SCHOOL OF COMPUTER SCIENCE

Encouraging Ethical Behaviours in Norm-Learning RAWL-E Agents

Dhillon Thurairatnam

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering **worth 40CP**.

Thursday 10th July, 2025

Abstract

Artificial intelligence agents are increasingly deployed in multi-agent systems (MAS), necessitating effective approaches to ensure ethically aligned outcomes. This dissertation investigates encouraging ethical behaviors through norm-learning agents with explicit ethics (RAWL-E), leveraging Rawlsian ethical principles within reinforcement learning frameworks; a framework proposed by Woodgate et al [1]. Current MAS methodologies in large state space environments typically focus on maximising the overall utility of an agents individual and ethical goals together, without explicit mechanisms to enforce fairness, potentially leading to ethically suboptimal outcomes. The key aims of the project were:

- To design novel RAWL-E architectures integrating multi-objective reinforcement learning (MORL) and Deep Q-Network (DQN) methodologies.
- To build upon the MORL RAWL-E model and develop function approximation mechanisms to derive scalar weights guiding agents toward ethically preferable policies, inspired by [2].
- To evaluate these models within simulated resource-sharing environments focusing on fairness and sustainability.

My research hypothesis is that the MORL based RAWL-E model will achieve comparable fairness and sustainability to the baseline RAWL-E model, and that the Ethical Embed RAWL-E model will outperform the MORL model in fairness and sustainability. My main contributions and achievements include:

- Developing two novel RAWL-E agent models: MORL RAWL-E, and Ethical Embed RAWL-E.
- Implementing and evaluating these models across two distinct simulation scenarios: the capabilities harvest and allotment harvest.
- Demonstrating through experimental evaluation that the MORL RAWL-E model significantly improves societal fairness, specifically benefiting the least advantaged agents.
- Identifying diminishing returns in sustainability gains despite substantial fairness improvements, clarifying trade-offs between fairness and overall societal welfare.
- Evaluating the ethical embedding model with the MORL model, which revealed limitations due to overly prioritising ethical considerations at the expense of agent survival and sustainability.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others including AI methods, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

Dhillon Thurairatnam, Thursday 10th July, 2025

AI Declaration

I declare that any and all AI usage within the project has been recorded and noted within Appendix A or within the main body of the text itself. This includes (but is not limited to) usage of text generation methods incl. LLMs, text summarisation methods, or image generation methods.

I understand that failing to divulge use of AI within my work counts as contract cheating and can result in a zero mark for the dissertation or even requiring me to withdraw from the University.

Dhillon Thurairatnam, Thursday 10th July, 2025

Contents

List of Figures

List of Tables

Ethics Statement

This project did not require ethical review, as determined by my supervisor, Nirav Ajmeri.

Supporting Technologies

The research was conducted using the following technologies:

- `Python (3.12.5)`
- `Numpy (1.26.4)` library for its mathematical operations
- `Mesa (2.3.2)` library for the agent based modelling
- `Tensorflow (2.16.2)` library for the neural network implementations
- `SciPy (1.15.2)` library for its mathematical operations
- `Pyplot (3.10.1)` library for the results visualisations

Notation and Acronyms

DQN	:	Deep Q-network
MARL	:	Multi-agent reinforcement learning
MAS	:	Multi-agent system
MDP	:	Markov decision process
MOMDP	:	Multi-objective Markov decision process
MORL	:	Multi-objective reinforcement learning
RL	:	Reinforcement learning

Chapter 1

Introduction

1.1 Motivation

Artificial intelligence agents are becoming increasingly prevalent in peoples lives, making ethical considerations crucial. Multi-agent systems (MAS) and the coordination of behaviour while ensuring socially desirable outcomes in these systems presents significant challenges. A key aspect of social coordination involves the emergence and adherence to norms, which are unwritten rules that govern coordinated behaviour in a multi-agent system (MAS). They can arise bottom-up from repeated interaction or be imposed top-down by an external authority. Agents learn these behaviours through reinforcement techniques such as Q-learning, by interacting with their environment, and receiving rewards to be learnt. Integrating ethical principles, such as those derived from Rawlsian ethics [3], which emphasises fairness by prioritising the welfare of the least advantaged in society, into MAS offers a promising avenue for designing systems that promote fairness. When agents pursue only their own utility instead of an ethical utility, emergent norms may privilege some parties at the expense of others, perpetuating exploitation. Existing methods often promote cooperation by observing existing behaviours or preferences without explicitly evaluating whether these norms are ethically acceptable, risking propagating unethical norms in the society. Recent research done by Woodgate et. al [1] addresses the challenge of promoting ethical and fair norms within MAS by using normative ethics, which seeks to define how agents ought to act based on Rawlsian ethical principles. However, current approaches focus on maximising overall utility and lack robust mechanisms for learning and enforcing norms or actions that explicitly embed the fairness criteria, potentially leading to some ethically undesirable outcomes where individual groups could be disadvantaged. As a result, I extend this project by addressing the challenge of encouraging the emergence of ethical and fair norms within MAS. The central challenge lies in designing agent architectures and learning mechanisms that ensure that the agent learns to behave ethically while allowing it to pursue its individual goals. The aim of this approach is to guide MAS towards learning cooperative and ethically justifiable behaviours autonomously, especially in resource-sharing scenarios where inequalities may arise.

1.2 Objectives

The main objective of this project is to builds upon the RAWL·E framework, extend it with multi-objective deep reinforcement learning, and to design a framework on top of this to encourage ethical behaviours. More specifically, the concrete aims are as follows:

1. To design a novel agent architectures that incorporate mechanisms to learn their individual and ethical objective, using multi-objective reinforcement learning (MORL) and deep Q-learning (DQN).
2. Build a function-approximation mechanism, called ethical embedding, to derive a scalar weight to guide agents towards policies that are likely to satisfy ethical objectives.
3. Implement these models within simulated resource-sharing environments to evaluate their effectiveness in promoting fairness and sustainability across the society.
4. Assess whether these novel implementations successfully enable the learning of individual and ethical objectives and achieve an increased ethical compliance, and suggest future directions for improving the model.

1.3 Current Limitations

Currently Multi-objective reinforcement learning relies on tabular Q-learning which memory expensive when it comes to complex environments with large state spaces. The state space in the environment I am using is too large for the ethical embedding approach I adapt from Rodriguez-Soto et. al [2]. It relies on the tabular Q-learning approach with a small state space since it gives a formal guarantee about ethical behaviour so it requires that the state-action values are stored in a table.

1.4 Novelty

Our work is the first to couple multi-objective DQN with the RAWL-E environment, allowing a single network to learn reward signals for individual success and Rawlsian fairness separately. This separation allows for explicit, controllable emergence of fair norms in large-scale MAS.

The RAWL-E model that incorporates ethical embedding is the first to learn an explicit linear weight that uses Rawlsian and individual rewards, mirroring Rodriguez-Soto et al.'s [2] ethical-embedding theory inside a deep-Q-network that spans a large, continuous state space. This brings the guarantees of linear ethical embedding to domains where tabular methods are infeasible, demonstrating that the weight search and Q-value approximation can coexist within a single DQN.

1.5 Organisation

Section 2 provides background information covering multi-agent systems, reinforcement learning principles, norm emergence, and the Rawlsian ethical framework. Section 3 outlines the execution of the project, describing the implementation details of the Base RAWL-E model, the MORL RAWL-E model, and the MORL RAWL-E with ethical embedding. Section 4 presents the experimental setup and simulation results, evaluating the performance of different RAWL-E agents in resource-sharing scenarios. Section 5 offers a critical evaluation, discussing the implementation critique, model evaluation, outcomes, and possible directions for future research. Section 6 concludes the dissertation, summarising the findings and implications of the work.

Chapter 2

Background

This chapter intends to survey the contextual and technical foundations that underpin this project. It focuses on multi agents and reinforcement learning principles, through the sociological and logical theories of norm emergence, to the Rawlsian ethical framework that motivates our design.

2.1 Multi-Agent Systems

A multi-agent system (MAS) consists of multiple independent agents that make decisions and operate within a shared environment. They can make observations, take actions to modify and interact with their environment, and they have their own goals. Usually, each agent's action choice modifies other agents' behaviour [4]. Agents can collaborate to achieve a shared goal or compete when their goals conflict, or there may be a mixed goal setting with a combination of cooperative goals and competitive goals.

2.1.1 Reinforcement Learning

Reinforcement learning (RL) is a machine learning paradigm in which agents learn optimal decisions through trial and error interactions and receive feedback from their environment in the form of rewards or penalties, as seen in Figure 2.1. More formally:

Definition 2.1.1 (Reinforcement Learning)**.** *A framework for solving control tasks by building agents that learn from their environment by interacting with it through trial and error and receiving rewards or penalties as unique feedback [5].*

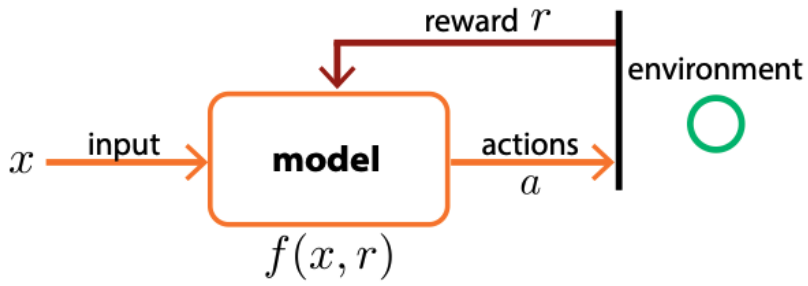


Figure 2.1: **Basic reinforcement learning setup** The model receives inputs x representing the state of the environment, selects action a, and receives feedback r in the form of rewards or penalties for these actions. Through repeated trial and error, the agent learns to optimise its action selection to maximise cumulative rewards.

By experimenting with different actions and observing the rewards received, an agent gradually improves its decision making strategy over time, which is analogous to the trial and error procedure. Agents decide the optimal decision making strategy by maximising their expected cumulative reward. This gives agents the framework for autonomous learning through interaction, relying on environmental responses from their actions.
In multi-agent reinforcement learning (MARL), the setting is non-stationary since the environment evolves

dynamically from the simultaneous learning processes of all the agents [6]. From a single agent perspective, all the other agents are continually changing their behaviour as they learn so the environment is dynamic.

Markov Decision Process

Reinforcement learning is formalised using the Markov Decision Process (MDP), which is the basis for decision making for agents. The MDP is a mathematical definition of the sequential decision problem in which the agent starts in a given state. The agent then takes an action and receives a reward, and the state of the environment transitions to another state with a given probability [7, Appendix C].

An MDP is a tuple $\langle S, A, T, R, \mu, \gamma \rangle$ which represents the following [8]:

- S: The set of states
- A: The set of actions, known as the action space
- $T : S \times A \times S \to [0, 1]$: The *transition function* which specifies , for each state, action and next state, what the probability of the next state occurring is. The probability of transitioning to the next state is only dependent on the current state-action pair regardless of the previous history, a characteristic known as the *Markov Propety* [9].
- $R : S \times A \times S \to \Re$: The *reward function* which specifies what the expected immediate reward is for each state, action and next state.
- $\mu : S \to [0, 1]$: The probability distribution over the states.
- $\gamma \in [0, 1]$: The *discount factor* specifying the relative importance of the immediate rewards.

The reward is a way for the agent to receive feedback so that it can learn over time, this is typically additive and agents try to maximise rewards over time through $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$.
The *discount factor* γ is a way for the agents to balance long term goals with short term goals. The larger γ is, the more agents care about their long term reward and vice-versa [5]. The *policy* π is essentially the brain of the agent which determines the action selection strategy it should take in its given state. Different types of policy exist, one being the *stationary policy*, which is the policy where the decision of the agent depends only on its current state. It's a mapping $\pi : S \times A \to [0, 1]$, for each state and action, and π is the probability of taking an action in that state. The *deterministic stationary policy* is the policy when the probability of choosing an action in each state is equal to 1. This is represented as a one to one mapping from states to actions $\pi : S \to A$ [8]. The goal of reinforcement learning is to find the optimal policy π which maximises the expected reward/return [10].
There are two ways an agent can learn its optimal policy, policy-based methods and value-based methods [5]. A policy-based method directly teaches the agent the action to take in its current state, without estimating a value function. The policy parameters are optimised to maximise the expected return. Value-based methods learn a value function which estimates the expected cumulative reward an agent gets for following its policy in the state it iss in. There are 2 types of value functions [5]. The *state-value function* is defined as follows:

$$V^{\pi} = E^{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid S_t = s] \tag{2.1}$$

The variable r_t is the reward at time t, and index k denotes how many steps in the future from the current time t you are looking at. This represents the expected reward when the agent starts in state s and follows the policy π onwards. Whereas the *action-value function* can be defined as:

$$V^{\pi} = E^{\pi}[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid S_t = s, A_t = a] \tag{2.2}$$

which represents the expected reward an agent gets if it starts in state s, takes action a and follows the policy onwards. So essentially the action-value function calculates the state-action value but the state-value function only calculates the value of the state.
We need to specify the behaviour of our policy learned to be *Greedy*, which means that given the value function we take the action that leads to the highest reward [5]. This is defined as:

$$\pi(s) = \underset{a}{\text{argmax}} V(s, a) \tag{2.3}$$

Q-learning

Q-learning is a reinforcement learning algorithm that learns the optimal action-value function (Q-function) for an MDP. It's a model-free method for RL, meaning that the agent does not have a model of the environment beforehand. Instead it learns exploratively, through trial and error with its actions, then learning from the choices it makes from rewards and sanctions [11]. Q-learning is a form of temporal-difference learning, which means that it uses an approach to update the value function V at each time-step. It uses a technique called bootstrapping which updates the value estimate of the current state using the immediate reward and estimate of the next state's value. This approach is useful in large state spaces where the model of the environment is not known. The Monte Carlo method on the other hand, waits for the entire episode of an environment to finish before updating the value function V [12].
At its core, Q learning is an RL algorithm which learns a Q-function (i.e. value function) represented as a table of state-action value pairs, known as the Q-table, as shown in figure 2.2. It maintains an estimate for the value $Q(s, a)$ by taking each action a in state s, and iteratively updating the estimates based on past experiences to give us a closer approximation to the optimal policy.

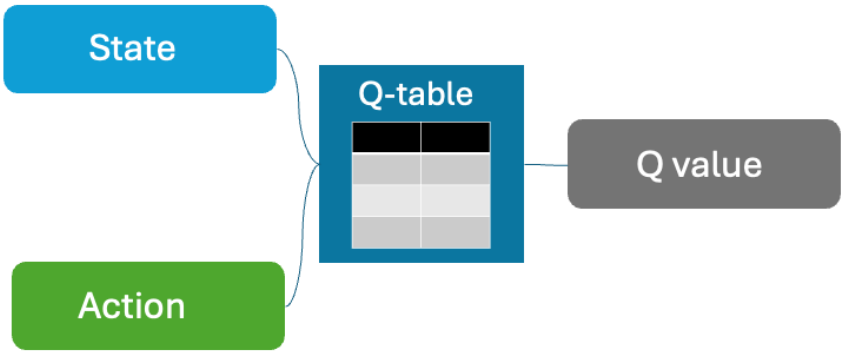


Figure 2.2: **Tabular Q-learning diagram** The Q-table stores Q-values associated with each discrete state-action pair. Given a specific state and action, the corresponding Q-value from the table can be retrieved.

Q-learning uses an *off-policy* method which means it uses a separate policy for acting (behaviour) and training. An agent can behave exploratively while still correctly learning values for an optimal-greedy policy, for example by using an ϵ-greedy policy, which balances the trade-off between exploration and exploitation of the environment. Q-learning can learn the optimal policy even while the agent is exploring randomly or learning other policies, by using the *Bellman optimality equation* as defined:

$$Q^*(s_t, a_t) = E^{\pi}[R_{t+1} + \gamma \max_a Q^*(s_{t+1}, a_{t+1})] \tag{2.4}$$

Equation 2.4 shows that $Q^*(s_t, a_t)$ is the maximum expected return from state s_t by taking action a_t and following the optimal policy. The Q-value for the state-action pair is based on the immediate reward R_{t+1} and the discounted maximum Q-value in the next state s_{t+1}. The maximisation over a_{t+1} in the next state ensures that the agent considers the best possible future rewards, regardless of the current behavioural policy. Even if the action in the next state is suboptimal, the update rule uses the maximum Q-value from the next state to back up the best possible return. Therefore, the Q-values converge towards the optimal policy [10].

In algorithm 1, the equation $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \underset{a}{max} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$ shows the update rule for the state-action pair (s_t, a_t), where α is the learning rate (step size), R_{t+1} is the reward obtained after taking action a_t in state s_t, s_{t+1} is the resulting next state, and γ is the discount factor for future rewards. We update our value function after each time step using the immediate value in the agent's current state and the estimated value of the action which maximises the current Q-function in the next state, from $R_{t+1} + \gamma \underset{a}{max} Q(s_{t+1}, a_{t+1})$, which I referred to earlier when I talked about the

optimal Bellman equation in Equation 2.4. To get the best state-action value for the next state, the agent uses a greedy policy to select the next action with the highest state-action value. Then once the update of the Q-value is done, the agent selects it's action using another policy, such as the ϵ-greedy policy. Repeating this update for many episodes of interaction will eventually make $Q(s, a)$ converge to $Q^*(s, a)$, the optimal action-value function [13].

