Fostering Multi-Agent Cooperation through Implicit Responsibility

Daniel E. Collins[0000−0002−1075−4063], Conor Houghton[0000−0001−5017−9473], and Nirav Ajmeri[0000−0003−3627−097X]

University of Bristol, Bristol, UK
{daniel.collins,conor.houghton,nirav.ajmeri}@bristol.ac.uk

Abstract. For integration in real-world environments, it is critical that autonomous agents are capable of behaving responsibly while working alongside humans and other agents. Although some promising models of responsibility for multi-agent systems have been introduced recently, existing models typically focus on responsibilities that are tied to some explicit shared standards and therefore fail to reflect the often unstated, or implicit, way in which responsibilities operate in the real world. We expand the conceptual framework by introducing the notion of implicit responsibilities: self-imposed standards of responsible behaviour that emerge and guide individual decision-making without any formal or explicit agreement.
We propose that incorporating implicit responsibilities into multi-agent learning and decision-making is a novel approach for fostering mutually beneficial cooperative behaviours and encouraging their adoption as social norms. As a preliminary investigation, we present a proof-of-concept approach for integrating implicit responsibility into independent reinforcement learning agents through reward shaping. We evaluate our approach through simulation experiments in an environment characterised by conflicting individual and group incentives. Our findings suggest that societies of agents modelling implicit responsibilities learn to cooperate more quickly and achieve higher individual and group rewards compared to societies that do not model implicit responsibilities.

1 Introduction

When tasked with navigating complex social decision-making scenarios alongside humans and other agents, it is important that agents can balance potential incentive conflicts, and find ways to perform their allocated role effectively whilst acting in a manner that is considered responsible and ethical by human standards [4, 11]. Existing works have outlined various facets of responsibility in multi-agent systems (MAS) [12].

Responsibility. A general definition of responsibility, outlined in [12], involves the expectation for an agent or group of agents A to realise a future state φ of the environment [5, 8].

Explicit Responsibility. Typically, responsibilities are modelled in terms of standards of behaviour that are prescribed "top-down", such as accountability for the fulfilment of allocated tasks or sanctionability for the violation of a social norm [12]. In this

paradigm, agents are responsible to the extent that they adhere to an explicit system of rules. Similarly, responsibility can be imposed through explicit agreements or commitments between agents [1, 6]. We group these treatments as *explicit responsibility*, which can always be described by "A is responsible for φ under z", where z represents the explicit source of the responsibility, which may be enforced top-down, agreed upon peer-to-peer, or otherwise entered into knowingly.

Example 1 (Explicit Responsibility). Alice adopts a puppy in the UK. By adopting the puppy, Alice has agreed to an explicit duty of care; they are aware that they are accountable for the welfare of the dog under UK law, and that adopting and subsequently neglecting a dog would violate social convention. If Alice proceeds to neglect the puppy, they may be subject to legal repercussions, or disapproval and alienation from family and friends.

Implicit Responsibility. In contrast to *explicit responsibility*, relatively little attention has been given to aspects of responsibility that emerge without any imposed standards or explicit agreement between parties. Self-imposed responsibilities can play an important role in ethical decision making among people. For example, affective responses to different scenarios and outcomes can motivate a sense of responsibility for cooperation and altruistic behaviour. We extend the conceptual framework of responsibility in MAS by introducing the notion of *implicit responsibility*: a self-imposed responsibility for bringing about some φ, that emerges bottom-up, and is internally motivated and voluntarily assumed without any external mandate, commitment or expectation.

Example 2 (Implicit Responsibility). Alice comes across a stunned pigeon near their home. Alice reasons that the pigeon will likely be in danger from a cat or a dog if left in its current state, and that they could carefully transfer the pigeon to a cardboard box and leave it to rest in a safe quiet area to recover. Alice is driven to help the pigeon by an internal sense of responsibility, although there is no explicit expectation to do so.

In Example 2, a situation emerges in which Alice becomes implicitly responsible for the fate of another entity. Even if Alice does not assume the responsibility for assisting the other entity as a goal, they are nevertheless aware that they are capable of providing that assistance, and the consequences of not doing so.

