicating the methods originally used by Hahn et al [13], some of the needed parameters were not given in their paper. Although they did provide a version of their code after that was requested, this code was not final and did not include all of the missing parameters from the original paper. Final implementations of the training loop and the predator algorithm were also missing.

This means that some of the parameters used in the experiments in this thesis had to be given a guessed, reasonable value. Examples of this are the range at which the predator could be confused, and the speed and duration of the predator's periodic short sprint. The goal of this thesis was to potentially further strengthen the idea that flocking is an efficient solution to predation under a lot of circumstances. If this were indeed the case, small differences in model parameters would probably not hinder flocking; after all, flocking occurs under many different parameter conditions in nature. Still, flocking occurred in the experiments of Hahn et al. [13], and not in the experiments conducted for this thesis, which could be due to parameter choices unfit for flocking.

Another possibility the reproduction of Hahn et al. [13] was unsuccessful could be because of implementation errors. Both the 2D and 3D models contain roughly a thousand lines of code each. Although both models have been thoroughly checked for errors, it is impossible to completely rule out that the reason for the failure of the replication is a simple coding error.

### 7.2.2 Process

The methods underwent a lot of changes over the duration of the research process. Initially, it was expected that only a 3D model was necessary; once this did not return promising results in early testing, the decision to also replicate Hahn et al.'s [13] experiment was made. The effect of this is that the eventually used methods may not be optimal for the eventually set goal.

A good example of a possibly sub-optimal choice was that of which learning algorithm to use. Hahn et al. [13] made functioning models displaying flocking-like behavior through the use of both Deep Q-Learning as well as Deep Deterministic Policy Gradient. In an early version of the 3D model, Deep Q-Learning showed better results faster than Deep Deterministic Policy Gradient. This is the reason behind the choice for Deep Q-Learning. If the goal at the beginning was not just extending the flocking behavior to 3D, but also to replicate the results achieved by Hahn et al., both Deep Q-Learning and Deep Deterministic Policy Gradient might have been implemented. More examples of things that could have been done differently are discussed next, in Section 7.3.

## 7.3 Potential changes to the methods

Since the main experiment of this thesis (the first 2D experiment, see Section 5.2) was intended to recreate the experiment conducted by Hahn et al. [13], the used methods were largely identical to those used by Hahn et al. Because the original expectation was that flocking behavior would emerge (as it had when Hahn et al. did their experiments [13]), and that this would then possibly transfer into 3D, alterations that could possibly increase the likelihood of flocking or improve the quality of any emergent flocking-like behavior were not made. This section will concisely discuss several of these potential changes, as well as their expected effect.

**More present boids** With more present boids, confusing the predator purely by coincidence could occur more frequently. This might cause flocking to emerge as a potential strategy. On the other hand, more present boids means that more processing needs to be done in order to run the model. Another disadvantage is that with more present boids, the one trained boid will be attacked less often, which might lead to a lack training opportunities.

**Train using all available state-action pairs** Instead of only training one boid and using its Q-network for all boids, it should also be possible to use state-action pairs from all boids to update a shared Q-network. This would mean that there are a lot more updates per model step, which speeds up training. The speedup would probably be small, since updating the network is relatively expensive.

**Smoother turning agents** Initially, the maximum angle the boids could turn in a single step of the model was a lot lower than it ended up being. This alteration was made in order to more closely match what Hahn et al. [13] did, since the aim was to replicate their results.

Having less maneuverable, smoother turning boids, would probably lead to different behavior than what was observed in this thesis' experiments. Perhaps the lessened agility of the boids would force them to rely more on clusters and flocks for defense.

**Use DDPG as a learning algorithm** The biggest outward difference between the utilized method of Deep Q-Learning and Deep Deterministic Policy Gradient is that DDPG supports a continuous action space. This would allow for more precise movement control on the part of the boids, perhaps making flocking easier to achieve.

**Training stop condition** Since the goal of the first 2D experiment was to replicate Hahn et al. [13], their training method was also copied. In this method the boids were trained for a set number of 500.000 steps. There is, however, no reason to expect that this number of steps is optimal. Training until a stop condition is met could make more sense. Possible stop conditions are a plateau in improvement, certain behavior resembling flocking, or something different entirely.