Algorithm 1 Q-Learning Algorithm [5]

Input: Policy π, Learning Rate α, *num_episodes*, ϵ_i
Output: Value function Q
Initialise Q arbitrarily (e.g., $Q(s, a) = 0$ for all $s \in S$ and $a \in A$ and $Q(terminalstate, \cdot) = 0$)
for $i \leftarrow 1$ **to** *num_episodes* **do**
  $\epsilon \leftarrow \epsilon_i$ Observe initial state s_0
  $t \leftarrow 0$
  repeat
    Choose action a_t using policy derived from Q (e.g., ϵ-greedy)
    Take action a_t and observe R_{t+1}, s_{t+1}
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \underset{a}{max} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$
    $t \leftarrow t + 1$
  until s_t *is terminal*
end
return Q

Initially, agents start with an empty Q-table and the agent explores its environment to build the table through iterative updates, thus giving the agent a better approximation to the optimal policy [5]. Typically, the exploration strategy used will be an ϵ-greedy policy. In this policy, with a probability of $1 - \epsilon$ the agent will explore the environment by taking a random action and with a probability of ϵ the agent will exploit the environment by choosing the best action to take [14]. Initially, the value of ϵ is close to one, but as the agent explores and learns from its experiences, ϵ decreases. This allows agents to explore new outcomes while allowing it to learn optimal decisions. Over time, the Q-values "fill in" and ideally converge to optimal values, at which point the greedy policy with respect to Q is optimal.

The simplicity of Q-learning and its ability to stand out in single-agent environments has made it an influential algorithm in single-agent systems. However, basic Q learning in its tabular form faces a few problems. In complex environments with large state-action environments, tabular Q-learning would need an extremely large number of episodes to visit enough state-action combinations to learn effective behaviour [9]. This also provides a storage problem when storing many state-action value pairs in a large state space, introducing scalability problems. As a result, these issues prompted the development of more advanced Q-learning methods.

Deep Q-learning (DQN)

Introduced in 2013, by Google DeepMind, DQN uses complex models that use non-linear functions, called deep neural networks, to handle large state spaces and approximate the different Q-values for each action in a state [15], as shown by Figure 2.3. It traditionally uses convolution neural network (CNN) to stack multiple past states to provide temporal context for decision-making. Then an output vector of Q-values is produced, representing the estimated future reward for each possible action in the current state. This helps to address the problem of temporal limitation where a single state does not capture enough information about the dynamics of the environment, thus passing in a series of states captures the temporal properties of the environment [5].

One issue with using a neural network and updating the Q-value with estimates is that the value approximation might be unstable [9]. The DQN introduces two techniques to address these issues, an experience replay and a fixed Q-target [16]. The experience replay uses a buffer to store the agent's experiences and re-sampling them randomly to break temporal correlations, as shown by lines 11 and 12 in Algorithm 2. This alleviates the problem where consecutive samples in reinforcement learning are highly correlated, which can introduce instability in training when directly learning from them. By using the experience replay, breaking these correlations allows the network to learn from different stages of the

Algorithm 2 Deep Q-learning

Initialise replay memory $\mathcal{D}$ to capacity N
Initialise action-value function Q with random weights
Initialise target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
for *episode = 1 to M* **do**
  Initialize preprocessed sequence $\phi_1 = \phi(s_1)$
  for $t = 1$ *to* T **do**
    With probability ϵ, select a random action a_t
    Otherwise, select $a_t = \arg\max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t
    Set $s_{t+1} = s_t, a_t$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j \leftarrow \begin{cases} r_j, & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^-), & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$
    Every C steps update $\hat{Q} = Q$
  end
end

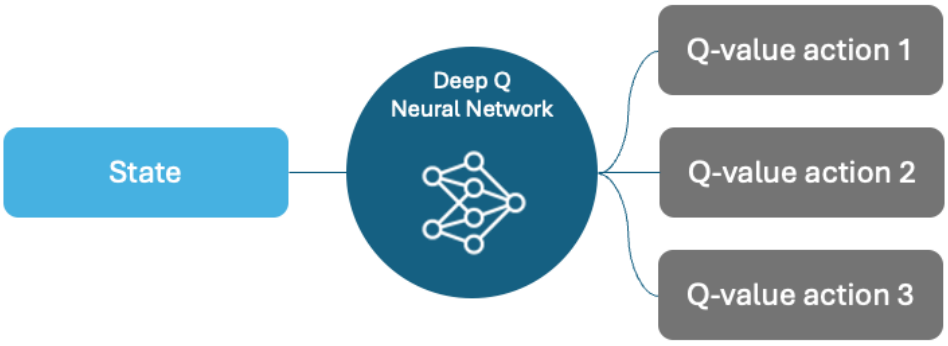


Figure 2.3: **Deep Q-learning diagram** A neural network estimates the Q-values for all the possible actions given a particular state. This allows it to handle environments with large or continuous state spaces.

environment, not just recent experiences[17].
In standard Q-learning, the target value for the Q-network is:

$$y = r + \gamma \max_a Q(s_{t+1}, a_{t+1}; \theta) \tag{2.5}$$

The weights θ used to estimate the current Q-value ($Q(s, a; \theta)$) and the target value y are the same. This leads to the problem of chasing a moving target, since the Q-values and target values shift. The fixed Q-target uses a main Q-network (Q) initialised on line 2 of Algorithm 2, which is updated at every step, and a target Q-network ($\hat{Q}$) initialised on the next line, which keeps the weights constant for several iterations [5]. Instead of estimating the Q-value and target value using the same network, the target Q-network computes the target value, as shown by Equation 2.5, then the weights of the target network are updated every C steps by copying the Q-network onto the target network, shown by the final line in Algorithm 2. Since the target network is not being updated at each step, this mechanism reduces instability as updates to the Q-networks weights are smoother. With these innovations, DQN was able to learn to play Atari 2600 video games directly from pixels, achieving human-level or better performance in many games, a milestone reinforcement learning [16].

2.1.2 Multi-Objective Reinforcement Learning

In Section 2.1.1 of Reinforcement learning, I discussed single objective MDPs and their corresponding value functions. Unlike the single-objective value function (Equation 2.2), where the sum of rewards is a scalar, the multi-objective value function represents the sum of rewards as a vector. The reward function in a multi-objective MDP (MOMDP) is defined as: $R : S \times A \times S \rightarrow \Re^n$, where $\Re^n$ is the vector of n rewards [18].
In a single objective MDP it is possible to have a partial ranking of the state-value function since policies might be better in one state but worse in another. For a single state, the ranking is complete, in other words, the value function for one policy must be greater than, equal to, or less than another policy's value only. For example, $V^{\pi}(s) > V^{\pi'}(s)$ can be possible, as well as $V^{\pi}(s') < V^{\pi'}(s')$. However, only $V^{\pi}(s) > V^{\pi'}(s)$, $V^{\pi}(s) = V^{\pi'}(s)$ or $V^{\pi}(s) < V^{\pi'}(s)$ can be possible. In MOMDP this process is more complicated since the value function is a vector of objectives instead of a scalar. This means that even for a single state, partial ordering may be possible. For example, $V_i^{\pi}(s) > V_i^{\pi'}(s)$ may be possible as well as $V_j^{\pi}(s) < V_j^{\pi'}(s)$. Here policy π is better in one objective but underperforms in another, so we can't compare policies easily in this scenario, even in the same state. As a result, we need a *scalarisation function* to combine objectives into scalar values, allowing us to rank policies and obtain an optimal policy [8].
Sutton's reward hypothesis states that "all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)" [19]. This implies that MOMDP's can always be converted into a single objective MDP. This process involves transforming multiple objectives into a single scalar reward, making standard MDP methods applicable.

In MOMDP, we follow the *utility-based approach* which selects policies by collapsing the value vector into a scalar utility using a scalarisation function.
Any policies that are never optimal under any scalarisation function are excluded. A policy is considered suboptimal if, no matter how the objectives are weighted, another policy performs better. The optimal policy can be formally defined by the definition below [8]:

Definition 2.1.2 (Undominated Policies)**.** *For an MOMDP m and a scalarisation function f, the set of undominated policies, $U(\Pi_m)$, is the subset of all possible policies Π_m for m for which there exists a w for which the scalarised value is maximal:*

$$U(\Pi_m) = \{\pi : \pi \in \Pi_m \wedge \exists w \forall (\pi' \in \Pi_m)\ V_w^{\pi} \geq V_w^{\pi'}\} \tag{2.6}$$

Definition 2.1.2 states that a policy is undominated if π belongs to the set of all policies and there exists a weight vector $\boldsymbol{w}$ such that, for all other policies π', the scalarised value is at least as high as that of any other policy π'. This means that the set $U(\Pi_m)$ consists of all policies that are optimal for at least one set of objective weights, so there is some preference weighting w where the policy in $U(\Pi_m)$ is the best choice.
Not all undominated policies are necessary since some two or more different policies can be optimal under the same weight settings. This means they never provide a unique advantage and can be safely removed without affecting the solution set. To solve m, we need only a subset of the undominated policies

such that, for any possible weight w, at least one policy in the set is optimal. This is called a *coverage set*.

Linear scalarisation functions f are the inner product of the weight vector $\boldsymbol{w}$ and a value vector:

$$V_\pi \cdot w = w \cdot V_\pi \tag{2.7}$$

The *convex hull* is the smallest coverage set that contains all undominated policies in a multi-objective setting. If a policy is outside of the convex hull, then it is always worse than a combination of other policies and should be discarded. In more formal terms:

Definition 2.1.3 (Convex Hull (CH))**.** *The subset of* Π^m *for which there exists a* $\boldsymbol{w}$ *for which the linearly scalarised value is maximal:*

$$CH(\Pi_m) = \{\pi : \pi \in \Pi_m \wedge \exists w \forall (\pi' \in \Pi_m)\ w \cdot V^\pi \geq w \cdot V^{\pi'}\} \tag{2.8}$$

Figure 2.4 shows different policy points represented as a value with two-dimensions, $V[0]$ and $V[1]$. The policies filled in with black circles form the convex hull and they are joined by a line forming a band around the rest of the points. The points that form the convex hull aren't dominated by others as there exists a weight vector $\vec{w}$ which makes each policy the best overall policy according to the scalarisation function $f(\vec{V}, \vec{w}) = w_0 \cdot V[0] + w_1 \cdot V[1]$. The white points are dominated by the black points so there is no weight $\vec{w}$ that can make a policy better than a given black point in either objective.

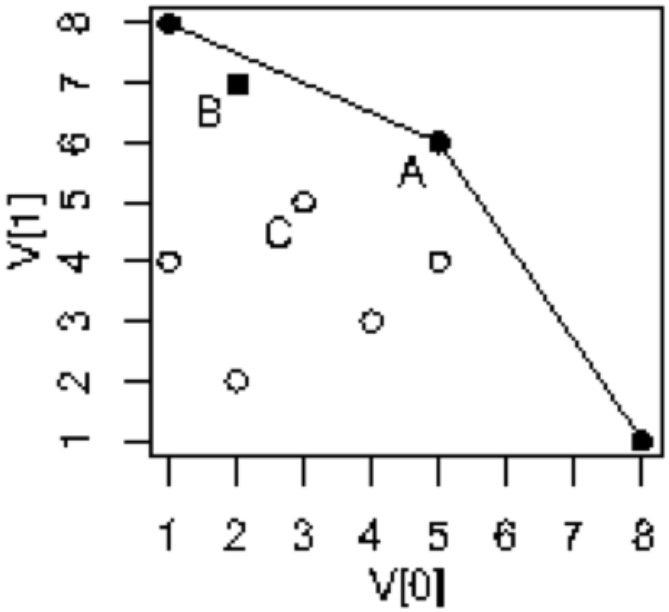


Figure 2.4: **Example convex hull graph** Shows multiple policy points in a two-objective value space $V[0]$ and $V[1]$. Policies filled in black circles lie on the convex hull and represent the non-dominated set of optimal trade-offs. Each of these policies can be made optimal by some scalarisation of the objectives. White points lie below the convex hull so no linear weighting can make them optimal. Taken directly from Roijers et al. [8].

2.2 Norm Emergence

Norms are patterns of behaviour that are widely practised and accepted within a society. From a sociological approach, norms are the evaluation of behaviour and sanctions. A *Normative MAS* is the overlap between normative systems and MAS. Institutions are seen as the structures of social order which governs the behaviour of agents [20]. According to [21] agents choose whether to comply with explicitly defined norms but the system also establishes rules and limitations on how agents can influence or modify these norms. From a philosophical approach, deontic logic is explored, which is a formal system for analysing obligation, permission and prohibition known as *normative expressions*. Norms can be classified as *substantive*, which defines legal relationships within a society or as *procedural* where the regulations are

connected to rewards and punishments [22].

In Normative MAS there are 2 perspectives to how norms are established in societies [23]:

- *Top-down* - This is where a centralised designer defines norms statically or dynamically, though an agent role model or leader.
- *Bottom-up* - This approach is aligned with the interactionist perspective on social norm evolution rather than pre-imposed rules. Norms emerge from agent interactions or observations which leads them to adopt behaviours. Since norms are shared through a society, a social mechanism for ensuring compliance needs to be built.

The top-down approach is expensive and impractical in regards to scalability unlike the bottom-up approach, which can adapt to different norms. In the bottom-up approach, Conte et al. [24] defines norm emergence as a macro level effect from agent interaction at the micro level where agents learn and internalise norms though a cyclic process between emergence and accepting the norms. In another viewpoint, Boella et al. [25] views emergence as a social delegation cycle where agent desires lead to the formation of group goals which turn into social norms and agents accept the norms.

For norms to emerge from a source [26], a mechanism needs to exist to allow for norms to propagate through a society. Savarimuthi et al. [27] uses a role model within the society where some agents who are seen as role models are more influential than others. Agents can seek advice from role models, but they can also retain autonomy by accepting or rejecting advice. Norm entrepreneurs [28] have been used who are agents that suggest new norms to other agents, and they decide whether to keep or replace the worst performing norm. However, the norm entrepreneurs can manipulate norms, potentially destabilising a society. Agents can also learn through interactions [29] where they learn norms through repeated interactions with each other and adjust their strategies and behaviour based on coordination success.

Norm Implementation

Criado et al. [20] discuss 2 mechanisms of how norms are implemented in a MAS, (i) *regulation* makes the violation of norms impossible, however, this restricts the autonomy of agents [30]. Bench-Capon and Modgil [31] contend that this should not be classified as one that operates based on norms. (ii) *Enforcement* is a mechanism that identifies norm deviations and takes action.

- *Self enforced norms* - In this mechanism agents observe their own behaviour and punish or reward themselves.
- *Second party enforcement* - Here agents involved in an interactions check the behaviour of other agents and apply sanctions when necessary.
- *Third part enforcement* - Here, an authority that acts as a judge makes decisions in disputes over social norms. This can include an agent involved in an interaction or not, acting as an institutional enforcer.

Norms can be viewed from 2 perspectives, *deontic/prescriptive* or *emergent* [29]. In the *prescriptive* approach, norms are influenced by law and contractualism, which is when all the agents agree on conventions [32]. There would be a central authority governing the society with explicit or codified norms, these norms are usually enforced in an authoritarian approach as being mandatory with sanctions clearly defined. However, this approach is limited in flexibility when agents change to different environments. In the emergent approach, norms are based on conventionality and decentralised coordination [33]. By this I mean that norms that develop through interactions within a social group are implicitly learned and adopted voluntarily based on social approval and sanctions allowing for high-adaptability for the agents. A hybrid approach combining elements of the *emergent* and *prescriptive* approaches has also been employed, which uses a centralised framework with decentralised norm emergence [29]. Some work on this includes Tzeng et. al [34], where norms are introduced through explicit reasoning with predefined norms, via the prescriptive approach, but the adoption of norms is spread dynamically through agent interaction and reinforcement learning.

Norms can also be represented in agent societies as *explicit*, where agents have an internal representation of their beliefs within their cognitive structure or in an external knowledge source that acts as a

normative system. The norms can be updated dynamically but require storage and management. These societies of agents are usually interactions which are governed by norms explicitly defined as permissions, prohibitions, or obligations [35]. They can also be represented as *implicit*, where norms aren't explicitly stored; there is no internal representation of the norms within the agent since they emerge naturally as agents learn and adopt common behaviours over time through interactions or observations [36]. Agents don't recognise norms but just learn behaviours through mimicking, machine learning mechanisms or other approaches [36]. This approach does raise some doubts about whether a norm has truly "emerged" if agents don't consciously recognise norms, however norms can also be defined as common behaviours observed within a population above some threshold [26]. Implicit norms are common in preference behaviour, whereas explicit norms are shown in deontic norms.

According to the how norms emerge, norms can be divided into several levels as shown by 2.1.

Norm Level (Criado et al.) [20]	**Norm Level (Cordova et al.) [29]**	**Explanation**
Social Level (Conventions)	s-norms (Social Norms)	Emergent norms arising from agent interactions, informally enforced through social sanctions. Not formally codified.
Interaction Level	i-norms (Interaction Norms)	Explicitly created norms through repeated interactions, clearly specifying obligations and consequences.
Social Level (Institutional)	r-norms (Institutional norms / Laws)	Explicitly defined and enforced norms created by authorities or institutions, involving sanctions or punishments upon violation.
Private Norms	m-norms (Personal/Moral norms)	Norms emerging from agents' internal principles, motivations, and values, ensuring individual autonomy without external enforcement.

Table 2.1: **Comparison of norm levels.** The first column lists the four norm levels as classified by Criado et al. [20], the second column shows the corresponding categories described by Corvoda et al. [29]. The third column provides a concise explanation of each level ranging from emergent social norms to privately held moral norms.