Contributions. In this work, we introduce the notion of *implicit responsibility* as an extension to the current framework of responsibility in MAS. We present a novel approach for promoting cooperation within the framework of multi-agent reinforcement learning (MARL) by operationalising *implicit responsibility* for reward shaping. We investigate our approach by conducting simulation experiments in a constrained task environment designed to incorporate well-defined implicit responsibilities. We compare the learning of cooperative behaviour by *implicit responsibility* agents to *baseline* reinforcement learning agents that do not shape rewards. We find that agents that model *implicit responsibility* learnt cooperative strategies faster, and demonstrate improved performance on the task compared to *baseline* agents.

2 Operationalising Implicit Responsibility in MAS

In MARL, reward shaping is the process of modifying an agent's reward function by introducing additional "pseudo-rewards" to guide agents towards learning specific patterns of behaviour that may not be adequately incentivised by the original reward function. Shaping rewards according to violation or satisfaction of implicit responsibility provides a novel framework for learning desirable behaviour.

For an agent A, an *implicit responsibility*, $R_A^{t_0}(\varphi_B)$, is formed with respect to another agent B if at some time, t_0, the environment state, s^{t_0} satisfies three conditions:

1. *Existence of Dependency.* Agent B's goals in a future state φ_B are contingent on the actions or resources of A.
2. *Capability to Influence.* Agent A possesses the capacity to address the needs of B and bring about φ_B through its actions or resources.
3. *Awareness or Capability of Perception.* Agent A can perceive or is capable of perceiving conditions (1) and (2) even if B does not communicate this explicitly.

Implicit responsibility describes cases in which the realisation of some state which satisfies φ_B is not possible through the actions of B alone, or from the influence of the dynamics of the environment itself. This provides binary attribution: $R_A^{t_0}(\varphi_B)$ is either *satisfied* if φ_B is realised, or *violated* if any of the above conditions are violated before φ_B is reached.

2.1 Forage Survival Simulation Environment

We designed a multi-agent grid-world environment that incorporates opportunities for well-defined implicit responsibilities for use as an evaluation test-bed. The environment supports scalable grid dimensions (M, N) and agent population size $|I|$ for an population of agents $i \in I$. For simplicity, we use an agent population of $|I| = 2$, the minimum population size to facilitate potential implicit responsibilities. The environment is illustrated in Appendix A.

In this environment, two or more agents navigate the grid with the goal of collecting berries, which provides an extrinsic reward signal as well as an in-world survival advantage. Initially, each agent $i \in I$ starts from a random position, and a berry is initialised at a random empty position for each agent in the environment.

Agent attributes Agents have two attributes which relate to their survival in the environment: (1) energy and (2) health. For an agent i, these are represented by the integers $e_i \in \mathbb{Z}^+ : e_i \in [0, E]$ and $h_i \in \mathbb{Z}^+ : h_i \in [0, H]$ respectively. Initially, energy and health are set to their maximum values, $e = E$, $h = H$.

Agent states Agents are in one of three possible states at any time, based on their attributes: (1) *Healthy*: $(e > 0, h = H)$, (2) *Helpless*: $(e = 0, h > 0)$, and (3) *Dead*: $(e = 0, h = 0)$.

Attribute decay While agents are *Healthy*, e_i decays by one per time step. While agents are *Helpless*, e_i stops decaying, and h_i begins to decay by one per time step. Agents can only take actions while *Healthy*. If the agent transitions into the *Dead* state, the agent is removed from the simulation for the remainder of the episode.