## 7.4 Future research

This section presents various interesting direction for potential future research on the topic of this thesis.

**Training methods** Different training methods could produce very different behavior in trained agents. Apart from Deep Deterministic Policy Gradient, which has already been discussed (see Section 7.3), Neuro Evolution of Augmenting Topologies [37] would also be interesting to test. This method more closely resembles evolution in nature. It works by starting with a sparse network with the state as an input and possible actions as output. This network then, through an evolutionary algorithm, increases in capability through the additions of new nodes and connections, as well as the alteration of existing ones. By adding some sort of drawback as the complexity of the evolving network increases, increased emphasis could be placed on the efficiency of the solution.

**Pre-training** The issue of boids displaying seemingly random behavior can likely be solved through the use of pre-training. For example, boids could be pre-trained to evade a predator with no other boids present in the environment. This then leads to boids that at the very least can avoid the predator. When adding the other boids back in, it may be that the boids will now learn to evade the predator even better through working together by forming flocks.

**More/better/worse/more realistic predators** The predator is supposed to be the catalyst for flocking-like behavior in the boids. Therefore, if the behavior of the predator is changed, it can be expected that the behavior of the boids would change as well. Possible alterations to the predator are simply adding more of them, making them more or less capable, or making more realistic predators.

More predators could lead to flocking because more predation reduces the effectiveness of simply moving away from the predator that is chasing you, as this might only move you towards another predator. Multiple predators also opens the door for pack hunting tactics, which could induce entirely different behavior in the boids.

Increasing or decreasing the effectiveness of the predator could also change the extent to which flocking is a good strategy; an ineffective predator would not require flocking to evade, but a predator that is too effective might not even give the boids a chance to learn to flock. Somewhere in the middle there could be a sweet spot where flocking is optimal and will therefore emerge.

The predator can be made more realistic in a number of ways. One way this could be done through the addition of a field of view or another more accurate simulation of perception that is currently done. The effect of this is hard to predict as it depends a lot on the made alterations. A field of view for the predator might affect the behavior of the boids significantly, as prey can now “hide” behind the predator.

**More accurate/physics-based model** If the results of Hahn et al. [13] are successfully replicated and extended to 3D, it would be interesting to see whether flocking behavior would still emerge in a more realistic model. An example of such a realistic model would be StarDisplay [15], which simulates several of the forces birds in flight are subject to. If flocking is indeed a good and efficient solution to predation, it should still emerge in a model based on realistic physics.

## Chapter 8

# Conclusion

The goal of this thesis was to find out whether the 2D flocking behavior in response to a predator observed by Hahn et al. [13] could be extended into a 3D environment. If this was possible, this would strengthen the claim that flocking is induced by, and an efficient solution to, predatory pressure. In order to attempt this extension to 3D, it was first tried to replicate Hahn et al. their 2D results. This was unsuccessful.

None of the performed experiments resulted in the trained boids displaying flocking-like behavior. Giving the boids an advantage by making the predator's sprint shorter, or adding a pause after eating for the predator, did not change this. Tipping the balance in favor of the predator by making the boids slower also did not induce flocking-like behavior.

Purely based on the results of the experiments conducted for this thesis, the answer to the question "Can predatory pressure induce flocking behavior in Deep Q-Learning agents?" would be no, it cannot. No evidence for this can be found in the results of the experiments conducted with the goal to find such evidence. The observations made by Hahn et al. [13], however, still stand. This discrepancy implies that either they had a freak success, or errors were made in the attempted recreation of their results. The actual answer to the question of whether predatory pressure can induce flocking therefore is unclear.

The forces behind the convergent evolution of flocking behavior are still a mystery. Apart from predatory pressure, many different factors could potentially be the source of flocking behavior. Maybe it is foraging for food, social advantages, ease of travel, or a combination of some or all of these. Further research is still necessary in order to come to a definitive conclusion here.