Norm Reasoning

Norm reasoning is all about how agents think about norms, reason about how they resolve norm conflicts and prioritise individual goals against norms. Decision making occurs at runtime to regulate the agent's behaviour in MAS. Agents adopt and comply with norms while deciding on what goals to pursue. An agent usually chooses a goal and then selects an action based on its compliance with norms, thus balancing norm compliance with achieving goals in an environment. Some early work done on how norms impact an agent's actions is explored by Castelfranchi [37]. The paper considers not only how norms influence behaviour but also the mental attitudes of agents. Castelfranchi [37] also describes 4 types of norm adoption mechanisms. These include *unconditional* adoption, where agents follow norms without making decisions, *instrumental* adoption, where agents follow norms only beneficial to themselves, also known as egoism, *benevolent* adoption, which is when agents adopt norms to favour specific individuals, and *cooperative* adoption, which is when agents comply with norms that benefit society as a whole. More information on the ethics of norms can be found in Section 2.3.

Several abstract architectures exist which models agents' decision making processes when considering norms to adjust their goals. These architectures consider how norms can be integrated into the agent's reasoning process [20]. In the *Belief Desire Intention* (BDI) [38] based models, norms are represented as preferences or desires with the agents mental state, which is one way to model an agent's normative reasoning process.

Given this and Section 2.1.1, agents will adhere to norms to avoid punishment and increase their payoff. If the payoffs are equal for any group of actions, it would be difficult for agents to reach a norm convergence, which was shown by Hu and Leung [39] where actions with equal payoff had a variety of actions that emerged within small societies and no global convergence.

The cognitive ability of agents is used in norm emergence to learn and reason about their environment, simulating human social behaviour [29]. Their reasoning abilities can be classified into 3 levels:

- Low-level reasoning: This involves a basic action selection using a fixed action or a random choice.
- Medium-level reasoning: Agents choose actions based on the context in a given scenario.
- High-level reasoning: Advanced autonomous decision making where agents can reason about their environment, norms and actions available to it.

Different approaches to learning have been employed in several papers. Some learning algorithms include imitation learning where agents replicate the behaviour of their peers, however, this approach struggles with learning multiple norms in a complex environment [40]. Reinforcement learning makes use of learning from past events through rewards and punishments [5], as discussed in Section 2.1.1.

Norm Life Cycle

To understand how norm emergence occurs, it is important to study the norm life cycle process. The life cycle of a social norm represents its dynamic nature and ability to adjust in response to societal changes [29]. A five-stage norm life cycle process has been proposed [41] with the following stages: (i) creation: a mechanism that establishes a norm, (ii) recognition: when agents become aware of the norm and it becomes part of their values and beliefs, (iii) dissemination/propagation: the process by which the norm spreads through a population, (iv) enforcement: when violators face sanctions to enforce compliance, and (v) emergence: the stage when the norm becomes widely accepted in the society. This is a process where a population of agents reaches a specified threshold of individuals adopting the same norm [26]. In [29] they introduce 2 extra stages at the end: (vi) forgetting: which is when norms disappear if they are no longer useful in an environment, and (vii) transformation: where in contrast to the phase (vi) norms don't disappear but go through reconstitution and maintenance [42].

The norm life cycle has some differences between the emergence of implicit norms and explicit norms as discussed in [26]. The norm life cycle in societies with implicit norms follow the stages of creation, dissemination and finally emergence. In the creation phase, norms are made from the initial predetermined strategies of agents from their available actions. Then dissemination phase involves the norms being learnt and spread through interactions or observations between agents. As this is implicit norm emergence, agents aren't aware that their behaviour influences others. There are 3 ways in which norms can be spread through a society [43], *vertical transmission*, which is when norms are spread from parents to their children, *oblique transmission*, which is when norms are spread from leaders to followers (role models or norm entrepreneurs), and *vertical transmission* which is when norms are spread through peer-to-peer interactions (interaction learning). I'll be focusing more *vertical transmission* later on in the paper.
With explicit norms, the stages for the norm life cycle are similar, however, the stages of recognition and then adoption follow from norm creation. In the adoption stage, agents will evaluate whether they should adopt the norm or not based on norm conflicts [44] and their goals. In both cases, enforcement mechanisms can be used to discourage violation through penalties and encourage emergence through rewards.

2.3 Rawlsian Ethics

John Rawls provides a philosophical framework with two fundamental principles of justice. One is the basic right to individual liberties for all citizens and the second is a principle governing social and economic

inequalities. The latter principle is known as the difference principle, which specifies that inequalities are only permissible if they are arranged to the greatest benefit of the least advantaged in society [3]. In other words, this principle uses the maximin criterion, which maximises the minimum welfare of individuals in a society. This emerges from Rawls's thought experiment where individuals would ignore their own future status and adopt a maximin strategy to protect those who might end up worse off [45].
This idea of fairness to uplift the least advantaged has been influential in MAS and AI where it has provided a normative framework for designing ethical MAS environments. Rather than the utilitarian approach of maximising the overall utility, a Rawlsian approach would use an ethical function or element which maximises the minimum individual reward or utility. This idea embodies the egalitarian position where all individuals in a society deserve equal opportunities [46]. Rawlsian ethics can be framed as a fairness optimisation criteria allowing agents to favour more equitable outcomes [47].
To operationalise this principle into AI, MAS agents require global awareness of each individuals outcomes. Agents must evaluate which group of agents are currently the worst off and then adjust their policies accordingly. Several researchers have implemented Rawlsian ethics in MAS. Zhang and Shah [47], develop a fairness optimisation solver for a 2 player zero-sum game which explicitly optimises a Rawlsian criterion. Their approach computes a joint policy which maximises the minimum discounted reward. This policy is seen to be Pareto-efficient (i.e no one can be made better off without making someone else worse off). Zimmer et al. [48] addresses the challenge of learning fairness policies within MARL. Fairness is conceived as two aspects, efficiency and equity. Efficiency is seen as maximising collective rewards whereas equity is seen as ensuring a fair distribution of the rewards among the agents. This approach however does not capture the idea of learning social norms.

Chapter 3

Project Execution

In this section I will discuss how the concepts mentioned in the background are used to create novel agent implementations that use Rawlsian ethics to encourage ethical behaviours in MAS. `Python` was chosen as the programming language for developing the models due to its ease of use and large ecosystem of library support for simulation data analysis and machine learning tasks. The following libraries were utilised to support the development process:

- `Numpy` for aditional mathematical capabilities.
- `Mesa` for an agent-based modelling framework to build, simulate, and analyse agent-based environments.
- `Tensorflow` for implementing the neural network models which underpin the Q-learning mechanism.
- `Pyplot` to plot graphs.

Imagine a university group project where a group of students of different backgrounds and different time commitments decide to complete this project. The group decides to divide the workload equally, and the norm is that everyone contributes the same amount. However, it becomes clear that some students fall behind due to external pressures and difficulty in understanding the material. Rather than sticking to a rigid equal workload rule, the group decides to redistribute tasks so that members with the stronger subject knowledge or more free time take on more technical responsibilities, while struggling individuals focus on other tasks like documentation and coordination tasks that they can handle better. The redistribution is seen as more fair since it makes sure the least advantaged members are supported, which can hopefully lead to better group coordination which is beneficial for everyone. This approach makes sure that norms in a system are fair and meet everyone's needs with the aim of promoting fairness.

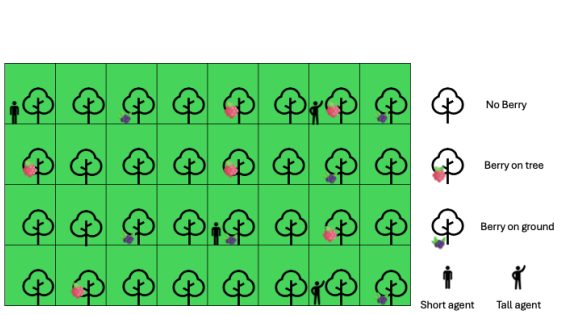
3.1 Base RAWL-E Implementation

This model is an implementation adapted from Woodgate et al. [1]. RAWL-E agents learn norms and decisions that take into account their own individual objectives and collective societal objectives to create a society that encourages a higher minimum well-being, as discussed in Section 2.3.

3.1.1 Environment

The environment E consists of the tuple $\langle AG, B, N \rangle$, with $AG = ag_1, ..., ag_n$ being the set of n agents, B is the total amount of berries (resources) in the environment available and N is the set of norms.
A RAWL-E agent is formally defined as the tuple $\langle b, wb, G, A, Z \rangle$, where $b \in B$ represents the quantity of berries accessible to the agent, wb denotes the agent's current well-being level with $G = g_1, ..., g_l$ being the set of goals. The variable $g \in G$ is the set of favourable states the agent wishes to achieve, A is the set of possible actions available to the agent and Z represents the set of behaviours the agent has learned from past interactions

The agents will be initialised on an $n \times m$ grid, populated by agents with $B_{initial}$ berries randomly distributed on the grid initially. Each agent $ag_i \in AG$ starts with an initial health value $h_{initial}$, and during the simulation agents perform one of the following actions at each timestep: *foraging* for berries,



(a) Capabilities harvest scenario: agents have diverse abilities, with tall agents only able to forage berries in trees and short agents are restricted to foraging berries found on the ground. Agents are able to share berries with each other by throwing them to each other.

(b) Allotment harvest scenario: Each agent is assigned its own designated grid plot, where berry harvesting is limited to that assigned plot. Each plot yields a distinct amount of berries and agents can also share berries with those in different allotments by throwing berries.

Figure 3.1: **Harvesting environments** (a) Capabilities harvest scenario investigates how agents learn to forage for berries based on the distinct capabilities of the agents. This evaluates how effectively they make ethical decisions and the impact on societal well-being. (b) Allotment harvest scenario looks at how agents learn to collect berries in restricted zones, while testing their effectiveness at ethical decision making and its effect on the well-being of the society.

eating berries, or *throwing* berries to other agents.
The health dynamics of an agent are governed by the health gain h_{gain}, when eating a berry, and the health decay h_{decay} at each timestep. Agents act asynchronously in a random order at each timestep, being able to select from the following actions:

- **Move** (*north, east, south, west*): Agents move towards the nearest berries based on a pathfinding algorithm. If an agent is on a berry it forages it, storing the berry in the agent's berry inventory.
- **Eat**: Agents consume a berry from their inventory, gaining health.
- **Throw**: Agent ag_i can throw a berry to another agent ag_j if ag_i possess berries $b_{ag_i} > 0$ and if their health exceeds threshold $h_{ag_i} > 0.6$.

The simulation employs a DQN reinforcement learning framework, as discussed in subsection 2.1.1. Agents are initially trained in a basic harvest scenario, where berries are universally accessible, with each agent assigned $b_{ag_i} = 0$. Agent behaviour is then tested on two distinct, more complex harvesting scenarios designed, shown in figure 3.1, to evaluate the application and effectiveness of Rawlsian ethical principle and norm emergence:

- **Capabilities Harvest**: In this scenario agents posses heterogenous abilities, allowing them to forage specific types of berries (tall agents can only forage berries from trees and short agents can only forage berries on the ground). This tests the agent's ability to make ethical decisions and cooperate despite capability constrain
- **Allotment Harvest**: Agents are assigned predefined spatial grid allocations, limiting their access to berries in specific subsections. Agents can only forage berries in their allocated allotment, with. This scenario tests the ethical decision-making within spatially constrained resource environments.

3.1.2 Interaction and Learning Mechanism

RAWL-E agents employ an architecture, shown in figure 3.2, with 3 interconnected modules: an interaction module, ethics module, and norms module. It's designed to balance individual objectives with collective societal interests, adaptively respond to changing conditions, and learn from both personal experience and social feedback.

At each step, the agent calculates its state (i.e. observation of the environment), which is a feature vector $x(s_t)$ in state s_t, consisting of its health h, its berry inventory $b \in B$ it has, its expected days to live which is its well-being wb, based on its health and berry reserves, and the closest distance to a berry

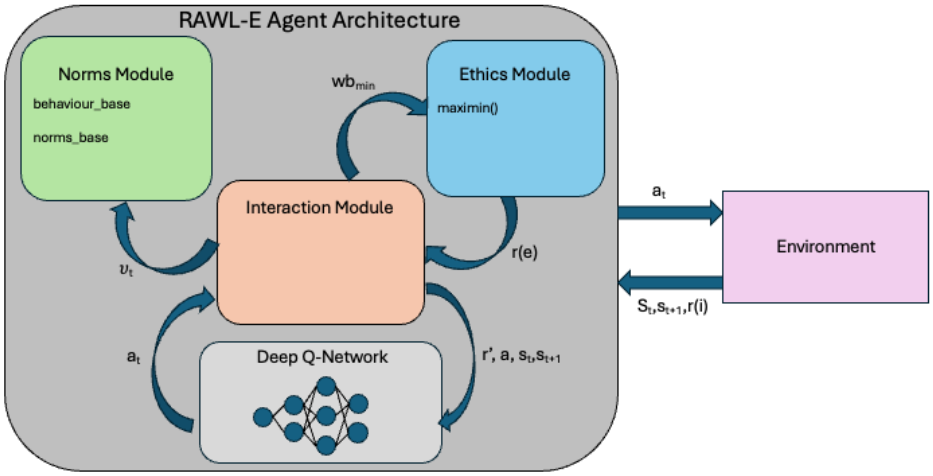


Figure 3.2: **RAWL-E agent architecture diagram** This illustrates the interaction between the norms module, ethics module, Interaction module and the DQN. The agent interacts with its environment, selecting actions and receiving its new states and rewards.

in the environment. Using this observation, it uses an ϵ-greedy policy to explore by choosing an action randomly or to exploit its environment using the policy π, which uses the DQN architecture to predict the estimated future rewards it receives from choosing each action in the state it is in. The action which yields the highest cumulative reward is selected, guiding the agent towards its goal G.

Ethics Module

The ethics module assesses how an agent's actions influence the well-being of others in the system, by using an ethical utility function. The module utilises Rawlsian ethics, aiming to maximise the well-being of the worst off agents in a society, as explained in Section 2.3. The ethics module uses the `_maximin_sanction` function, presented in Figure 3.3, to dynamically calculate ethical sanctions based on changes in the minimum societal well-being. Specifically, it considers the minimum well-being value wb_{min} and the number of agents experiencing the minimum before and after an action is taken.
A base ethical reward of 0.4 is set as `self.reward`. Adjustments to this reward depend on the constants $\alpha_{ethical\_multiplier}$ (for changes in the minimum well-being) and $\beta_{ethical\_multiplier}$ (for changes in the number of agents at minimum well being). The rationale behind this reward mechanism is to incentivise actions that improve the minimum societal welfare. Larger improvements are rewarded proportionally more which strongly encourage more ethical actions. Actions reducing the societal minimum also bring upon proportional sanctions.

Norms Module

The norms module stores and updates norms that the agent learns through interactions with their environment. The module contains a `behaviour_base`, which stores the behaviours of the individual agent, and `norm_base`, which stores the behaviours widely adopted by the society as norms.

Definition 3.1.1 (Behaviour)**.** *The behaviour, defined as $\zeta \in Z$ is structured as the tuple $\langle pre, act \rangle$, where $pre \in Expr$ is the precondition and $act \in Expr$ is the action. $Expr$ is a logical expression that can be determined as true or false.*

In Definition 3.1.1, *pre* is the condition under which the behaviour arises and *act* is the associated action that is triggered when the precondition is satisfied. A behaviour can be encoded explicitly in an if-then format:

$$\langle behaviour \rangle ::= IF \langle pre \rangle THEN \langle act \rangle \tag{3.1}$$

```
    def _maximin_sanction(self, prev_min, no_prevs_mins, soc_wb):
        curr_min = self.maximin_welfare(soc_wb)
        curr_no_prev_mins = np.count_nonzero(soc_wb==prev_min)

        # Differences for partial penalty/reward
        min_diff = curr_min - prev_min
        count_diff = no_prevs_mins - curr_no_prev_mins
        beta = 0.05
        alpha = 0.04

        if curr_min > prev_min:
            return self.reward + (alpha * min_diff)
        elif curr_min < prev_min:
            return -self.reward - (alpha * min_diff)
        elif curr_no_prev_mins < no_prevs_mins and curr_min == prev_min:
            return self.reward + (beta * count_diff)
        elif curr_no_prev_mins > no_prevs_mins and curr_min == prev_min:
            return -self.reward - (beta * count_diff)
        return 0
```

Figure 3.3: **Code for generating dynamic ethical sanction** The function rewards or penalises the agent based on whether the minimum societal welfare, `curr_min`, or number of agents experiencing this minimum, `curr_no_prev_mins`, has improved or declined compared to the previous value before the action was taken.

The module assesses behaviours based on how frequently they occur num and the cumulative reward r' an agent receives when executing them. The behaviour fitness $\tau(\zeta)$ is calculated by:

$$\tau(\zeta) = num \cdot r' \cdot \lambda^{\eta} \tag{3.2}$$

where η represents the behaviours age in timesteps and λ is the decay rate.
Behaviours transition into societal norms when widely adopted throughout the society, where norm emergence is recognized when behaviours reach a predetermined threshold of societal adoption of 90%. Thus, norms are explicitly emergent, procedural, arising from the bottom-up, via vertical transmission. The norms module also employs self-enforcement as its primary enforcement mechanism by using self-directed sanctions to guide agents towards their goals and beneficial societal behaviours. From this we can see that RAWL-E agents are instrumental and cooperative in their norm adoption mechanism.