Berry collection An agent collects berry by moving to a position occupied by a berry. The agent receives a reward r_b for collecting a berry. When an agent collects a berry, a new berry is generated at a random unoccupied position.
Berry inventory Agents *store* collected berries in an inventory, where the number of stored berries is $b_i \in \mathbb{Z}^+ : b_i \in [0, B]$. When the agent is *Healthy* and $b_i < B$, b_i increases by one each time the agent collects a berry. When $b_i = B$, the agent will automatically *consume* any collected berries.
Berry consumption Agents *consume* stored berries to fully restore e_i and h_i. If an agent has $b_i > 0$ at the point of transition from *Healthy* to *Helpless* due to e decay, the agent automatically *consumes* one of their stored berries, and their health and energy are fully restored.
Survival score This represents the number of time steps an agent would survive based on their current state, with no subsequent berry collections. Agent's survival score is computed based on their energy and health: $\mu_i = h_i + e_i + E * (b_i)$.
Agent actions Agents have five discrete movement actions for navigating the environment:*up*, *down*, *left*, *right*, and *stay*. Additionally, agents have a *throw* action which removes a stored berry from their inventory, and passes it to the agent, j with the lowest *survival score*, μ_j within a Manhattan distance D. If $b_i = 0$, the *throw* fails and the berry remains in the agents inventory.
Decision module Agents automatically *consume* a berry if: (1) $h < H$ and $b_i > 0$ at the start of a time step, (2) $h_i < H$ and i has just been passed a berry by another agent, or (3) $b_i = B$ and i has just collected a new berry.
Immediate Self-Interested Incentive: Agents have an immediate incentive to act in self-interest and collect berries as quickly as possible. Further, when an agent dies, there is less competition in the environment, therefore greater potential reward per unit time for the surviving agents.
Long-Term Mutual Cooperation Incentive: The *Throw* mechanic allows *Healthy* agents to cooperate by paying an in-world cost to revive *Helpless* agents and prevent their death. We introduce a long-term incentive for mutual cooperation through the choice of environment parameters, (M, N), E, H, B, r_b, $|I|$, and D. We set these parameters such that eventually, due to stochasticity in the environment, agents will be too far away from their closest berry to collect it before their energy depletes. Long-term survival and the upper bound of total episode reward is therefore contingent on group survival and mutual cooperation. We show in Appendix B that with $D = 2$, this mechanic is true if: $H = E = N + M - 2$

2.2 Implicit Responsibility Conditions

We now apply the conditions for *implicit responsibility* described in Section 2 to the environment specification. We derive the conditions for forming, satisfying and violating *implicit responsibilities* in Appendix C. We assume that an agent i is never responsible for throwing a berry unless after doing so, they would be left with *spare effective energy*, ω_i greater than E:

$$\omega_i = e_i + E \cdot (b_i - 1) \tag{1}$$

This is the sum of the current energy and the potential energy that can be gained by consuming stored berries after a berry has been thrown. The condition $\omega_i > E$ implies

that $b_i \geq 1$, therefore an agent with no berries is never implicitly responsible for another agent. For two agents $i, j \in I$ at a time t, that are separated by a Manhattan distance of $d_{i,j}$, i has an *implicit responsibility*, $R_i^t(\varphi_j)$, with respect to j, if all of the following conditions are satisfied:

Existence of Dependency. $\psi_{i,j}(1)$ The object j is dependent on the cooperation of another agent i to reach a state φ_j from which they can continue to pursue their own rewards. Here, φ_j is any future state in which j is *Healthy*.

- True if j is *Helpless*:

$$\psi_{i,j}(1) : e_j < E, h_j > 0 \tag{2}$$

- False if j is *Healthy* or *Dead*:

$$\neg\psi_{i,j}(1) : h_j = H \qquad \text{or} \qquad \neg\psi_{i,j}(1) : h_j = 0 \tag{3}$$

Capability to Influence. $\psi_{i,j}(2)$ The subject i is capable of bringing about φ_j.

- True if either:
  - i has a spare berry and is close enough to j to *Throw* a berry to them.

$$\psi_{i,j}(2) : d_{i,j} - D \leq 0 \qquad \text{and} \qquad \omega_i \geq E \tag{4}$$

- Or
  - i has enough energy and berries to get close to j and *Throw* a berry.

$$\psi_{i,j}(2) : \omega_i > E - d_{i,j} + D \qquad \text{and} \qquad d_{i,j} - D > 0 \tag{5}$$

- False otherwise.