# Bibliography

- [1] W. E. Allen. Behavior of loon and sardines. *Ecology*, 1(4):309–310, 1920.
- [2] Guy Beauchamp. Reduced flocking by birds on islands with relaxed predation. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 271(1543):1039–1042, 2004.
- [3] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym, 2016.
- [5] Harvey R. Bullis Jr. Observations on the feeding behavior of white-tip sharks on schooling fishes. *Ecology*, 42(1):194–195, 1961.
- [6] Claudio Carere, Simona Montanino, Flavia Moreschini, Francesca Zoratto, Flavia Chiarotti, Daniela Santucci, and Enrico Alleva. Aerial flocking patterns of wintering starlings, *sturnus vulgaris*, under different predation risk. *Animal Behaviour*, 77(1):101–107, 2009.
- [7] Will Cresswell. Flocking is an effective anti-predation strategy in redshanks, *tringa totanus*. *Animal Behaviour*, 47(2):433–442, feb 1994.
- [8] J. Michael Davis. The coordinated aerobatics of dunlin flocks. *Animal Behaviour*, 28(3):668–673, 1980.
- [9] Maxim Egorov. *Multi-agent deep reinforcement learning*. Stanford University Stanford, CA, USA, 2016.
- [10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, page 226–231. AAAI Press, 1996.
- [11] Sir Ronald Aylmer Fisher. *Statistical Methods for Research Workers*. Oliver & Boyd, Harlow, England, 1970.
- [12] Sylvia D. Gillett, Peter J. Hogarth, and F.E. Jane Noble. The response of predators to varying densities of gregaria locust nymphs. *Animal Behaviour*, 27:592–596, 1979.
- [13] Carsten Hahn, Thomy Phan, Thomas Gabor, Lenz Belzner, and Claudia Linnhoff-Popien. Emergent escape-based flocking behavior using multi-agent reinforcement learning. In *Conference on Artificial Life*, pages 598–605, 2019.
- [14] W.D. Hamilton. Geometry for the selfish herd. *Journal of Theoretical Biology*, 31(2):295–311, 1971.
- [15] Charlotte K Hemelrijk, Lars van Zuidam, and Hanno Hildenbrandt. What underlies waves of agitation in starling flocks. *Behavioral Ecology and Sociobiology*, 69(5):755–764, 2015.
- [16] CK Hemelrijk, A Costanzo, H Hildenbrandt, and C Carere. Damping of waves of agitation in starling flocks. *Behavioral Ecology and Sociobiology*, 73(9):1–7, 2019.
- [17] F. Heppner and U. Grenander. A stochastic nonlinear model for coordinated bird flocks. In *The Ubiquity of Chaos*, 1990.

- [18] Hanno Hildenbrandt, Cladio Carere, and Charlotte Hemelrijk. Self-organized aerial displays of thousands of starlings: A model. *Behavioral Ecology*, 21(6):1349–1359, 2010.
- [19] Benedict G Hogan, Hanno Hildenbrandt, Nicholas E Scott-Samuel, Innes C Cuthill, and Charlotte K Hemelrijk. The confusion effect when attacking simulated three-dimensional starling flocks. *Royal Society Open Science*, 4(1):160564, 2017.
- [20] Harold Hotelling. The generalization of student’s ratio. *The Annals of Mathematical Statistics*, 2(3):360 – 378, 1931.
- [21] Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *CoRR*, abs/1708.04133, 2017.
- [22] J. Krause, P.F.B.E.J. Krause, G.D. Ruxton, and G. Ruxton. *Living in Groups*. Oxford Series in Ecology and Evolution. OUP Oxford, 2002.
- [23] Longxin Lin. *Reinforcement Learning for Robots Using Neural Networks*. Carnegie Mellon University, Schenley Park Pittsburgh, PA, United States, 1992.
- [24] AE Magurran and TJ Pitcher. Provenance, shoal size and the sociobiology of predator-evasion behaviour in minnow shoals. *Proceedings of the Royal society of London. Series B. Biological sciences*, 229(1257):439–465, 1987.
- [25] Guillaume Matheron, Nicolas Perrin, and Olivier Sigaud. The problem with DDPG: understanding failures in deterministic environments with sparse rewards. *arXiv preprint arXiv:1911.11679*, 2019.
- [26] Manfred Milinski and Rolf Heller. Influence of a predator on the optimal foraging behaviour of sticklebacks (*Gasterosteus aculeatus* L.). *Nature*, 275(5681):642–644, 1978.
- [27] Robert C Miller. The significance of the gregarious habit. *Ecology*, 3(2):122–126, 1922.
- [28] Robin Mills, Hanno Hildenbrandt, Graham K Taylor, and Charlotte K Hemelrijk. Physics-based simulations of aerial attacks by peregrine falcons reveal that stooping at high speed maximizes catch success against agile prey. *PLoS Computational Biology*, 14(4):e1006044, 2018.
- [29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. In *Conference on Neural Information Processing Systems*, 2013.
- [30] Koichiro Morihiro, Haruhiko Nishimura, Tejiro Isokawa, and Nobuyuki Matsui. Learning grouping and anti-predator behaviors for multi-agent systems. In *Lecture Notes in Computer Science*, pages 426–433. Springer Berlin Heidelberg, 2008.
- [31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [32] Matthias Plappert. keras-rl. <https://github.com/keras-rl/keras-rl>, 2016.
- [33] Steven J. Portugal, Tatjana Y. Hubel, Johannes Fritz, Stefanie Heese, Daniela Trobe, Bernhard Voelkl, Stephen Hailes, Alan M. Wilson, and James R. Usherwood. Upwash exploitation and downwash avoidance by flap phasing in ibis formation flight. *Nature*, 505(7483):399–402, jan 2014.
- [34] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, aug 1987.
- [35] Edmund Selous. Thought transference (or what?) in birds. *Nature*, 129(3251):263–263, feb 1932.
- [36] Stewart Springer. Some observations of the behavior of schools of fishes in the gulf of Mexico and adjacent waters. *Ecology*, 38(1):166–171, 1957.