For each agent interaction, the agent checks if the current behaviour matches any $\langle pre, act \rangle$ in the `behaviour_base`. If it does, then the algorithm updates the behaviours fitness $\tau(\zeta)$ by incorporating new rewards and increasing its usage count. Otherwise, it creates and stores this new behaviour. Every 10 steps, if the behaviour base exceeds its capacity, the least relevant behaviours are removed based on their fitness.

Interaction Module

The interaction module uses a DQN architecture, receiving the action that is predicted to produce the highest reward or a random action by the ϵ-greedy policy. The module then obtains its new observation s_{t+1} and individual reward $r(i)$ from the environment, as shown by the table in figure 3.1, by observing the action performed and how this action affected the agent. The minimum well-being of the society is obtained, which is then passed into the ethics module to produces an ethical reward $r(e)$ to encourage agents to learn behaviours to promote ethical norms. The interaction module then produces a combined reward score r' with the individual $r(i)$ and ethical reward $r(e)$ so that $r' = r(i) + r(e)$. Before the agent has performed an action the interaction module uses part of its observation to obtain the antecedent of an action from the norms module, this is then used after the action is performed to update the behaviour base in the norms module.

DQN

The DQN architecture of each agent is implemented using the `Tensorflow` library to train and predict Q-values from the agents' experiences. It is the core decision making mechanism for the RAWL-E agents, building on principles initially introduced by Google Deepmind in 2013 as discussed in Section 2.1.1. This

Action or Outcome	Reward Value
Death	−1
No berries	−0.1
No benefactor	−0.1
Insufficient health	−0.1
Neutral reward	0
Throw	0.5
Forage	1
Eat	1
Survive	1

Table 3.1: **Individual reward values for agent actions/outcomes.** The column on the left represents the actions performed by an agent or the outcomes of actions felt by the agent. The right column represents the associated positive or negative individual reward received.

implementation uses neural networks to approximate the optimal Q-values (state-action values), within the multi-agent harvesting scenarios.

The fully connected Neural Network Architecture consists of:

- An input layer corresponding to the observed state features.
- Two hidden layer with 128 hidden units, using a ReLU activation function, as shown in Table 4.2.
- An output layer which generates 4 Q-values corresponding to the expected reward received carrying out each of the 4 actions in the given state.

The *experience replay buffer* is used to stabilise the learning process, which stores the tuples $\langle s_t, a_t, r_t, s_{t+1}, done \rangle$. During training, random batches of 64 experiences are sampled from the buffer to learn from a diverse range of experiences, thus mitigating correlations between the states of consecutive training samples.
Following the methodology outlined in Section 2.1.1, a separate *target network* is used to compute stable targets for the Q-values. The target network $\hat{Q}(S, A; \theta^-)$ maintains fixed parameters, copying the main Q-network's $Q(S, A; \theta)$ parameters every $C = 50$ learning timesteps.
A *Huber loss function* is also utilised since it's less sensitive to large errors which would lead to outliers in the predicted Q-values. It combines the properties from the mean squared error and mean absolute error. The overall training loop for the DQN is summarised as follows:

1. Initialise the replay buffer, Q-network parameters θ and the target network parameters θ^-.
2. For each training episode, agents interact with the environment and store experiences.
3. Random sets of batches are sampled from the replay buffer.
4. The Q-network's parameters θ are updated through back-propagation from the Huber losses from the Q-values.
5. The target networks parameters θ^- are updated periodically from the main Q-network's weights.

3.2 MORL RAWL-E Implementation

This implementation adapts the RAWL-E agents to use multi-objective reinforcement learning (MORL) to learn a single policy π. MORL has traditionally been used in tabular Q-learning methods in simple environments, however due to scalability and memory limitations with tabular Q-learning in large state spaces there is a need for this to be implemented with DQN and more complex environments.
The MODQN framework being implemented for RAWL-E agents is adapted from Nguyen et al. [49]. Single policy RL involves computing an optimal policy for the combination of the multiple objectives simultaneously. While the implementation described considers image based state representations using convolution neural networks, my implementation adapts this framework for simpler state inputs that take in tuples of numerical data, focusing explicitly on balancing ethical and individual objectives.

$$L(\theta) = \sum_{i=1}^{2} L_i(\theta) \tag{3.3}$$

Each $L_i(\theta)$ is the loss for each objective, ethical and individual, of the DQN given by:

$$L_i(\theta) = E[\mathcal{H}_\delta(\gamma \max_{a'}(y - Q_i(s, a; \theta))^2)] \tag{3.4}$$

In Equation 3.4, $\mathcal{H}_\delta$ represents the Huber loss equation defined as:

$$\mathcal{H}_\delta(x) = \begin{cases} \frac{1}{2}(x)^2, & \text{if } |x| \leq \delta, \\ \delta(|x| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \tag{3.5}$$

Figure 3.4 shows the neural network configuration for the MODQN, which includes the input layer with 4 nodes and 2 hidden layers with 128 nodes as was configured in the Base RAWL-E implementation in Section 3.1, however the final layer includes double the nodes compared to the Base RAWL-E implementation, since one half outputs the estimated Q-values for the individual objective and the other half outputs it for the ethical objective, resulting in a prediction separated into two distinct dimensions. Each prediction resulted in a two-dimensional Q-value, $Q(s, a; \theta) = [Q_{individual}, Q_{Ethical}]$.
The training function has been modified to manage the two-dimensional output for the ethical and individual objectives, requiring the adaptation of the loss function and action selection criteria. The Q-values from both dimensions are combined into a single scalar score through weighted linear scalarisation, to select the actions the agent wants to perform via the equation:

$$a = \underset{a}{\text{argmax}}[w_{individual} \cdot Q_{individual} + w_{ethical} \cdot Q_{ethical}] \tag{3.6}$$

The weights $w_{individual}$ and $w_{ethical}$ determine the relative priority given to each objective and they are set as $w_{individual}, w_{ethical} = 1$ to test this model.

3.3 MORL RAWL-E Implementation with Ethical Embedding

The problem with the base RAWL-E agent model is that ethical behaviours are not guaranteed. If the individual reward overshadows the ethical reward for a particular action, the ethical incentive can be ignored, leading the agent to behave in a way that prioritises self-interest over fairness. Following Rodriguez-Soto et al. [2], I built upon the model in section 3.2 by using an adapted partial convex hull computation algorithm and an ethical embedding function to find the linear weighting $w_{ethical}$ of the ethical rewards such that learning of ethical policies is guaranteed.

A policy can be described as ethical if and only if:

$$V_1^{\pi^*} = \underset{\pi \in \Pi}{max} V_1^{\pi} \tag{3.7}$$

where Π is the set of all policies and V_1^π is the accumulated ethical reward under policy π. Equation 3.7 essentially states that the policy must maximise the ethical objective. The ethical optimal policy is a decision making guideline for normative agents which prioritise ethical objectives over purely individual objectives. Formally this is defined as:

Definition 3.3.1 (Ethical-optimal policy)**.** *Given an MOMDP $\mathcal{M}$, a policy π^* is an ethical-optimal policy if and only if*

$$V_0^{\pi^*} = \underset{\pi \in \Pi_e}{max} V_0^{\pi} \tag{3.8}$$

where Π_e is the set of ethical policies and V_0^π is the accumulated individual reward under policy π.

This means that ethical optimal policy must maximise the ethical objective Q-value dimension. Once the ethical objective has been optimised, the policy selects the remaining policy/policies that maximises the individual objective Q-value dimension. This should help to ensure the agent strives to achieve its individual goals optimally while also operating within its ethical limits.

The ethical embedding problem can be formally defined as:

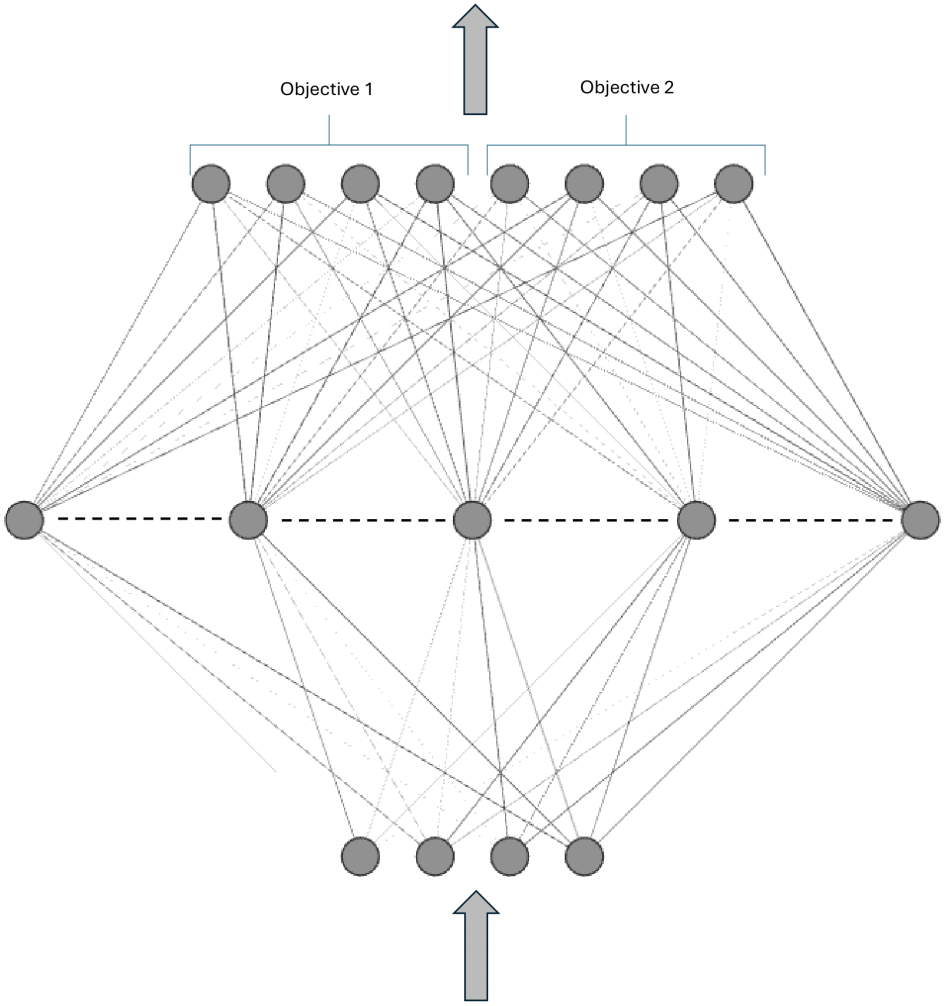


Figure 3.4: **Neural network architecture for the DQN of the MORL RAWL-E agents**. The input layer contains 4 nodes, corresponding to the 4 parts of the agent's observation. The output layer contains 8 nodes corresponding to each dimension of the Q-values, which are $Q_{individual}$ and $Q_{ethical}$. There is two hidden layers, each with 128 nodes and the network is fully connected, utilising a ReLU activation function.

Definition 3.3.2 (Ethical embedding problem)**.** *Given $\mathcal{M}$ is an ethical MOMDP, the ethical embedding problem requires a weight vector w such that all optimal policies in the MDP $\mathcal{M}' = \langle S, A, T, w_0 R_0 + w_1 R_1, T \rangle$ are ethical-optimal in $\mathcal{M}$*

In other words, for any weight vector $w_{ethical} > 0$, all policies that are optimal under $R_0 + w_1 R_1$ are also ethical-optimal as stated in Definition 3.3.1. I make use of a scalarisation function f, to apply the embedding once this weight has been calculated.
Our aim is to find an ethical embedding function to guarantee an agent learns ethical-optimal policies. The embedding function combines multiple reward objectives into a single scalar value for the agents action selection mechanism. This can be done with a linear scalarisation function f expressed as:

$$f(\vec{V}^{\pi}) = \vec{w} \cdot \vec{V}^{\pi} = w_0 V_0^{\pi} + w_1 V_1^{\pi} \tag{3.9}$$

where $\vec{w} = (w_0, w_1)$ is the weight vector determining how much each objective influences decision making, with each weight $w_0, w_1 > 0$ to guarantee the agent considers both types of rewards. In our implementation the individual objective is fixed at $w_0 = 1$, so the weight vector w_1 is referred to as w for simplicity. To guarantee that the ethical embedding problem always has a solution, an ethical policy must exist in the MOMDP (i.e it must be possible for the agent to behave ethically). For example, take a look at the convex hull in Figure 3.5. The green passing through the best ethical policy point represents the weight $w_1 = 0.5$. By $R_0 + w \cdot R_1 = score$, this gives us a score of 4.5. Any point above the line is a policy with a higher score, such as $C1$, so the best ethical point does not dominate policy $C1$ with the weight $w_1 = 0.5$. Increasing w tilts the decision criteria to favour ethics more, (the higher the weight, the larger the individual objective must be to dominate the best ethical policy for a point with a low ethical objective). We can see that the line representing the weight $w = 2.0$ makes the best ethical policy dominate all other points, so it solves the ethical embedding problem, since this weight means that any policy that is optimal under the weighting $w = 2$ will be ethically optimal.

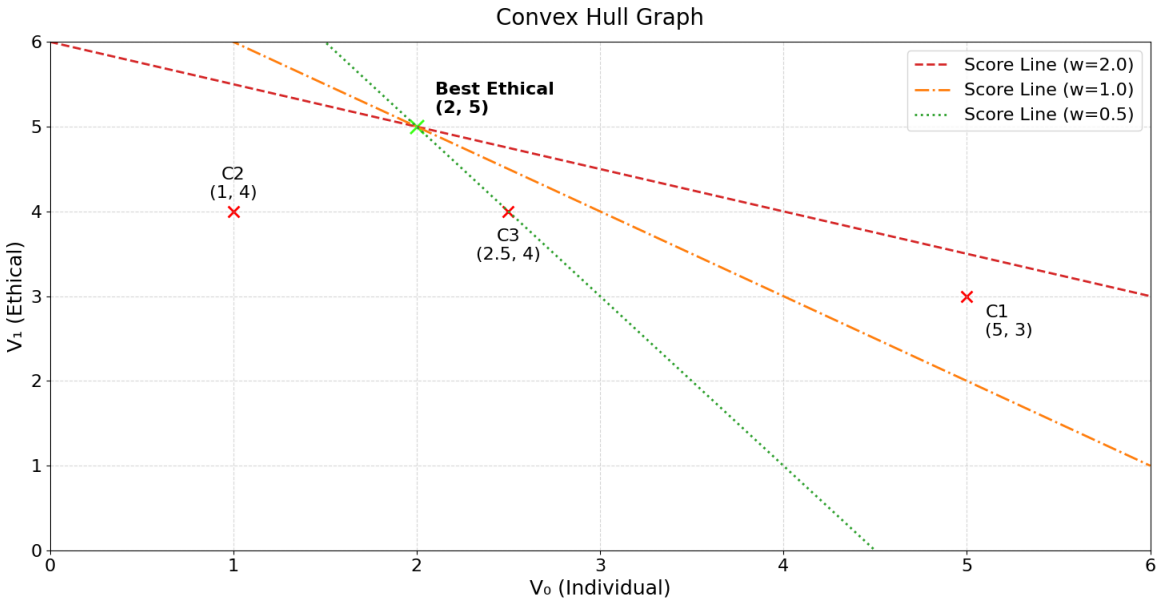


Figure 3.5: **Example objective space of the convex hull of $\mathcal{M}$.** Composed of the ethical objective on the y-axis and the individual objective on the x-axis. Each point represents the multi-objective value of a given policy, where each line represents the linearly scalarised scores passing through the best ethical point.

If at least one ethical policy exists, this guarantees that there exists a weight w, such that any unethical policy can always be made suboptimal, ensuring that ethical behaviour is encouraged.

In our implementation, the steps required to extract the weights that solve the ethical embedding problem is as follows, compute the partial hull, extract the ethical-optimal policies from the partial convex hull, and finally compute the embedding function solution.

To compute the ethical embedding problem, we calculate the partial convex hull $P \subseteq CH$, which is the minimal convex hull, as defined in 2.1.3, needed to solve the ethical embedding problem. Since we only care about the optimal ethical policies, we do not consider the optimal policies for the individual objective, only the ethical objective. Hence, we only have one weight vector w for the ethical objective scalarisation. In order to compute the partial convex hull, we create a function, as shown by Algorithm 3, which updates the dictionary V that maps each state to a set of 2D points, which is the potential outcome for that state, given by policy π. The algorithm retrieves the current state s from a random sample in the replay buffer, which the Q-network uses to predict the Q-values for that state. Each action gives a predicted 2D reward for and this is combined with the discounted future rewards, which is the hull of the expected next state. The `compute_next_state()` function computes an expected next state using the current state and a given action. The partial convex hull is created by combining the set of points and represents the best trade-offs for the ethical objective at state s. The reason why it is sample based is that rather than processing every single state [2] in the large state space, a batch of experiences are sampled from the replay buffer, which reduces the computational load.