For $\psi_{i,j}(3)$, *Awareness or Capability of Perception*, assuming full-observability for all agents, i always has sufficient information to know if $\psi_{i,j}(1)$ and $\psi_{i,j}(2)$ are satisfied, and therefore $\psi_{i,j}(3)$ is always satisfied by default.

Once formed, an implicit responsibility is violated at the next time step in which any of the individual conditions $\psi_{i,j}(1, 2)$ are violated. We now define a fourth condition, $\vartheta_{i,j}$, for *satisfaction*, which states that if a state $s^{t'} \in \varphi_j$ is reached at a time t' after t, where no conditions $\psi_{i,j}(1, 2)$ have been violated in $[t, t']$, $\vartheta_{i,j}$ is satisfied, the responsibility $R_i^{t'}(\varphi_j)$ is closed, and i has *fulfilled* their implicit responsibility to j.

3 Simulation Experiments and Results

We conduct preliminary simulation experiments using the environment described in Section 2.1 with the parameters outlined in Appendix B, Table 1. We simulate and compare societies comprising pairs of agents, which are trained using Deep Q Learning as described in Appendix D, with hyper-parameters listed in Table 2. We train a *baseline* agent society using only the extrinsic reward signals, r_b, from berry collection. We then train an *implicit responsibility* agent society using both the extrinsic reward signals, and additional penalties using a reward shaping algorithm (Appendix D) which penalises agents for violating implicit responsibilities according to the conditions described in Section 2.2. To evaluate *implicit responsibility* agents, we measure how quickly they learn to survive through mutual cooperation compared to *baseline* agents.

Figure 1 shows the training curves for *baseline* agents and *implicit responsibility* agents. These plots indicate that *baseline* agents learn to survive through cooperation after roughly 75 thousand episodes, illustrated by a sharp increase in both rewards and agent survival times. We truncate the figure to 75 thousand episodes as beyond this point, the *implicit responsibility* agents consistently survive beyond our maximum episode length. By 100 thousand episodes, the maximum total episode reward and survival time for the *baseline* agents are still an order of magnitude lower than those reached by the *implicit responsibility* agents by 75 thousand episodes.

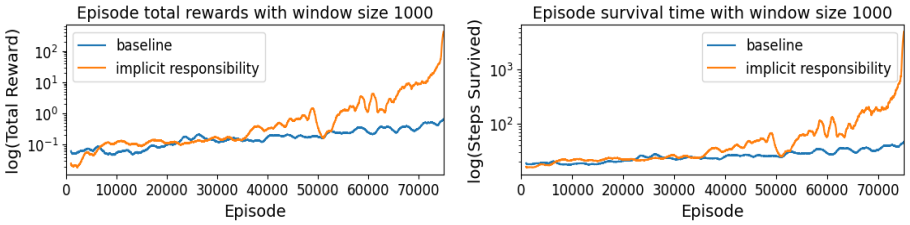

Fig. 1: Implicit responsibility agents have higher total episode reward and episode survival time.

4 Discussion

Our results are a promising indication that shaping rewards according to *implicit responsibility* can improve the speed at which reinforcement learning agents learn to exploit mutually beneficial cooperation behaviours. However, there are several limitations which must be addressed. Firstly, we only evaluate under one set of environment parameters and learning hyper-parameters. It is possible that the benefits of our approach are less significant when we compare to baseline under an optimised training protocol, or in societies of more than two agents. Further experimentation would be needed to validate our findings and assess scalability.

Further, we only test in one environment, which we designed to include easily defined scenarios for *implicit responsibility* to arise, and in which satisfaction is globally beneficial and violation is easily attributed. In doing so, we were able to test our approach by shaping rewards according to rules representing an idealised model of implicit responsibility for that environment. For application to unseen and more complex environments, agents must be designed such that they are able to approximate these rules independently. Causal attribution of responsibility and blameworthiness for outcomes are non-trivial problems [9, 12], posing a challenge for reward function design. However, our approach limits the scope to circumstances in which backward-looking attribution of implicit responsibility is subjective, without considering the relative contribution of stakeholders. This may simplify bottom-up learning of responsible behaviour.