- [37] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.
- [38] Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*, 8, 1995.
- [39] Joseph H. Tien, Simon Asher Levin, and Daniel Ian Rubenstein. Dynamics of fish shoals: Identifying key decision rules. *Evolutionary Ecology Research*, 6(4):555–565, May 2004.
- [40] Arryon D. Tijmsma, Madalina M. Drugan, and Marco A. Wiering. Comparing exploration strategies for q-learning in random stochastic mazes. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016.
- [41] James R. Usherwood, Marinos Stavrou, John C. Lowe, Kyle Roskilly, and Alan M. Wilson. Flying in a flock comes at a cost in pigeons. *Nature*, 474(7352):494–497, June 2011.
- [42] Henri Weimerskirch, Julien Martin, Yannick Clerquin, Peggy Alexandre, and Sarka Jiraskova. Energy saving in flight formation. *Nature*, 413(6857):697–698, oct 2001.
- [43] Andrew J Wood and Graeme J Ackland. Evolving the selfish herd: Emergence of distinct aggregating strategies in an individual-based model. *Proceedings of the Royal Society B: Biological Sciences*, 274(1618):1637–1642, 2007.
- [44] Zhuobin Zheng, Chun Yuan, Zhihui Lin, and Yangyang Cheng. Self-adaptive double bootstrapped DDPG. In *International Joint Conference on Artificial Intelligence*, 2018.



# Appendix A

## Full results 2D experiment 1

This appendix lists a full summary of the results of 2D experiment 1. In order to limit the variance between the extend to which trained networks differ in performance, three separate models were trained and tested for each configuration. This results in three complete sets of models. From this, a ‘best set’ is then constructed. The best set contains for each configuration the model which had the lowest number of boids eaten during the experiment. This best set is what is presented in Section 6.1.

In this appendix, the experiment summaries are first grouped by the number of observed neighbours, and then grouped by the set number. Models that made it into the best set have their configuration name presented in **bold**. After all data has been listed, a summary of the best set is also given.

### Five observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
5_obs_10_boids	2653	284	3,250 (0,487)	0,171 (0,043)	0,925 (0,321)
<b>5_obs_10_boids_short_chase</b>	521	176	3,176 (0,463)	0,167 (0,046)	0,896 (0,309)
5_obs_10_boids_slower	2433	254	3,236 (0,576)	0,165 (0,046)	0,939 (0,308)
5_obs_10_boids_slower_short_chase	1457	193	3,161 (0,421)	0,174 (0,044)	0,910 (0,314)
5_obs_20_boids	7620	1302	3,483 (0,868)	0,174 (0,050)	0,905 (0,310)
5_obs_20_boids_short_chase	13392	1404	3,659 (1,043)	0,175 (0,050)	0,935 (0,309)
5_obs_20_boids_slower	14579	1323	3,638 (1,063)	0,178 (0,050)	0,942 (0,303)
5_obs_20_boids_slower_short_chase	14007	1417	3,663 (1,095)	0,178 (0,051)	0,934 (0,307)