Algorithm 3 Sample-based partial convex hull iteration

Initialise mapping V (state $\rightarrow$ set of 2D value vectors), replay buffer $\mathcal{D}$, and DQN Q
for $k = 1$ *to* K **do**
  Sample a batch $\mathcal{I}$ of B indices from $\mathcal{D}$
  for *each index* $i \in \mathcal{I}$ **do**
    Let s, s_{next}, *done* be from $\mathcal{D}[i]$
    $Q\_vals \leftarrow Q$.predict(s); $s\_key \leftarrow$ tuple(s)
    $old\_hull \leftarrow V[s\_key]$ if defined, else $\emptyset$
    $accum \leftarrow \emptyset$
    for *each action* a *with reward* r *in* Q_vals **do**
      $next \leftarrow$ compute_next_state(s, a, r)
      $future \leftarrow \begin{cases} V(\text{tuple}(next)), & \text{if not } done \text{ and exists} \\ \emptyset, & \text{otherwise} \end{cases}$
      $accum \leftarrow accum \cup$ translate_hull($r, \gamma, future$)
    end
    $V[s\_key] \leftarrow$ get_hull($old\_hull \cup accum$)
  end
end
return V

Once we have the partial convex hull for a subset of the states, we can extract the ethical policies and solve the ethical embedding problem, as shown by Algorithm 4. The algorithm computes an ethical weight w, which ensures the most ethical policies, strictly dominates all the other policies for a given partial hull for a state, using the subroutine GetEthicalWeight. GetEthicalWeight works by taking a set of hull points, 2D value vectors (V_0, V_1), each representing different policies that choose actions and receive rewards in a particular state. To identify the most ethical policy, it sorts the set of points $V[s]$ by the ethical dimension V_1 in ascending order. If there are several points with the same maximum R_1, then the point with the highest R_0 is chosen.

Algorithm 4 Compute global ethical weight

Input: Dictionary V mapping each state to a set of 2D value vectors (each row is (R_0, R_1)); small constant $\epsilon > 0$
Output: Global ethical weight w

Initialize $best \leftarrow 0.0$
for *each state* s *in* V **do**
  $hull\_points \leftarrow V[s]$
  $local\_w \leftarrow$ GetEthicalWeight($hull\_points$)
  if $local\_w > best$ **then**
    $best \leftarrow local\_w$
  end
end
return $w \leftarrow best + \epsilon$

3.4 Metrics

To evaluate the effectiveness of the emerged norms N, we evaluate their impact on the individual agents and the societal outcomes. To assess the effectiveness of these norms, we define a set of variables and metrics computed for each simulation run to test our hypothesis.

3.4.1 Variables

- **well-being (*wb*)**: The well-being is calculated to be the expected number of days the agent has to live, represented by a function of the number of berries the agent has in its inventory and its current health as shown by Equation 3.10.

$$ag_{wb} = \frac{ag_{health} + (ag_b \times h_{gain})}{h_{decay}} \tag{3.10}$$

- **berries consumed ($b_{consumed}$)** The number of berries consumed by an agent.

3.4.2 Metrics

The fairness of a society is measured by computing the inequality and minimum experience. The sustainability of the society is measured by evaluating its social welfare and duration to assess how well the society copes as a whole.

- **Inequality** The Gini index is used to measure inequality in a society. This measure can be applied to many variables within the context of MAS due to it being easy to interpret, since a Gini index of 0 indicates perfect equality and a Gini index of 1 indicates perfect inequality [50]. This measures egalitarianism, which aims to treat all agents equally [46]. The lower this value, the better.
- **Minimum Experience** This represents the lowest experience an agent experiences across a society. This is one of the metrics for examining fairness through the Rawlsian ethics principle, since the ethical goal is to maximise the minimum welfare in the society [3]. The higher this value, the better.
- **Social Welfare** This is a measure of how much society gains as a whole. This follows the principle of utilitarianism [51]. The higher this value, the better.
- **Duration** This represents the total length of the episode and indicates the society's capability to withstand pressures and maintain stability over time. The higher this value, the better.

3.4.3 Hypothesis

The following hypotheses will be evaluated. The null-hypotheses for hypotheses 2, 4, 6 and 8 indicates that one group is greater than the other and vice versa. The null-hypothesis for hypotheses 1, 3, 5 and 7, indicates no difference.

1. Inequality: Norms emerging from the MORL RAWL-E model lead to the same inequality as the baseline RAWL-E model.
2. Inequality: Norms emerging from the ethical embedding RAWL-E model lead to a lower inequality than the norms from the MORL RAWL-E model.
3. Minimum Experience: Norms emerging from the MORL RAWL-E model lead to the same minimum experience as the baseline RAWL-E model.
4. Minimum Experience: Norms emerging from the Ethical Embed RAWL-E model lead to a higher minimum experience than the norms from the MORL RAWL-E model.
5. Social Welfare: Norms emerging from the MORL RAWL-E model lead to the same social welfare as the baseline RAWL-E model.
6. Social Welfare: Norms emerging from the Ethical Embed RAWL-E model lead to higher social welfare experience than the norms from the MORL RAWL-E model.
7. Duration: Norms emerging from the MORL RAWL-E model lead to the same duration as the baseline RAWL-E model.

8. Duration: Norms emerging from the Ethical Embed RAWL-E model lead to a longer duration than the norms from the MORL RAWL-E model.

To test hypotheses $2, 4, 6$ and 8, we conduct the Mann-Whitney U test, a non-parametric method to determine whether there is a significant difference between the distributions of two small independent samples. A p-value less than the significance level of 0.01 indicates that the difference between the groups is statistically significant. To test hypotheses $1, 3, 5$ and 7 we use equivalence testing, since my goal is to test whether the difference between the two groups is sufficiently small to be considered irrelevant. We use an equivalence margin of 10% of the combined mean total between the two groups since it adjusts naturally with the magnitude of the combined means, making my equivalence criteria dynamically adaptive to the context-specific baseline performance. We perform the two one sided test procedure at a significance level of 0.01. Cohen's d measures the magnitude of difference between the two groups means. Along with the Mann-Whitney U test this will not only determine whether the difference between the two groups is statistically significant, but also determine how meaningful this difference is. Cohen's d categorises effect sizes into negligible (< 0.2), small ($0.2 - 0.5$), medium ($0.5 - 0.8$) or large (> 0.8). The tests will be performed on data from the allotment harvest of the simulation.

Chapter 4

Results

In this chapter, we present our quantitative findings from our experiments, showing how each model performs in terms of inequality, minimum experience, social welfare, and robustness in both capabilities and allotment harvest scenarios.

4.1 Experimental Setup

4.1.1 Computing infrastructure

The series of simulations were performed on an Isambard 3 MACS supercomputer node on the ampere partition. The processor used was an AMD EPYC 7543P (32C 3.7GHZ), with 50GB of RAM and a Nvidia A100 SXM4 40GB GPU.

4.1.2 Hyperparameter Selection

The series of parameters for the simulation, in Table 4.1, and interaction module, in Table 4.2, have been selected with reference to Woodgate et. al [1].

Description	**Parameter**	**Final Value**
Capabilities grid size	$n_{\text{capabilities}} \times m_{\text{capabilities}}$	8×4
Allotment grid size	$n_{\text{allotment}} \times m_{\text{allotment}}$	16×4
Number of agents	k	4
Initial number of berries	B_{initial}	12
Initial health of agent	h_{initial}	5.0
Health gain from eating berry	h_{gain}	0.1
Health decay	h_{decay}	-0.01
Minimum health to throw	h_{throw}	0.6
Number of episodes	e	2000
Maximum steps in episode	t_{max}	50
Ethical reward minimum wellbeing value multiplier constant	$\alpha_{ethical\_multiplier}$	0.04
Ethical reward number of minimum wellbeing multiplier constant	$\beta_{ethical\_multiplier}$	0.05

Table 4.1: **Simulation experiments parameters.** The first column lists each experimental setting, the second gives its symbolic parameter and the third reports the final value chosen to be controlled in my experiments.

Most of the parameter values are retained from Woodgate et al, since experimenting with alternative values would give impractically long training times or negligible changes in the test results. For the MORL approach, since the number of predicted values doubled, I decided experimented with the neural network parameters, testing Hl values of 2 and 3, and Hn values of $128, 256$, in Table 4.2. I ultimately stuck with the configuration with 2 hidden layers and 128 hidden units, since increasing the size of the neural network led to significantly longer training times without notable improvements in the test performance.

Description	Parameter	Final Value
Batch size	S	64
Iteration for updating weights of target network	C	50
Probability of exploration	ϵ	0.0
Learning rate	α_{lr}	0.0001
Number of neural network hidden units	Hn	128
Number of neural network hidden layers	Hl	2

Table 4.2: **Interaction module parameters.** The first column describes each hyperparasite of the interaction module, the second shows the corresponding notations and the third provides the final value chosen to be controlled in my experiments.

The parameters $\alpha_{ethical\_multiplier}$ and $\beta_{ethical\_multiplier}$ were chosen by me based on the test performance of the RAWL-E society baseline. The tested values of $\alpha_{ethical\_multiplier}$ and $\beta_{ethical\_multiplier}$ included 0.03, 0.04 and 0.05.

4.2 Simulation Results

We introduced the multi-objective extension of the RAWL-E model (MORL RAWL-E) so that two distinct reward objectives are produced by the neural network rather than relying on the post hoc reward/sanctioning mechanism of the baseline RAWL-E model. Every decision has two dimensions, an individual reward and an ethical reward. I hypothesise that this model performs on par with the base RAWL-E implementation.

To ensure that more ethical behaviours consistently guide agent behaviour, an ethical embedding enhancement was integrated on top of the MORL RAWL-E framework. This is done by calculating a specific weight for the ethical reward component to ensure that the agent's learning process prioritises policies that are ethically optimal, as per Definition 3.3.1. These ethical priorities are embedded into the agents action selection mechanism to address the potential of an agent's self-interest considerations overriding the fairness considerations present in RAWL-E.

For testing we run each simulation 2000 times with each simulation running until all the agents have died or a maximum of 50 timesteps have been reached.

A summary of the results from the 3 simulation models are provided in Tables 4.3 and 4.4.

4.2.1 Inequality

Table 4.3 shows that the mean Gini index for ag_{wb} falls from 0.17 to 0.12 for the capabilities harvest from the base RAWL-E to the MORL RAWL-E model, yielding a Cohen's d of 1.10. This shows that there is a large effect size so about 86% of MORL episodes have a more equitable distribution of agent well-being compared to baseline RAWL-E episodes. The mean Gini index for $ag_{b_{consumed}}$ falls from 0.10 to 0.08, with a small effect size of 0.24. This suggests that about 60% of MORL episodes have a more equitable distribution of the berries eaten by the agents than the baseline RAWL-E episodes
For the allotment harvest simulation, similar results were obtained; the mean Gini index for ag_{wb} falls from 0.22 to 0.17 between the same models, with a Cohen's d of 1.00. This means that there is a large effect size with a more equitable well-being distribution in approximately 84% of MORL episodes. The mean Gini index for $ag_{b_{consumed}}$ also falls, from 0.15 to 0.11, with a small Cohen's d value of 0.45, indicating that there is a more equitable berries eaten distribution in 67% of MORL RAWL-E episodes. From hypothesis 1, our null-hypothesis 1 represents the inequality difference being positive, so the MORL RAWL-E model having a greater inequality. Null-hypothesis 2 represents the difference being negative, so the base RAWL-E model having a greater inequality. For null-hypothesis 2, $p > 0.01$ for both ag_wb and $ag_{b_{consumed}}$, so we fail to reject null-hypothesis 2. Therefore, we cannot conclude that the norms emerging from the MORL RAWL-E and base RAWL-E models lead to the same inequality. This disagrees with our original hypothesis 1, leading us to conclude that the MORL RAWL-E model leads to a lower inequality compared to the base RAWL-E model.

When comparing the Ethical Embed RAWL-E and MORL RAWL-E models, Table 4.3 shows the mean Gini index for ag_{wb} for the capabilities harvest scenario. The mean Gini index changes from 0.12 in

the MORL RAWL-E model to 0.11 in the Ethical Embed model. This corresponds to a small effect size of 0.46, suggesting that about 68% of Ethical Embed RAWL-E episodes have a more equitable well-being distribution compared to the MORL RAWL-E model. The mean Gini index for $ag_{b_{consumed}}$ increases from 0.08 to 0.10, with a small effect size of -0.16. This suggests that about 56% of the MORL RAWL-E episodes give a more equitable berries eaten distribution than the Ethical Embed RAWL-E model episodes.
For the allotment harvest simulation, the mean Gini index for ag_{wb} changes from 0.17 to 0.18 with an effect size of -0.32, which is small, indicating that 62% of MORL RAWL-E episodes have a more equitable well-being distribution compared to the Ethical Embed RAWL-E model. The mean Gini index for $ag_{b_{consumed}}$ increases, from 0.11 to 0.13, with a small effect size value of -0.24. We can conclude from this that about 60% of the MORL RAWL-E episodes yield a more equitable well-being distribution than the Ethical Embed RAWL-E model episodes. A more detailed visualisation of the results is displayed in Figures 4.1, 4.2, 4.3 and 4.4.
From hypothesis 2, we find that $p > 0.01$ for ag_{wb} and $ag_{b_{consumed}}$, so we cannot reject the null-hypothesis corresponding to hypothesis 2, leading us to conclude that conclude that the Ethical Embed model does not lead to a lower inequality compared to the MORL RAWL-E model.

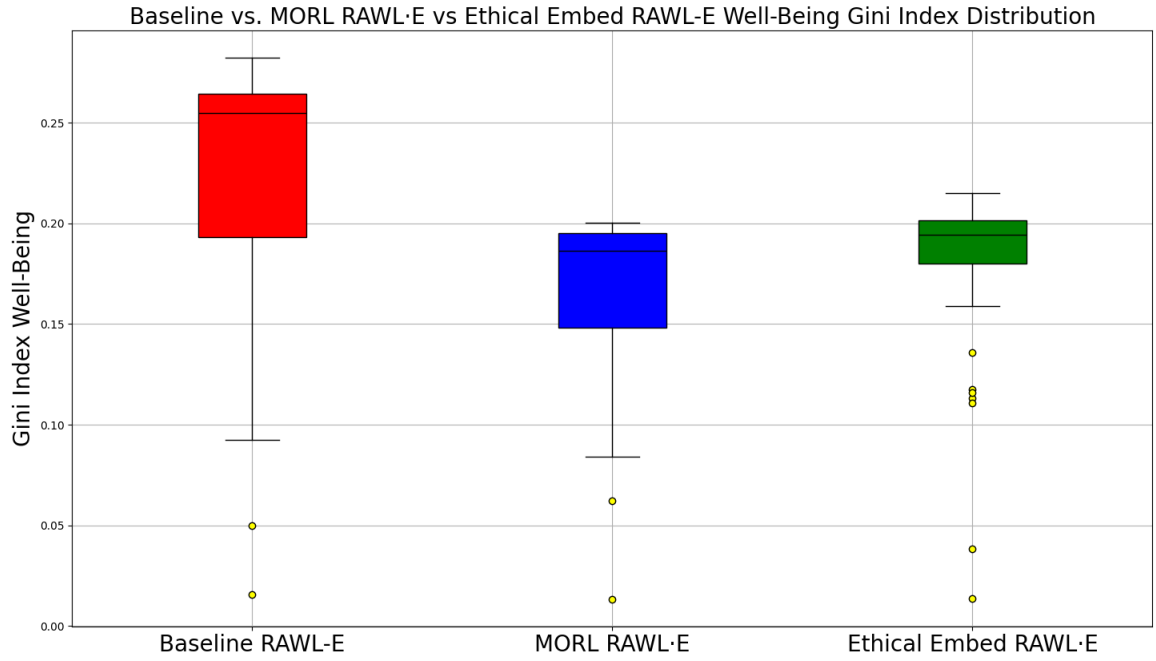


Figure 4.1: **Box plot of the distribution of Gini index of the well-being in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

4.2.2 Minimum Experience

We can see that in Table 4.3, that the mean minimum experience increases for ag_{wb}, from 7.36 to 7.81, from the base RAWL-E to the MORL RAWL-E model. A small effect size of 0.48 suggests that 67% of episodes in the MORL RAWL-E model have a higher minimum well-being experience compared to the base RAWL-E model. The mean minimum experience for $ag_{b_{consumed}}$ also increases from 3.88 to 4.02, with a Cohen's d of 0.05, indicating that the effect size is negligible.
The allotment harvest simulation yields similar results between the two models. We notice that the mean minimum experience for ag_{wb} increases from 7.37 to 10.84, while having a very large effect size of 2.45. This indicates that 99% of the MORL RAWL-E episodes have a higher minimum well-being experience than the base RAWL-E episodes. The mean minimum experience also increases for $ag_{b_{consumed}}$, from 3.80 to 4.45. A small effect size of 0.25 indicates that 60% of MORL RAWL-E episodes yield a higher minimum berries consumed experience than the base RAWL-E model.
From hypothesis 3, our null-hypothesis 1 represents the minimum experience difference being positive, so the MORL RAWL-E model having a greater minimum experience. Null-hypothesis 2 represents the difference being negative, so the base RAWL-E model having a greater minimum experience. We find