Finally, we consider only a subset of implicit responsibilities that capture mutually beneficial outcomes, and thus neglects the role of altruism captured by other approaches for bottom-up learning of responsible behaviour [2, 3, 10].

Bibliography

[1] Dastani, M., van der Torre, L., Yorke-Smith, N.: Commitments and interaction norms in organisations. Autonomous Agents and Multi-Agent Systems (JAAMAS) **31**(2), 207–249 (Mar 2017)

[2] Deshmukh, J.: Emergent responsible autonomy in multi-agent systems. In: Proc. AAMAS. pp. 3029—3031. (May 2023)

[3] Deshmukh, J., Adivi, N., Srinivasa, S.: Resolving the dilemma of responsibility in multi-agent flow networks. In: Proc. PAAMS. pp. 76–87. (Jul 2023)

[4] Murukannaiah, P.K., Ajmeri, N., Jonker, C.M., Singh, M.P.: New foundations of ethical multiagent systems. In: Proc. AAMAS. pp. 1706–1710. (May 2020)

[5] van de Poel, I.: The Relation Between Forward-Looking and Backward-Looking Responsibility. In: Moral Responsibility: Beyond Free Will and Determinism, pp. 37–52. Library of Ethics and Applied Philosophy, Springer (2011)

[6] Singh, M.P.: Norms as a basis for governing sociotechnical systems. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(1), 21:1–21:23 (Dec 2013)

[7] Szita, I., Lőrincz, A.: The many faces of optimism: a unifying approach. In: Proc. ICML. pp. 1048–1055. ACM (2008)

[8] Triantafyllou, S.: Forward-Looking and Backward-Looking Responsibility Attribution in Multi-Agent Sequential Decision Making. In: Proc. AAMAS. pp. 2952–2954 (May 2023)

[9] Triantafyllou, S., Radanovic, G.: Towards Computationally Efficient Responsibility Attribution in Decentralized Partially Observable MDPs. In: Proc. AAMAS. pp. 131–139. (May 2023)

[10] Wang, J.X., Hughes, E., Fernando, C., Czarnecki, W.M., Duenez-Guzman, E.A., Leibo, J.Z.: Evolving intrinsic motivations for altruistic behavior (Mar 2019), arXiv:1811.05931 [cs]

[11] Woodgate, J., Ajmeri, N.: Macro ethics for governing equitable sociotechnical systems. In: Proc. AAMAS. pp. 1824–1828. (May 2022).

[12] Yazdanpanah, V., Gerding, E.H., Stein, S., Cirstea, C., Schraefel, M.C., Norman, T.J., Jennings, N.R.: Different Forms of Responsibility in Multiagent Systems: Sociotechnical Characteristics and Requirements. IEEE Internet Computing **25**(6), 15–22 (Nov 2021)

[13] Zhu, Z., Hu, C., Zhu, C., Zhu, Y., Sheng, Y.: An Improved Dueling Deep Double-Q Network Based on Prioritized Experience Replay for Path Planning of Unmanned Surface Vehicles. Journal of Marine Science and Engineering **9**(11), 1267 (Nov 2021)

A Foraging Environment with Cooperative Survival

Figure 2 shows our foraging environment.