Table A.1: The results of experiment set one for five observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
5_obs_10_boids	2227	248	3,270 (0,565)	0,174 (0,042)	0,915 (0,294)
5_obs_10_boids_short_chase	1202	174	3,190 (0,509)	0,159 (0,046)	0,840 (0,330)
<b>5_obs_10_boids_slower</b>	1944	242	3,269 (0,582)	0,167 (0,044)	0,916 (0,317)
5_obs_10_boids_slower_short_chase	923	178	3,225 (0,557)	0,172 (0,045)	0,894 (0,317)
5_obs_20_boids	6985	1092	3,474 (0,859)	0,176 (0,048)	0,939 (0,307)
<b>5_obs_20_boids_short_chase</b>	6925	1157	3,455 (0,811)	0,176 (0,046)	0,911 (0,315)
<b>5_obs_20_boids_slower</b>	6954	1098	3,492 (0,851)	0,175 (0,047)	0,921 (0,308)
<b>5_obs_20_boids_slower_short_chase</b>	11808	1202	3,562 (0,913)	0,180 (0,049)	0,942 (0,308)

Table A.2: The results of experiment set two for five observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
<b>5_obs_10_boids</b>	1391	233	3,197 (0,440)	0,167 (0,047)	0,865 (0,352)
5_obs_10_boids_short_chase	710	194	3,144 (0,367)	0,169 (0,044)	0,868 (0,297)
5_obs_10_boids_slower	2481	255	3,282 (0,594)	0,167 (0,047)	0,909 (0,300)
<b>5_obs_10_boids_slower_short_chase</b>	870	178	3,242 (0,524)	0,172 (0,044)	0,888 (0,298)
<b>5_obs_20_boids</b>	3614	1171	3,442 (0,778)	0,174 (0,047)	0,871 (0,322)
5_obs_20_boids_short_chase	10356	1601	4,082 (1,580)	0,181 (0,053)	0,922 (0,313)
5_obs_20_boids_slower	16386	1406	3,765 (1,215)	0,177 (0,052)	0,941 (0,302)
5_obs_20_boids_slower_short_chase	14939	1452	3,718 (1,157)	0,176 (0,051)	0,951 (0,298)

Table A.3: The results of experiment set three for five observed neighbour boids

**Two observed neighbour boids**

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
<b>2_obs_10_boids</b>	364	154	3,162 (0,404)	0,169 (0,042)	0,886 (0,370)
<b>2_obs_10_boids_short_chase</b>	676	185	3,157 (0,480)	0,169 (0,042)	0,847 (0,315)
2_obs_10_boids_slower	2680	275	3,331 (0,707)	0,166 (0,044)	0,909 (0,332)
2_obs_10_boids_slower_short_chase	2180	236	3,246 (0,537)	0,171 (0,047)	0,903 (0,320)
2_obs_20_boids	14534	1458	3,691 (1,089)	0,174 (0,052)	0,935 (0,303)
2_obs_20_boids_short_chase	8337	1236	3,495 (0,841)	0,172 (0,050)	0,910 (0,320)
2_obs_20_boids_slower	10295	1202	3,526 (0,934)	0,176 (0,049)	0,883 (0,317)
<b>2_obs_20_boids_slower_short_chase</b>	6561	1183	3,456 (0,800)	0,173 (0,047)	0,868 (0,322)

Table A.4: The results of experiment set one for two observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
2_obs_10_boids	536	159	3,170 (0,424)	0,164 (0,045)	0,885 (0,311)
2_obs_10_boids_short_chase	2214	241	3,207 (0,473)	0,167 (0,043)	0,914 (0,308)
2_obs_10_boids_slower	2619	265	3,275 (0,587)	0,174 (0,041)	0,919 (0,292)
<b>2_obs_10_boids_slower_short_chase</b>	792	157	3,185 (0,436)	0,169 (0,046)	0,878 (0,318)
2_obs_20_boids	8486	907	3,455 (0,837)	0,180 (0,047)	0,906 (0,308)
2_obs_20_boids_short_chase	4442	1257	3,498 (0,850)	0,172 (0,050)	0,823 (0,324)
<b>2_obs_20_boids_slower</b>	6532	1132	3,511 (0,874)	0,176 (0,049)	0,901 (0,305)
2_obs_20_boids_slower_short_chase	15629	1444	3,678 (1,175)	0,174 (0,051)	0,932 (0,306)