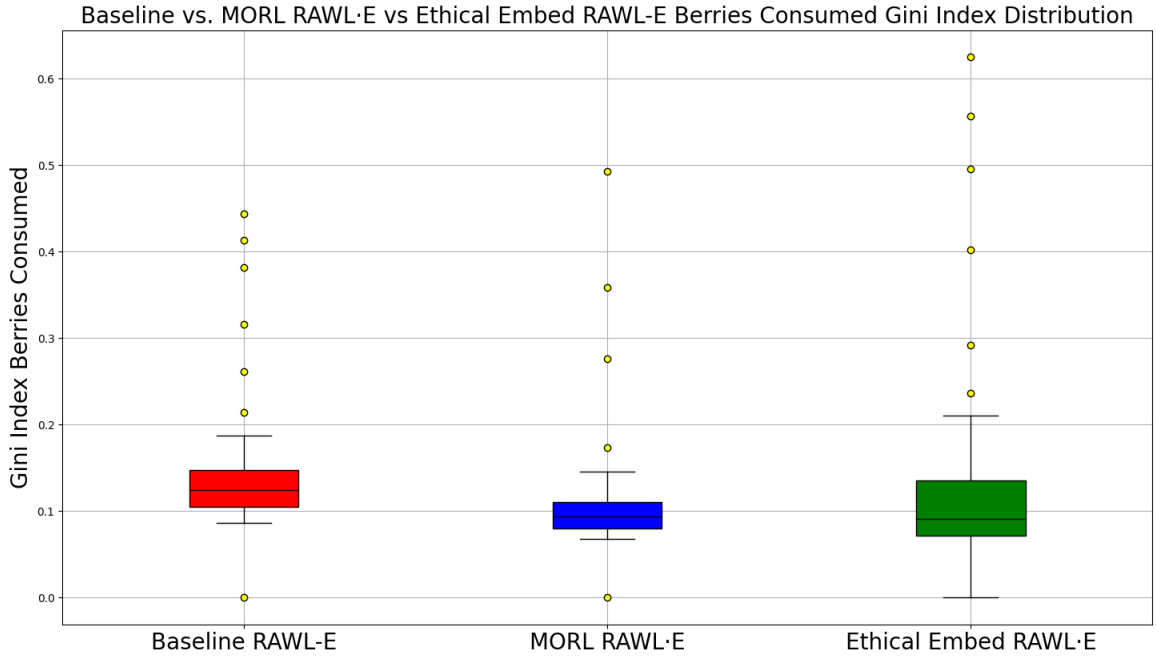


Figure 4.2: **Box plot of the distribution of Gini index of the berries eaten in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

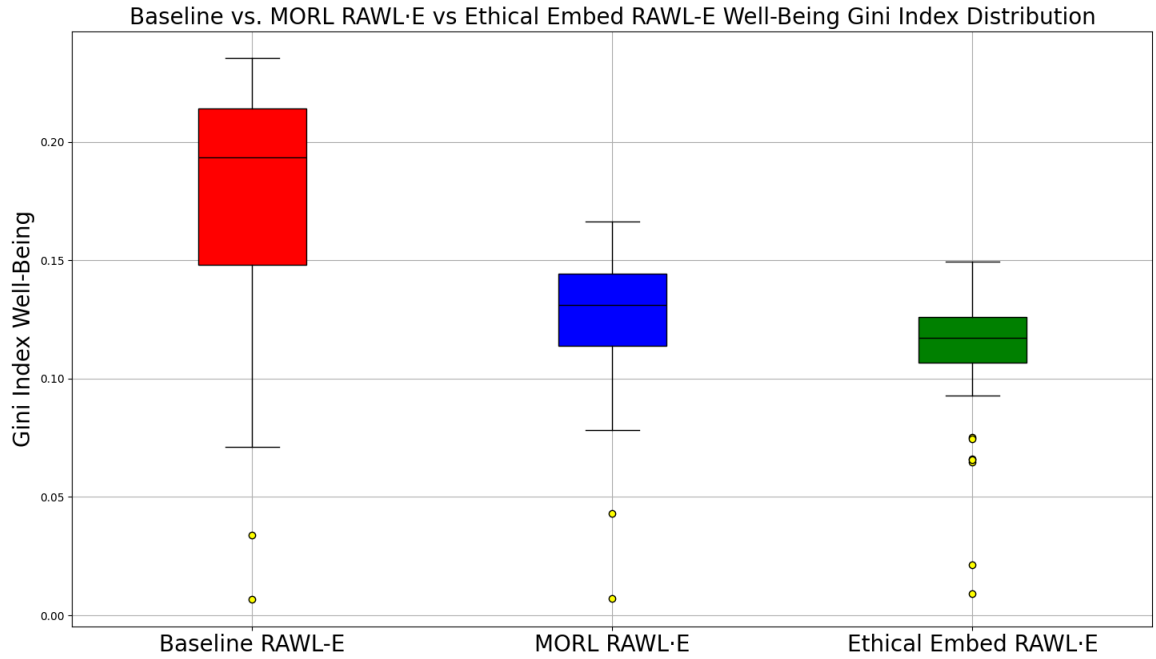


Figure 4.3: **Box plot of the distribution of Gini index of the well-being in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

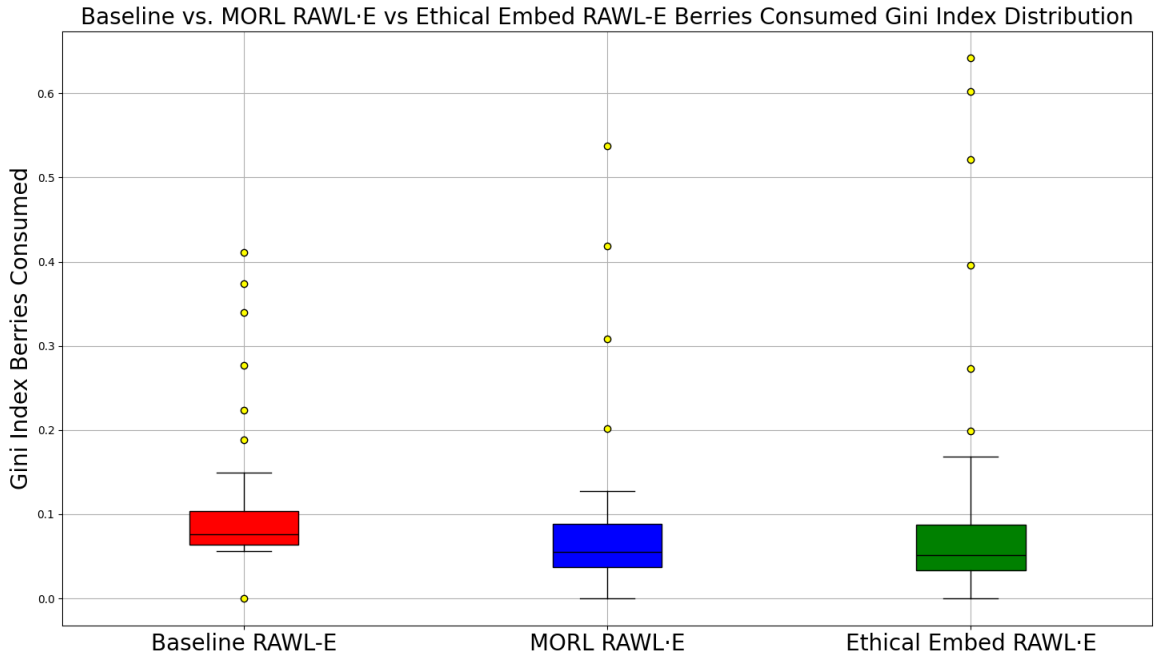


Figure 4.4: **Box plot of the distribution of Gini index of the berries eaten in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

that $p > 0.01$ for null hypothesis 1 with ag_wb and $ag_{b_{consumed}}$, leading us to fail to reject it. As a result, we cannot conclude that norms emerging from the MORL RAWL-E model lead to the same minimum experience as the norms from the base RAWL-E model. This does not support our original hypothesis of equivalence in the minimum experience between the MORL RAWL-E and RAWL-E models, leading us to conclude that the MORL RAWL-E model leads to a higher minimum experience compared to the base RAWL-E model.

We can see that in Table 4.3 that the mean minimum experience for ag_{wb} in the capabilities harvest scenario changes from 7.81 in the MORL RAWL-E model to 6.56 in the Ethical Embed model. The large effect size of -1.52 represents the fact that about 94% of episodes in the MORL RAWL-E model show a higher minimum well-being experience compared to the Ethical Embed RAWL-E model. The mean minimum experience for $ag_{b_{consumed}}$ decreases from 4.02 to 3.93. This corresponds to a negligible effect size of 0.03.
In the allotment harvest simulation, the mean minimum experience changes from 10.84 to 8.37. This difference yields an effect size of -1.61, which is large, suggesting that about 94% of the MORL RAWL-E episodes yield a higher minimum well-being experience than the Ethical Embed RAWL-E model. The mean minimum experience for $ag_{b_{consumed}}$ also decreases from 4.45 to 4.05. This leads to a small effect size of -0.15, indicating that 56% of MORL RAWL-E episodes yield a higher minimum berries consumed experience than the Ethical Embed RAWL-E model.
A more detailed visualisation of the results is displayed in Figures 4.5, 4.6, 4.7 and 4.8.
From hypothesis 4, we find that $p > 0.01$ for ag_{wb} and $ag_{b_{consumed}}$, so we cannot reject the null-hypothesis corresponding to hypothesis 4, leading us to conclude that that the Ethical Embed model does not lead to a higher minimum experience compared to the MORL RAWL-E model.

Between the baseline RAWL-E model and the MORL RAWL-E model, we can see that the MORL RAWL-E model leads to fairer societies compared to the baseline RAWL-E model, which contrasted our expectation of both models performing on par with each other. This implies that the MORL RAWL-E models are more successfully fulfilling the RAWL-E ethical objective by improving the minimum experience of agents better than the baseline RAWL-E model. A possible reason for this includes the fact that the baseline RAWL-E model predicts a grouped Q-value containing ethical objective and individual objective together. This means that during the training of the baseline RAWL-E model, when the ethical reward may be too small or noisy compared to the individual reward it leads to the learned policy prioritising the individual objective, while not accurately learning the ethical objectives. Also, the ethical

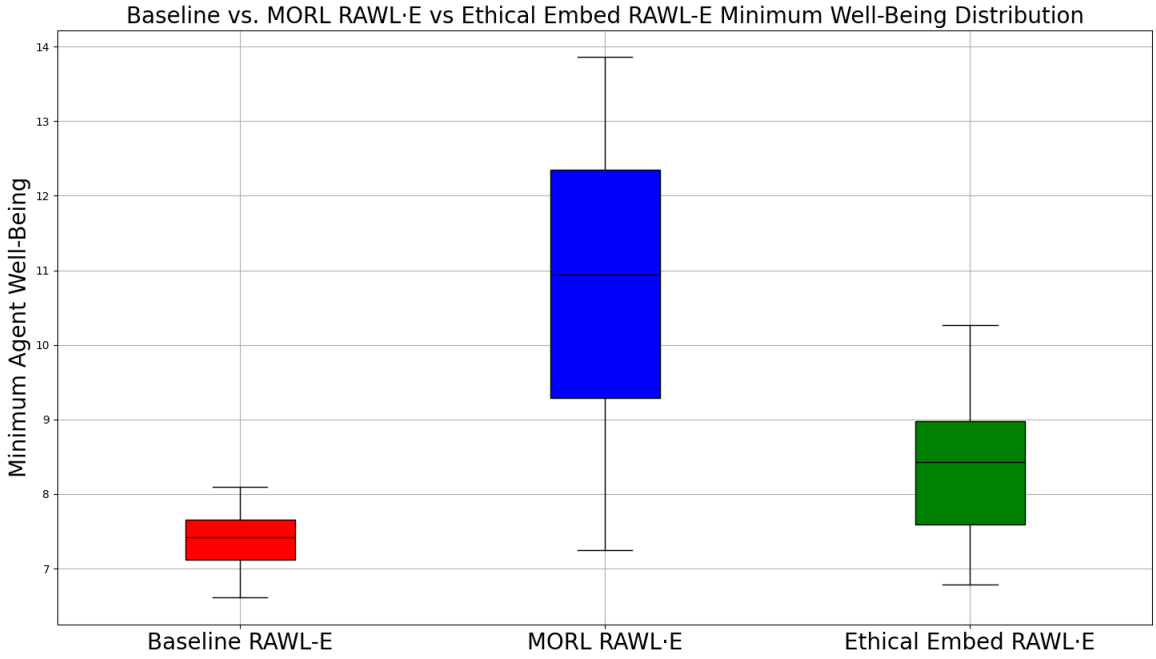


Figure 4.5: **Box plot of the distribution of minimum experience of the agent well-being in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

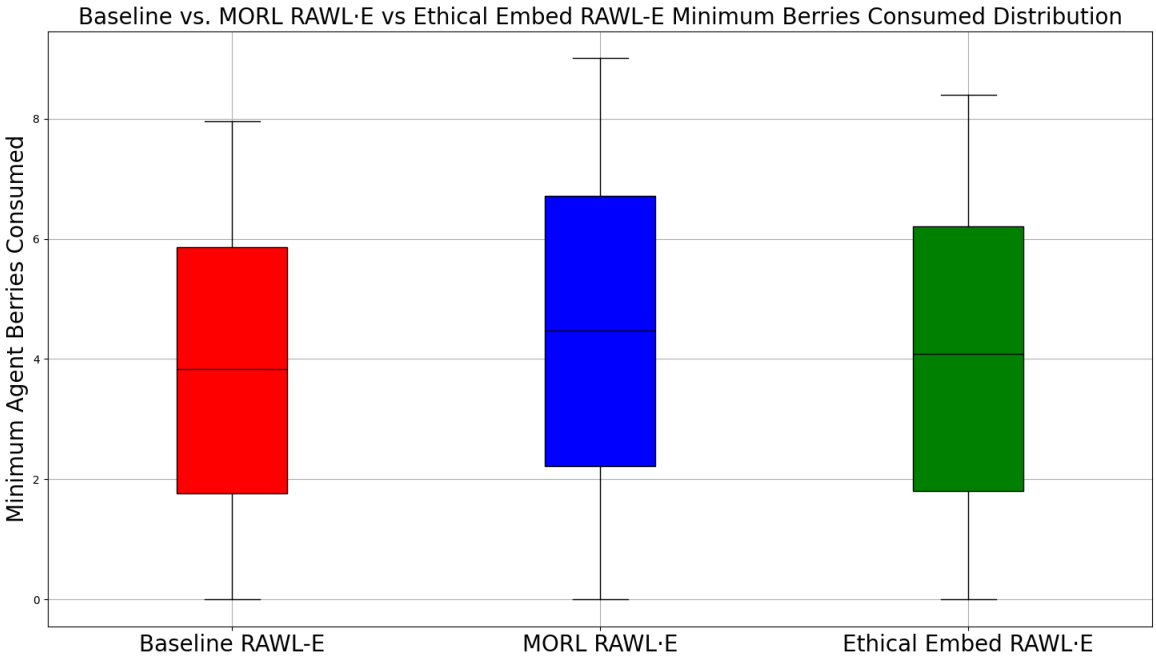


Figure 4.6: **Box plot of the distribution of minimum experience of the berries eaten by agents in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

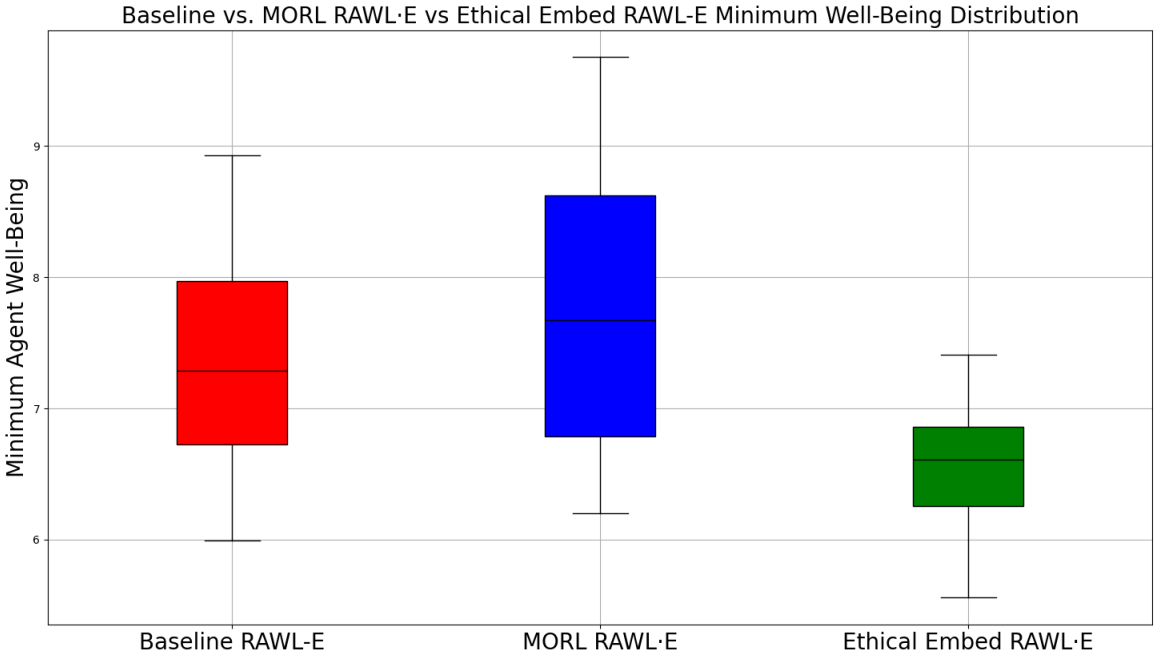


Figure 4.7: **Box plot of the distribution of minimum experience of the agent well-being in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