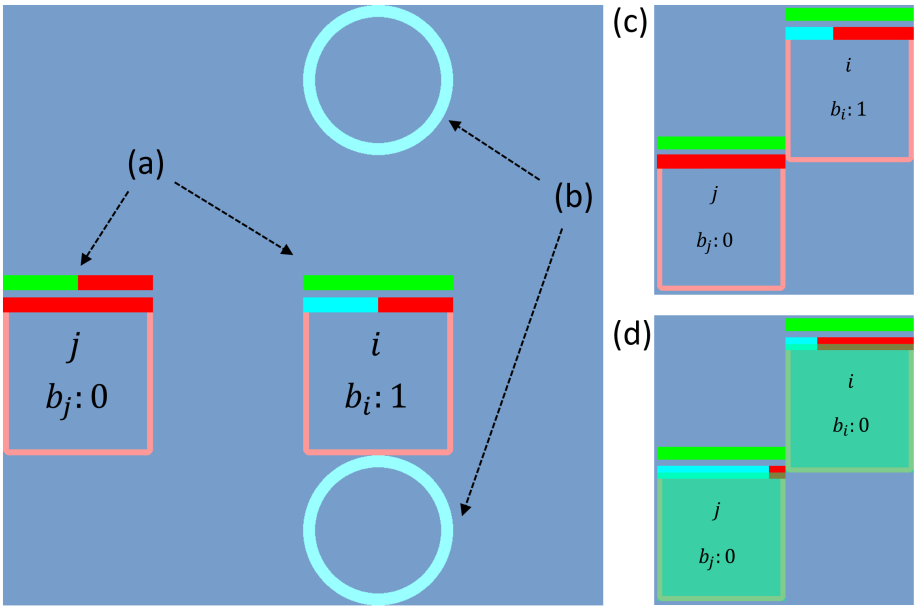

Fig. 2: (Left) Two agents i and j (a) navigate a 4×4 grid world and collect berries (b). The agents health $h_{i,j}$ and energy $e_{i,j}$ are indicated by indicated by the upper and lower bars above the agents respectively, and the number of stored berries is indicated by $b_{i,j}$. (Right) (c) In the illustrated scenario, j has $e_j = 0$, and no stored berries, and i has $e_i > 0$ and one stored berry. (d) In the next time step, i throws their stored berry to j, illustrated by the green shading, and e_j is restored.

B Environment Parameters for Mutual Cooperation

The environment parameters, grid dimensions (M, N), maximum energy E, maximum health H, maximum berry inventory capacity B, berry reward r_b, population size $|I|$, and throw distance D are chosen such that agents have the potential to survive indefinitely through cooperation, but will eventually die otherwise due to random variations in berry generation positions.

The furthest Manhattan distance between two points in the grid is

$$d_{max} = (N-1) + (M-1) \tag{6}$$

By setting E to satisfy the condition that the number of moves an agent can make from full energy is less than the number of moves required to move between the two furthest point on the grid

$$E < d_{max} \tag{7}$$

we ensure the agents have the potential to enter states from which they cannot survive without cooperation.

If we also require that the maximum initial energy is greater than the energy required to move from one corner of the grid to a point within throwing range of the opposite corner at d_{max}, and that a dying agent can survive long enough to be revived in the case described by Equation 8,

$$E > d_{max} - D \qquad \text{and} \qquad H \geq E \tag{8}$$

These conditions ensure that any "healthy" agent with full health or more than one spare berries can always reach a position from which they can throw a spare berry to a "helpless" agent and throw that berry before their energy depletes or one of their spare berry is eaten, and that the "helpless" agent can survive at least as long as the maximum time this would take.

We can reformulate 7 in terms of the integer $z \in [1, d_{max} - 1]$ to set the difficulty of independent survival by limiting the distance an agent can travel before running out of energy

$$E = d_{max} - z \tag{9}$$

The condition described by 8 then holds if D is greater than z

$$D > z \tag{10}$$

For simplicity, we set $z = 1$, $D = z + 1 = 2$ and $E = H$, such that health and energy values depend only on the grid dimensions.

$$H = E = N + M - 2 \tag{11}$$

Table 1: Default experiment parameters.

Parameter	**Default Value**
grid_size	(4, 4)
num_agents	2
max_health	8
max_energy	8
throw_mdistance	2
max_episodes	100.000
max_episode_length	10.000
steps_before_learn	10.000
update_interval	1.000
checkpoint_interval	500
episode_plot_interval	500
berry_reward	0.1
responsibility_punishment	−0.9
death_punishment	0.0

C Implicit Responsibility Conditions

Using the parameters $z = 1$, $D = 2$, $E = H$ we now derive conditions for forming implicit responsibilities between agents in the environment.