Table A.5: The results of experiment set two for two observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
2_obs_10_boids	722	196	3,184 (0,482)	0,166 (0,043)	0,797 (0,355)
2_obs_10_boids_short_chase	690	158	3,165 (0,405)	0,167 (0,047)	0,857 (0,307)
<b>2_obs_10_boids_slower</b>	1843	220	3,205 (0,556)	0,164 (0,048)	0,900 (0,324)
2_obs_10_boids_slower_short_chase	918	179	3,123 (0,377)	0,163 (0,045)	0,858 (0,329)
<b>2_obs_20_boids</b>	4118	1193	3,539 (0,901)	0,176 (0,049)	0,776 (0,350)
<b>2_obs_20_boids_short_chase</b>	4175	1111	3,497 (0,872)	0,174 (0,048)	0,886 (0,317)
2_obs_20_boids_slower	11279	1321	3,634 (1,035)	0,177 (0,051)	0,902 (0,316)
2_obs_20_boids_slower_short_chase	13820	1335	3,734 (1,149)	0,179 (0,051)	0,949 (0,305)

Table A.6: The results of experiment set three for two observed neighbour boids

### One observed neighbour boid

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
1_obs_10_boids	1463	214	3,290 (0,504)	0,170 (0,045)	0,878 (0,313)
<b>1_obs_10_boids_short_chase</b>	476	169	3,172 (0,512)	0,167 (0,043)	0,876 (0,312)
1_obs_10_boids_slower	2618	269	3,257 (0,537)	0,171 (0,047)	0,885 (0,303)
1_obs_10_boids_slower_short_chase	967	185	3,189 (0,491)	0,171 (0,043)	0,865 (0,317)
1_obs_20_boids	12028	1373	3,749 (1,129)	0,181 (0,051)	0,894 (0,319)
1_obs_20_boids_short_chase	12882	1168	3,603 (1,005)	0,177 (0,050)	0,929 (0,308)
1_obs_20_boids_slower	7306	1150	3,461 (0,773)	0,175 (0,048)	0,888 (0,327)
1_obs_20_boids_slower_short_chase	12250	1346	3,654 (1,058)	0,175 (0,052)	0,916 (0,301)

Table A.7: The results of experiment set one for one observed neighbour boid

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
1_obs_10_boids	823	183	3,186 (0,455)	0,168 (0,044)	0,866 (0,320)
1_obs_10_boids_short_chase	949	292	3,312 (0,639)	0,167 (0,049)	0,722 (0,357)
<b>1_obs_10_boids_slower</b>	1393	199	3,201 (0,471)	0,175 (0,041)	0,860 (0,316)
<b>1_obs_10_boids_slower_short_chase</b>	869	178	3,197 (0,464)	0,171 (0,044)	0,896 (0,308)
1_obs_20_boids	12179	1183	3,614 (1,004)	0,178 (0,049)	0,928 (0,306)
<b>1_obs_20_boids_short_chase</b>	4122	1328	3,626 (1,035)	0,175 (0,049)	0,634 (0,331)
1_obs_20_boids_slower	7898	1151	3,415 (0,787)	0,173 (0,050)	0,892 (0,309)
<b>1_obs_20_boids_slower_short_chase</b>	11038	1272	3,723 (1,153)	0,182 (0,052)	0,869 (0,329)