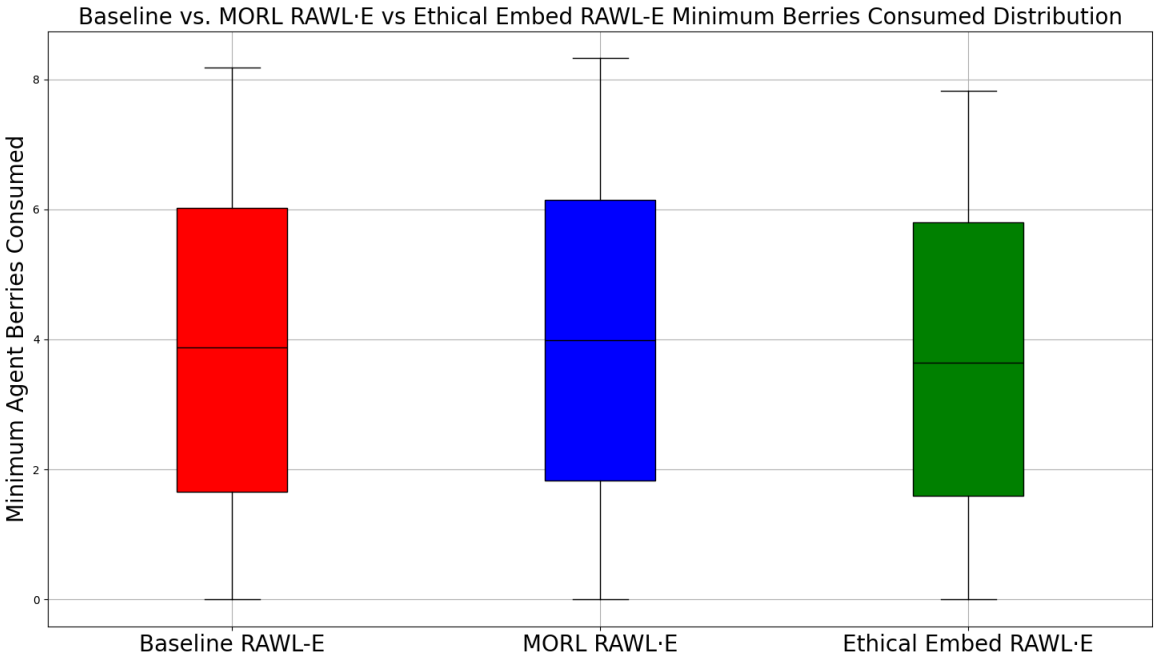


Figure 4.8: **Box plot of the distribution of minimum experience of the berries eaten by agents in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

objective has its own head in the DQN of the MORL RAWL-E agents. This means that the potential for conflicting gradient updates, which can happen when objectives are combined into a single output, is reduced. With a separate head for the ethical objective, the ethical gradient is better preserved in its own loss function component, decreasing the chance that its update direction will be opposed by the gradient of the individual objective [52]. This makes the ethical gradient less likely to get diluted during training, unlike in the baseline RAWL-E model. From this we can conclude that the MORL RAWL-E agents learn ethical objectives more effectively than baseline RAWL-E agents, leading them to choose actions that lead to fairer outcomes in the society, which is reflected in their better performance on the fairness metrics.
Contrary to our expectations, the Ethical Embed RAWL-E model did not outperform the MORL RAWL-E model on fairness metrics. Reasons for this include the fact that the model uses a function approximation mechanism to find the weight to satisfy the ethical embedding problem, as defined in Definition 3.3.2. The ethical-optimal policy proof [2] assumes tabular Q-values, but with a DQN the estimated ethical-optimal policy is only approximate. In practice, the computed $w_{ethical}$ can be too small to dominate or so large that it overwhelms the individual dimension in the action selection. The fairness performance deficit can also be attributed to the fact that when $w_{ethical}$ increases, it should lead to more ethical actions being chosen. This may not be the case since, by prioritising the ethical objective, the individual objective may be ignored leading to agents failing to sustain themselves, so the group may be harmed in the pursuit of fairness [53]. The more agents die, the lower the minimum experience will be for that society and any agent that at least keeps enough berries to stay alive can continue foraging, leading to the overall Gini index rising.

4.2.3 Social Welfare

Based on results in Table 4.3, we can see the impact on the social welfare of the society between the baseline RAWL-E model and the MORL RAWL-E model. In the capabilities harvest scenario, the mean social welfare for ag_{wb} shows a small increase from 38.35 to 38.73. This also corresponds to a negligible effect size of 0.06. Interestingly, the mean social welfare for $ag_{b_{consumed}}$ slightly decreases from 17.68 to 17.27, with a negligible negative effect size of -0.04. This indicates that slightly fewer than a half of MORL RAWL-E episodes show a higher social welfare for berries consumed relative to the base model.
Looking at the allotment harvest simulation, the social welfare metrics exhibit similar slight improvements. The mean social welfare for ag_{wb} increases from 57.92 to 58.45 from the baseline RAWL-E model to the MORL RAWL-E model. This difference also yields a negligible effect size of 0.04. Similarly, the mean social welfare for $ag_{b_{consumed}}$ increases marginally from 20.23 to 20.45. The effect size of 0.01 is negligible.
From hypothesis 5, our null-hypothesis 1 represents the social welfare difference being positive, so the MORL RAWL-E model having a greater social welfare. Null-hypothesis 2 represents the difference being negative, so the base RAWL-E model having a greater social welfare. We find that $p < 0.01$ for null hypothesis 1 and 2 with ag_wb and $ag_{b_{consumed}}$, leading us to fail to reject hypothesis 5. As a result, we conclude that norms emerging from the MORL RAWL-E model lead to the same social welfare as the norms from the base RAWL-E model. This does support our original hypothesis of equivalence in the social welfare between the MORL RAWL-E and RAWL-E models.

We can also observe the impact on the social welfare of the society between the MORL RAWL-E model and the Ethical Embed RAWL-E model. In the capabilities harvest scenario, the mean social welfare for ag_{wb} drops from 38.73, in the MORL RAWL-E model, to 34.93 in the Ethical Embed RAWL-E model. This corresponds to an effect size of -0.77, which is medium sized. This suggests that about 77% of the MORL RAWL-E episodes show higher well-being social welfare compared to the Ethical Embed RAWL-E model. Likewise, the mean social welfare for $ag_{b_{consumed}}$ decreases from 17.27 to 16.31, resulting in a negligible effect size of -0.09.
Looking at the allotment harvest simulation, the mean social welfare of ag_{wb} changes from 58.45 to 51.13. The medium effect size of -0.59, suggests that about 72% of MORL RAWL-E episodes show higher social welfare for well-being compared to the Ethical Embed RAWL-E model. We also see that the social welfare for $ag_{b_{consumed}}$ drops from 20.45 to 19.21, producing a negligible effect size of -0.10.
A more detailed visualisation of the results is displayed in Figures 4.9, 4.11, 4.10 and 4.12.
Examining hypothesis 6, we find that $p > 0.01$ for ag_{wb} and $ag_{b_{consumed}}$, so we cannot reject the null-hypothesis corresponding to hypothesis 6, leading us to conclude that the Ethical Embed model does not lead to a higher social welfare compared to the MORL RAWL-E model

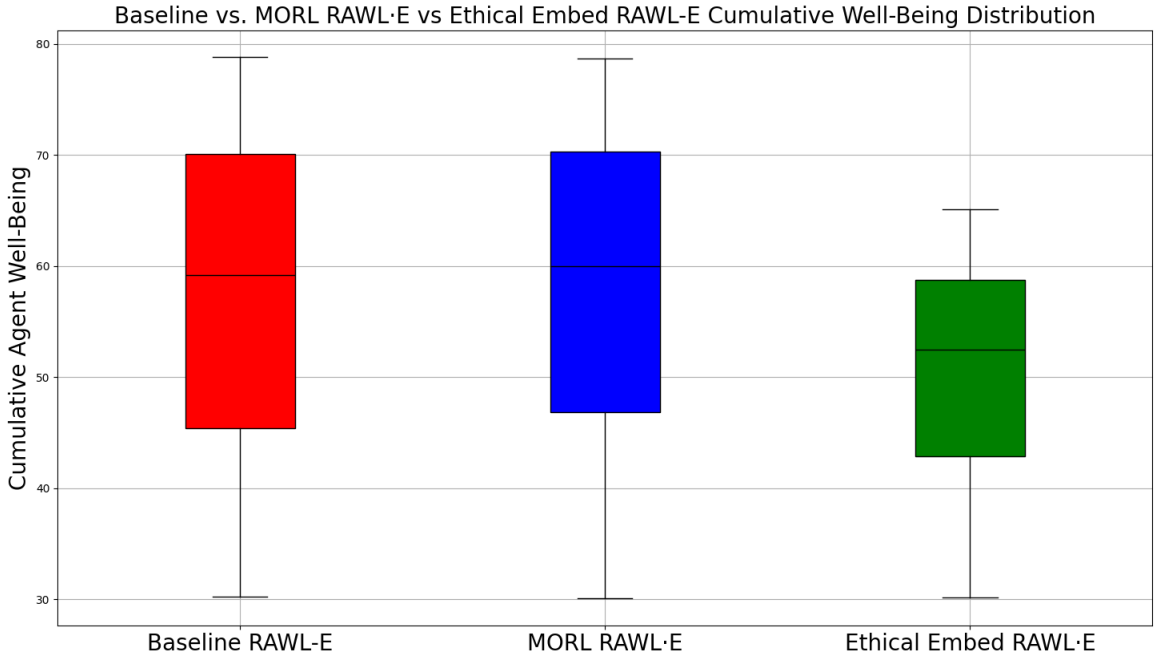


Figure 4.9: **Box plot of the distribution of total agent well-being in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

4.2.4 Durability

The durability of the models are also summarised in Table 4.3. Between the baseline RAWL-E and MORL RAWL-E models, see that for the capabilities harvest there is a slight decrease in the average length of the episodes from 39.24 to 38.24. This corresponds to a negligible effect size of -0.02.
In the allotment harvest simulation we see that there is an increase in the average length of episodes, between these models, from 44.95 to 46.42. The effect size of this is also negligible at 0.14.
From hypothesis 7, our null-hypothesis 1 represents the durability difference being positive, so the MORL RAWL-E model having a greater durability. Null-hypothesis 2 represents the difference being negative, so the base RAWL-E model having a greater durability. We find that $p < 0.01$ for null hypothesis 1 and 2, leading us to fail to reject hypothesis 7. As a result, we can conclude that norms emerging from the MORL RAWL-E model lead to the same durability as the norms from the base RAWL-E model. This supports our original hypothesis of equivalence in the durability between the MORL RAWL-E and RAWL-E models.

We can also examine the mean duration between the MORL RAWL-E model and the Ethical Embed RAWL-E model for the capabilities harvest scenario in Table 4.3. The mean episode duration decreases from 38.84 to 31.18. The medium effect size value of -0.52 suggests that about 70% of episodes in the MORL RAWL-E model had a longer duration compared to the Ethical Embed RAWL-E model.
For the allotment harvest scenario, the mean episode duration also decreases from 46.42 to 42.22. The small effect size of -0.37 suggests that about 64% of the episodes in the MORL RAWL-E model had a longer duration compared to the Ethical Embed RAWL-E model.
A more detailed visualisation of the results is displayed in Figures 4.13 and 4.14.
The analysis of hypothesis 8 revealed that $p > 0.01$ for ag_{wb} and $ag_{b_{consumed}}$. Therefore, we cannot reject the null-hypothesis corresponding to hypothesis 8 leading us to conclude that the Ethical Embed model does not lead to a higher durability compared to the MORL RAWL-E model

Since the MORL RAWL-E model improves overall fairness of societies compared to the baseline RAWL-E model, we expect that it leads to more sustainable societies than the baseline model. While it slightly boosts the mean social welfare in both scenarios and also increases the duration of the allotment harvest scenario, from our results we conclude that the MORL RAWL-E model leads to the same sustainability as the baseline RAWL-E model. I believe this happens because fairness improvements plateau

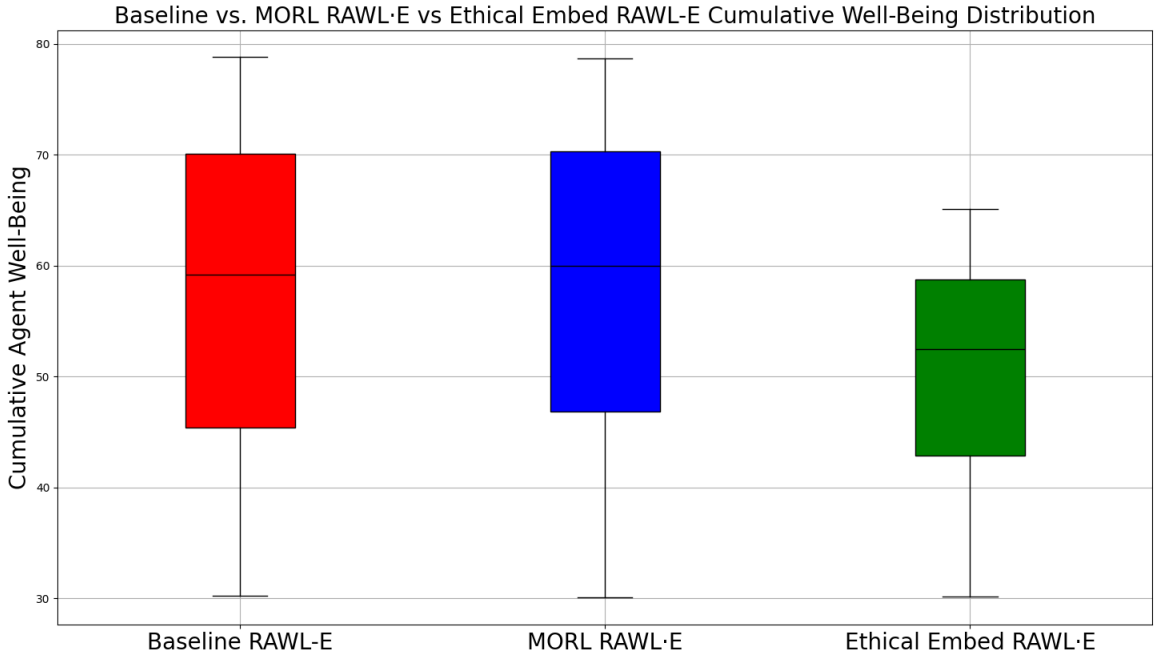


Figure 4.10: **Box plot of the distribution of total agent well-being in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

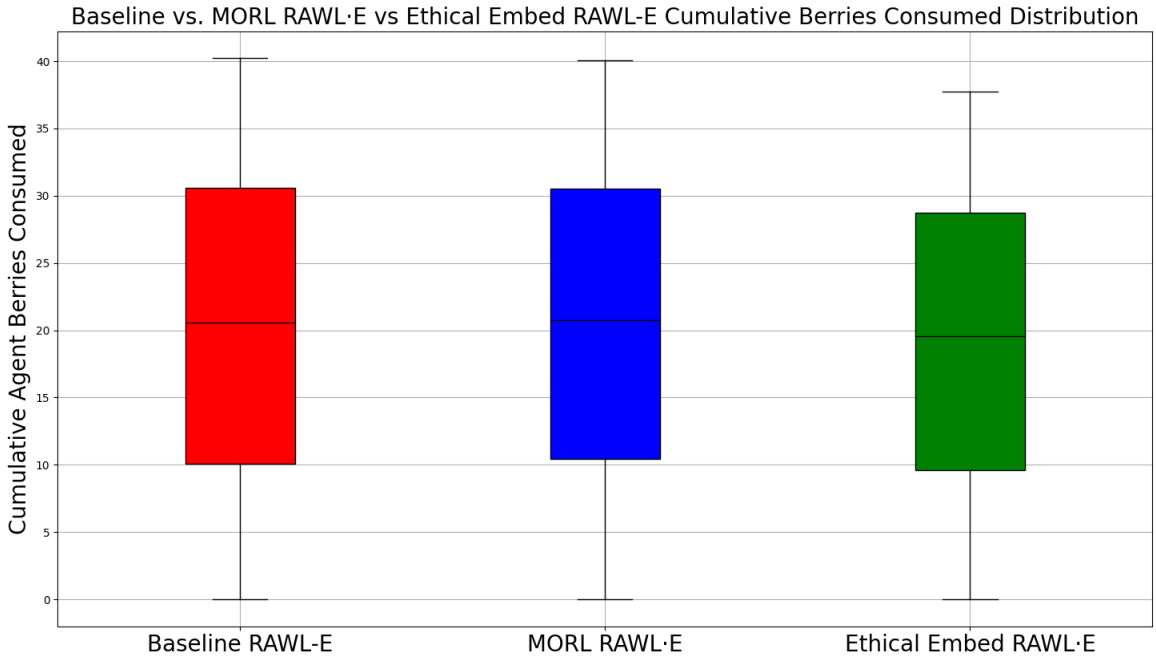


Figure 4.11: **Box plot of the distribution of total berries consumed in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