Using the Manhattan distance between two agents, $d_{i,j} = |x_i - x_j| + |y_i - y_j|$, we calculate the distance that i would need to travel to get to a position from which they could help j

- Distance to help: $d_{i,j}^{help} = d_{i,j} - D$
- Spare energy: $\omega_i = e_i + E * (b_i - 1)$

This gives the following two conditions for i to enter a responsibility event with respect to j:

i can help j from current position i has an *implicit responsibility* to throw a berry to help j if:

- i is close enough to help j

$$d_{i,j}^{help} \leq 0 \tag{12}$$

- i has enough health and collected berries to remain at full health after helping j

$$b_i \geq 1 \qquad \text{and} \qquad \omega_i > E \tag{13}$$

***i*is not close enough to help B** i has a responsibility to move closer to j if:

- i is not close enough to help j

$$d_{i,j}^{help} > 0 \tag{14}$$

- i has enough health and berries to remain be at full health after helping j

$$b_i \geq 1 \qquad \text{and} \qquad \omega_i > E - d_{i,j}^{help} \tag{15}$$

- B has enough health remaining for i to reach j and throw a berry before they die

$$h_j > d_{i,j}^{help} \tag{16}$$

D Agent Architecture and Reward Shaping Module

Here we describe a schematic of the modular architecture used for our *baseline* and *implicit responsibility* agents. In our experiments, both *baseline* and *implicit responsibility* agents are trained using using independent Deep Q learning implemented with PyTorch. Agents comprise a Deep Q-Network (DQN) architecture with with *DQN Neurons* fully connected layers. We employ experience replay [13] with a buffer size of *DQN Buffer* to stabilise the learning process. Agents explore their shared environment using an epsilon-greedy [7] exploration strategy. Table 2 lists the hyper-parameters of the learning procedure.

Algorithm 1 Reward shaping for *implicit responsibility* agents

Let i, j be agents from a population of $|I|$ agents
D be the constant Manhatten distance defining how far an agent can *Throw* a berry
Let b_t^i be the number of berries an agent i has *Stored* in their inventory at time t.
Let e_t^i and h_t^i be the integer energy and health of i at t, with maximum values E and H respectively
Let $d_t^{i,j}$ be the Manhatten distance between i and j at t.
Let ω_t^i be the spare effective energy of i at t, where $\omega_t^i = e_t^i + E \cdot (b_t^i - 1)$
Let s_t represent the full environment state at time t.
Let r_t^i be the reward to i at time t.
let $\psi_{i,j}(1) = \begin{cases} 1, & \text{if } e_t^j < E \text{ and } h_t^j > 0 \\ 0, & \text{otherwise} \end{cases}$
let $\psi_{i,j}(2) = \begin{cases} 1, & \text{if } d_t^{i,j} - D \leq 0 \text{ and } \omega_t^i \geq E \\ 1, & \text{if } d_t^{i,j} - D > 0 \text{ and } \omega_t^i > E - d_t^{i,j} + D \\ 0, & \text{otherwise} \end{cases}$
Let $R_t^{i,j}$ be the boolean representing whether i has a implicit responsibility towards j at time t, false by default.
Let p be the constant representing the penalty for violation of an implicit responsibility.
for $i \in |I|$ **do**
    for $j \in |I| : j \neq i$ **do**
        if $\psi_{i,j}(1)$ AND $\psi_{i,j}(2)$ **then**
            $R_t^{i,j} =$ True
        else
            $R_t^{i,j} =$ False
        end if
        if $R_t^{i,j}$ AND NOT $R_{t+1}^{i,j}$ AND $e_{t+1}^j = 0$ **then**
            $r_{t+1}^i = r_{t+1}^i - p$
        end if
        if $R_t^{i,j}$ OR ($R_{t+1}^{i,j}$ AND $h_{t+1}^j = 0$) **then**
            $r_{t+1}^i = r_{t+1}^i - p$
        end if
    end for
end for

Table 2: DQN hyperparameters.

Hyperparameter	**Value**
Batch Size	128
Gamma (Discount Factor)	0.999
Epsilon Start	0.9
Epsilon End	0.05
Epsilon Decay	1000
Tau	0.005
Learning Rate	0.001
Loss Function	MSE
Gradient Clipping	True