Table A.8: The results of experiment set two for one observed neighbour boid

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
<b>1_obs_10_boids</b>	438	175	3,251 (0,508)	0,169 (0,046)	0,858 (0,343)
1_obs_10_boids_short_chase	625	179	3,212 (0,462)	0,169 (0,046)	0,889 (0,327)
1_obs_10_boids_slower	1555	226	3,270 (0,519)	0,169 (0,046)	0,922 (0,290)
1_obs_10_boids_slower_short_chase	2296	248	3,210 (0,455)	0,165 (0,043)	0,907 (0,318)
<b>1_obs_20_boids</b>	9675	1364	3,771 (1,162)	0,179 (0,052)	0,845 (0,338)
1_obs_20_boids_short_chase	13256	1545	3,739 (1,177)	0,174 (0,052)	0,928 (0,311)
<b>1_obs_20_boids_slower</b>	7120	1265	3,655 (1,094)	0,177 (0,050)	0,617 (0,367)
1_obs_20_boids_slower_short_chase	11452	1081	3,604 (1,021)	0,186 (0,049)	0,910 (0,306)

Table A.9: The results of experiment set three for one observed neighbour boid

## No observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
0_obs_10_boids	1203	245	3,261 (0,540)	0,166 (0,047)	0,617 (0,376)
0_obs_10_boids_short_chase	1485	354	3,350 (0,644)	0,166 (0,048)	0,851 (0,356)
<b>0_obs_10_boids_slower</b>	844	170	3,171 (0,422)	0,167 (0,042)	0,889 (0,318)
<b>0_obs_10_boids_slower_short_chase</b>	1728	255	3,271 (0,569)	0,166 (0,047)	0,802 (0,353)
0_obs_20_boids	8166	1713	4,039 (1,521)	0,169 (0,055)	0,749 (0,349)
<b>0_obs_20_boids_short_chase</b>	6035	1470	3,824 (1,265)	0,174 (0,053)	0,882 (0,328)
<b>0_obs_20_boids_slower</b>	6053	1132	3,458 (0,838)	0,177 (0,047)	0,936 (0,308)
0_obs_20_boids_slower_short_chase	11738	1355	3,592 (0,974)	0,171 (0,051)	0,809 (0,341)

Table A.10: The results of experiment set one for no observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
<b>0_obs_10_boids</b>	360	156	3,205 (0,450)	0,168 (0,043)	0,903 (0,290)
0_obs_10_boids_short_chase	1821	379	3,528 (0,817)	0,163 (0,051)	0,782 (0,374)
0_obs_10_boids_slower	2816	285	3,211 (0,515)	0,166 (0,046)	0,888 (0,310)
0_obs_10_boids_slower_short_chase	2251	287	3,279 (0,572)	0,171 (0,048)	0,828 (0,332)
0_obs_20_boids	9995	1830	4,211 (1,713)	0,168 (0,059)	0,741 (0,357)
0_obs_20_boids_short_chase	10527	1295	3,559 (0,899)	0,170 (0,050)	0,869 (0,320)
0_obs_20_boids_slower	7945	1186	3,531 (0,900)	0,176 (0,048)	0,887 (0,334)
<b>0_obs_20_boids_slower_short_chase</b>	9066	1378	3,581 (0,997)	0,172 (0,050)	0,865 (0,319)

Table A.11: The results of experiment set two for no observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
0_obs_10_boids	2012	293	3,324 (0,652)	0,165 (0,044)	0,880 (0,327)
<b>0_obs_10_boids_short_chase</b>	1222	329	3,322 (0,671)	0,165 (0,047)	0,681 (0,355)
0_obs_10_boids_slower	2634	279	3,294 (0,611)	0,176 (0,044)	0,893 (0,325)
0_obs_10_boids_slower_short_chase	2446	263	3,232 (0,513)	0,168 (0,043)	0,899 (0,344)
<b>0_obs_20_boids</b>	4411	1251	3,571 (0,979)	0,173 (0,050)	0,844 (0,338)
0_obs_20_boids_short_chase	10137	1179	3,530 (0,908)	0,177 (0,048)	0,907 (0,317)
0_obs_20_boids_slower	13859	1466	3,837 (1,298)	0,176 (0,052)	0,910 (0,319)
0_obs_20_boids_slower_short_chase	14258	1394	3,666 (1,065)	0,176 (0,050)	0,924 (0,307)

Table A.12: The results of experiment set three for no observed neighbour boids

## Best performing 2D experiment results

In this section the best performing models across all three sets have been compiled. These models make up the data used in

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
5_obs_10_boids	1391	233	3,197 (0,440)	0,167 (0,047)	0,865 (0,352)
5_obs_10_boids_short_chase	521	176	3,176 (0,463)	0,167 (0,046)	0,896 (0,309)
5_obs_10_boids_slower	1944	242	3,269 (0,582)	0,167 (0,044)	0,916 (0,317)
5_obs_10_boids_slower_short_chase	870	178	3,242 (0,524)	0,172 (0,044)	0,888 (0,298)
5_obs_20_boids	3614	1171	3,442 (0,778)	0,174 (0,047)	0,871 (0,322)
5_obs_20_boids_short_chase	6925	1157	3,455 (0,811)	0,176 (0,046)	0,911 (0,315)
5_obs_20_boids_slower	6954	1098	3,492 (0,851)	0,175 (0,047)	0,921 (0,308)
5_obs_20_boids_slower_short_chase	11808	1202	3,562 (0,913)	0,180 (0,049)	0,942 (0,308)

Table A.13: The results of the best experiment set for five observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
2_obs_10_boids	364	154	3,162 (0,404)	0,169 (0,042)	0,886 (0,370)
2_obs_10_boids_short_chase	676	185	3,157 (0,480)	0,169 (0,042)	0,847 (0,315)
2_obs_10_boids_slower	1843	220	3,205 (0,556)	0,164 (0,048)	0,900 (0,324)
2_obs_10_boids_slower_short_chase	792	157	3,185 (0,436)	0,169 (0,046)	0,878 (0,318)
2_obs_20_boids	4118	1193	3,539 (0,901)	0,176 (0,049)	0,776 (0,350)
2_obs_20_boids_short_chase	4175	1111	3,497 (0,872)	0,174 (0,048)	0,886 (0,317)
2_obs_20_boids_slower	6532	1132	3,511 (0,874)	0,176 (0,049)	0,901 (0,305)
2_obs_20_boids_slower_short_chase	6561	1183	3,456 (0,800)	0,173 (0,047)	0,868 (0,322)

Table A.14: The results of the best experiment set for two observed neighbour boids

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
1_obs_10_boids	438	175	3,251 (0,508)	0,169 (0,046)	0,858 (0,343)
1_obs_10_boids_short_chase	476	169	3,172 (0,512)	0,167 (0,043)	0,876 (0,312)
1_obs_10_boids_slower	1393	199	3,201 (0,471)	0,175 (0,041)	0,860 (0,316)
1_obs_10_boids_slower_short_chase	869	178	3,197 (0,464)	0,171 (0,044)	0,896 (0,308)
1_obs_20_boids	9675	1364	3,771 (1,162)	0,179 (0,052)	0,845 (0,338)
1_obs_20_boids_short_chase	4122	1328	3,626 (1,035)	0,175 (0,049)	0,634 (0,331)
1_obs_20_boids_slower	7120	1265	3,655 (1,094)	0,177 (0,050)	0,617 (0,367)
1_obs_20_boids_slower_short_chase	11038	1272	3,723 (1,153)	0,182 (0,052)	0,869 (0,329)

Table A.15: The results of the best experiment set for one observed neighbour boid

Configuration name	Eaten boids	Clusters	Cluster size	Pairwise distance	Heading deviation
0_obs_10_boids	360	156	3,205 (0,450)	0,168 (0,043)	0,903 (0,290)
0_obs_10_boids_short_chase	1222	329	3,322 (0,671)	0,165 (0,047)	0,681 (0,355)
0_obs_10_boids_slower	844	170	3,171 (0,422)	0,167 (0,042)	0,889 (0,318)
0_obs_10_boids_slower_short_chase	1728	255	3,271 (0,569)	0,166 (0,047)	0,802 (0,353)
0_obs_20_boids	4411	1251	3,571 (0,979)	0,173 (0,050)	0,844 (0,338)
0_obs_20_boids_short_chase	6035	1470	3,824 (1,265)	0,174 (0,053)	0,882 (0,328)
0_obs_20_boids_slower	6053	1132	3,458 (0,838)	0,177 (0,047)	0,936 (0,308)
0_obs_20_boids_slower_short_chase	9066	1378	3,581 (0,997)	0,172 (0,050)	0,865 (0,319)

Table A.16: The results of the best experiment set for no observed neighbour boids