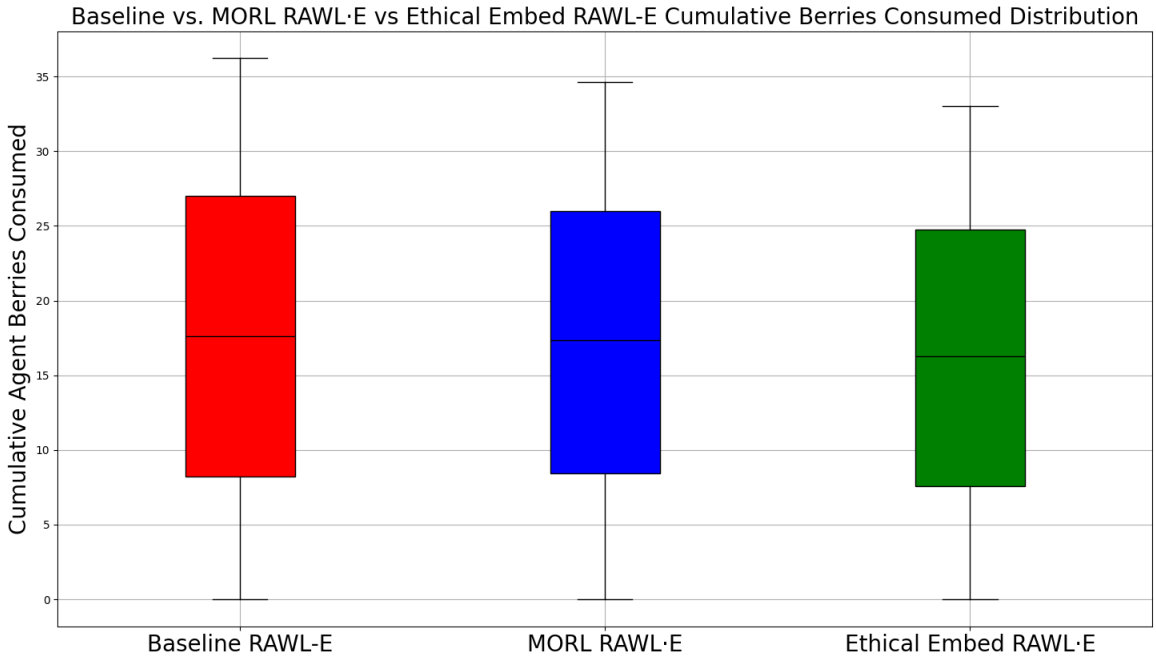


Figure 4.12: **Box plot of the distribution of total berries consumed in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

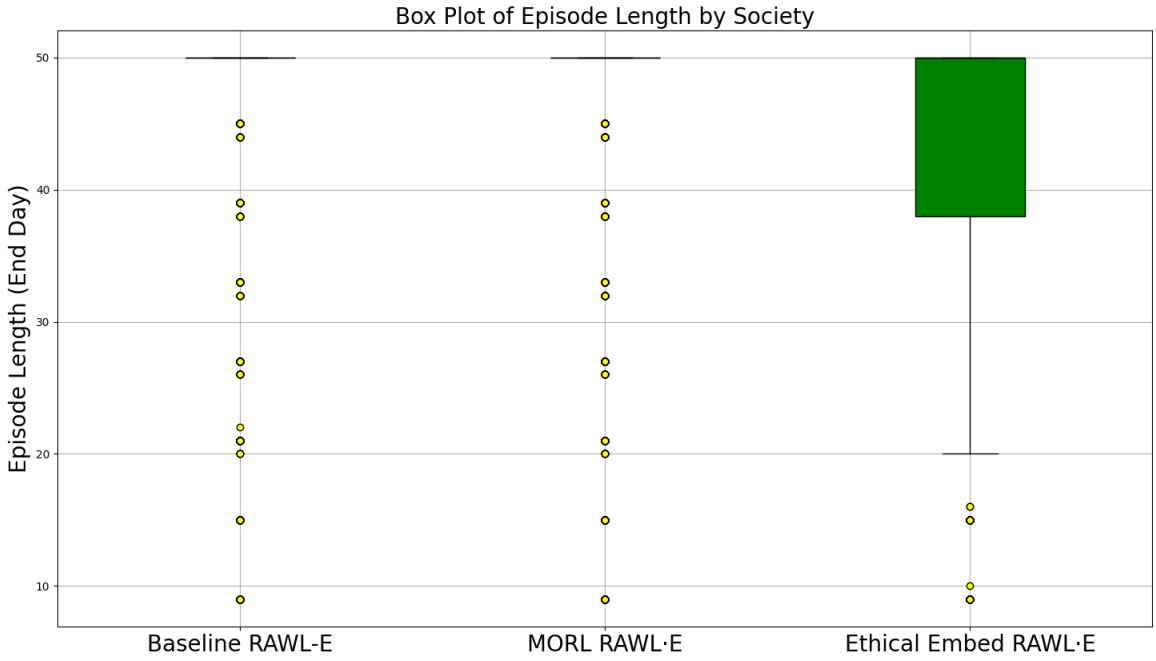


Figure 4.13: **Box plot of the distribution of the episode lengths in the allotment harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

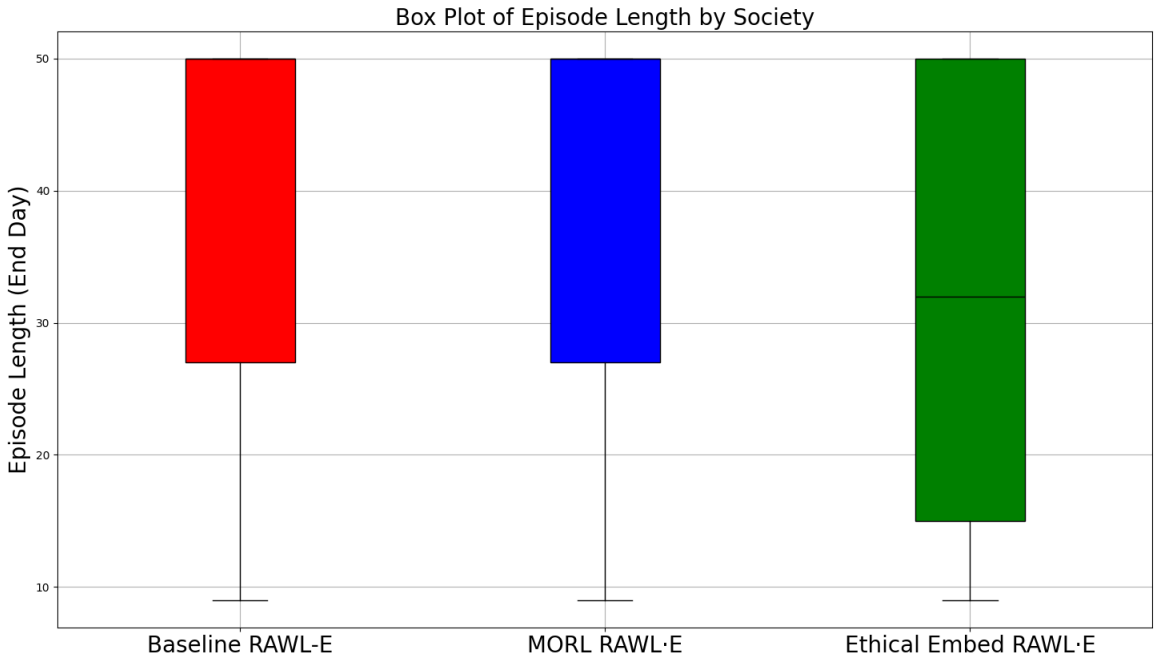


Figure 4.14: **Box plot of the distribution of the episode lengths in the capabilities harvest for the Baseline RAWL-E, MORL RAWL-E, and Ethical Embed RAWL-E models.**

in their ability to drive longer episodes or higher social welfare, so you end up with much higher fairness but only negligible gains in sustainability. In effect this is a diminishing returns trade-off between fairness and sustainability, because once you've improved the worst off agents up to a certain point, any further redistribution of berries can't meaningfully extend the societies duration or increase overall social welfare. The Ethical Embed RAWL-E model performed worse in fairness compared to the MORL RAWL-E model so as expected this causes a lower welfare, duration and overall sustainability. This means that more worse off agents fail to gather or conserve enough berries for survival, shortening episode length, reducing total welfare.

Scenario	Metric	Var.	Mean $\bar{x}$		
			RAWL·E	MORL RAWL·E	Ethical Embed
Capabilities Harvest	Inequality	ag_{wb}	0.17	0.12	0.11
		$ag_{b_{\text{cons}}}$	0.10	0.08	0.10
	Min. experience	ag_{wb}	7.36	7.81	6.56
		$ag_{b_{\text{cons}}}$	3.88	4.02	3.93
	Social welfare	ag_{wb}	38.35	38.73	34.93
		$ag_{b_{\text{cons}}}$	17.68	17.27	16.31
	Duration	–	39.24	38.84	31.18
Allotment Harvest	Inequality	ag_{wb}	0.22	0.17	0.18
		$ag_{b_{\text{cons}}}$	0.15	0.11	0.13
	Min. experience	ag_{wb}	7.37	10.84	8.37
		$ag_{b_{\text{cons}}}$	3.80	4.45	4.05
	Social welfare	ag_{wb}	57.92	58.45	51.13
		$ag_{b_{\text{cons}}}$	20.23	20.45	19.21
	Duration	–	44.95	46.42	42.22

(a) Mean table

Scenario	Metric	Var.	Std. dev. σ		
			RAWL·E	MORL RAWL·E	Ethical Embed
Capabilities Harvest	Inequality	ag_{wb}	0.06	0.03	0.03
		$ag_{b_{\text{cons}}}$	0.08	0.10	0.14
	Min. experience	ag_{wb}	0.80	1.10	0.47
		$ag_{b_{\text{cons}}}$	2.55	2.56	2.47
	Social welfare	ag_{wb}	8.23	5.58	4.20
		$ag_{b_{\text{cons}}}$	11.08	10.51	10.20
	Duration	–	14.77	14.61	15.03
Allotment Harvest	Inequality	ag_{wb}	0.07	0.04	0.04
		$ag_{b_{\text{cons}}}$	0.08	0.08	0.13
	Min. experience	ag_{wb}	0.37	2.00	0.91
		$ag_{b_{\text{cons}}}$	2.45	2.71	2.63
	Social welfare	ag_{wb}	14.62	14.31	9.96
		$ag_{b_{\text{cons}}}$	12.18	12.05	11.45
	Duration	–	10.88	9.39	12.81

(b) Standard deviation table

Scenario	Metric	Var.	Cohen's d	
			BR–MR	MR–EE
Capabilities Harvest	Inequality	ag_{wb}	1.10	0.46
		$ag_{b_{\text{cons}}}$	0.24	–0.16
	Min. experience	ag_{wb}	0.48	–1.52
		$ag_{b_{\text{cons}}}$	0.05	–0.03
	Social welfare	ag_{wb}	0.06	–0.77
		$ag_{b_{\text{cons}}}$	0.04	–0.09
	Duration	–	–0.02	–0.52
Allotment Harvest	Inequality	ag_{wb}	1.00	–0.32
		$ag_{b_{\text{cons}}}$	0.45	–0.24
	Min. experience	ag_{wb}	2.45	–1.61
		$ag_{b_{\text{cons}}}$	0.25	–0.15
	Social welfare	ag_{wb}	0.04	–0.59
		$ag_{b_{\text{cons}}}$	0.01	–0.10
	Duration	–	0.14	–0.37

(c) Cohen's d table

Table 4.3: **Split summary comparing inequality, minimum experience, and robustness of different RAWL·E society models in capabilities harvest and allotment harvest scenarios.** For each table, the first two columns of each table specify the scenario (capabilities or allotment harvest) and the performance metric (inequality, minimum experience, social welfare, duration), along with the variable (agent well-being (ag_{wb}) or berries consumed by agent ($ag_{b_{consumed}}$)). The (a) Mean ($\bar{x}$), (b) Standard deviation (σ) and (c) Cohen's d are gathered for each variable and metric combination in the Capabilities and Allotment harvest scenarios. The Cohen's compares the baseline to the MORL RAWL-E model (BR–MR) and the MORL to the Ethical Embed RAWL-E model (MR–EE).

Scenario	**Metric**	**Mean $\bar{x}$**			**Std Dev σ**		
		RAWL-E	M-RAWL·E	E-Embed	RAWL-E	M-RAWL·E	E-Embed
Capabilities	Fitness	34.81	74.40	77.40	40.01	114.64	132.09
	Numerosity	24.50	27.80	24.96	15.97	32.77	28.80
Allotment	Fitness	53.28	65.17	93.43	56.58	62.04	82.92
	Numerosity	30.70	43.88	37.42	25.70	41.45	39.32

Table 4.4: **Norm Metrics Across scenarios and methods.** The first two columns specify the scenario (capabilities or allotment harvest) and the metric (fitness and numerosity). The next six columns report the mean ($\bar{x}$) and standard deviation (σ) under the RAWL-E, MORL RAWL-E and Ethical Embed RAWL-E models.

Chapter 5

Critical Evaluation

In this chapter we critically examine the challenges of our implementations, assess each model's strengths and weaknesses against our objectives, and highlight lessons for future refinements of our models.

5.1 Implementation Critique

`Tensorflow` was used to implement the neural network of all the models for the DQN of the agents. The `Mesa` library was used for the agent-based modelling framework because of its easy to use modular architecture, which allowed for customisation to build upon the implemented RAWL-E model. The base RAWL-E model was taken from Woodgate et. al [1] and adapted to include dynamical ethical rewards/-sanctions based on the magnitude of the difference of resulting actions evaluated based on fairness. The workings of the novel models were influenced by previous work in the field of reinforcement learning and ethical AI [8], [18], [49]. Since the RAWL-E model developed by Woodgate et. al was successful, I felt confident to use the leverage the codebase. I modified the existing functions and incorporated additional functionalities to construct my other models.

5.2 Model Evaluation

5.2.1 Baseline RAWL-E

The baseline RAWL-E model, adapted from Woodgate et al. [1], serves as a foundational model incorporating ethical considerations within a reinforcement learning context. The model's strength lies in its ability to integrate dynamic ethical rewards and sanctions based on fairness well. It's major weakness arises from combining ethical and individual rewards into a single grouped Q-value. This process sometimes leads to the ethical objective being overshadowed during action selection, particularly when ethical rewards are small or noisy compared to the individual rewards.

While the simplicity and abstraction of our scenarios limit real-world applicability, this design choice allows us to focus on demonstrating the operationalisation of normative ethics rather than achieving realism. We mitigate this drawback by presenting our agent architecture as decoupled from the environment.

5.2.2 MORL Model

The concept underpinning the MORL RAWL-E model involved adapting RAWL-E agents to incorporate multi-objective reinforcement learning, utilising a DQN to simultaneously learn distinct reward dimensions for both individual and ethical objectives. Contrary to initial expectations that it would perform merely on par with the baseline RAWL-E model, this model surprisingly demonstrated superior performance on fairness metrics, achieving lower inequality and higher minimum experience in both tested scenarios. Fairness improved likely because the DQN's separate output heads made it less likely that the ethical gradient during training was diluted. This suggests that the structure of the model, with its two-dimensional Q-value output and scalarisation for action selection, facilitated more effective learning of the ethical objective.

5.2.3 Ethical Embed Model

The concept behind the Ethical Embed RAWL-E model was to guarantee the learning of ethical policies and improving the overall welfare and sustainability of the societies as a result. This aimed to ensure the agents prioritises policies that are ethically optimal, as defined by maximising the ethical objective's Q-value dimension and then the individual objective. Surprisingly, this model did not outperform the MORL RAWL-E model on fairness and sustainability metrics. The fairness performance deficit can be attributed to the fact that the model uses a function approximation to find the weight leading to an ethical weight that is either too small or large. Also, there is a potential that the individual objective is being ignored when prioritising the ethical objective. This is reflected in Table 4.4, where the Ethical Embed model shows a higher fitness for learned behaviours, indicating that they are more rewarding, which is the result of a higher ethical weight on the reward vector. The lower numerosity, suggesting that these behaviours are not being adopted as frequently as in the MORL RAWL-E model since the struggle for survival caused by ignoring the individual objective directly reduces the opportunities for any behaviour, ethical or otherwise, to be performed repeatedly. Even though ethical embedding can be powerful in theory, it is unsuited for this specific implementation with a DQN due to approximation issues and the complexity of the environment.

5.3 Outcome

The ultimate result of this project involves a deeper understanding of how the architecture of a learning agent through implementing multi-objective reinforcement learning and ethical embedding techniques influence norm emergence and societal outcomes in multi-agent systems.
Our results have shown that the MORL RAWL-E model represents a significant improvement over the baseline RAWL-E model to promote fairness within the society. While its impact on sustainability was less pronounced, the model's ability to achieve better fairness outcomes by explicitly learning distinct objectives demonstrates its potential as a strong candidate for further work in developing ethical multi-agent systems.
The Ethical Embed RAWL-E model shows worse overall results compared to the MORL RAWL-E model for promoting fairness within a society. It also achieved lower sustainability indicating that my implementation of this model was not suitable for encouraging more ethical behaviours in the RAWL-E environment.

5.4 Future Work

There is huge potential for ethical multi-agent systems research to proceed in several directions. Our finding that the MORL RAWL-E model is not only as effective as the baseline RAWL-E model (with a single DQN output head) but also contributes to the creation of more just and sustainable societies suggests a valuable method for exploring ethical policy and reward shaping by building on top of the MORL RAWL-E model.
One possible path of research could explore the effects of promoting fairness in MAS, could be by utilising thresholded lexicographic ordering (TLO) [49] instead of ethical embedding. This approach operates by prioritising one objective while ensuring that other objectives meet specific threshold levels. By ensuring a minimum threshold of the individual objective is being met, the problems presented by the Ethical Embed RAWL-E model can potentially be resolved.
Another avenue would could involve investigating the MORL approach in more complex environments. Real-world dilemmas rarely involve a single ethical consideration as was done in this project. Future research should explore advanced MORL techniques that can effectively handle complex trade-offs between more than one ethical objective [54], such as safety and accountability in the RAWL-E environment.

Chapter 6

Conclusion

This dissertation began with the baseline RAWL-E society adapted from Woodgate et al. [1] by using dynamic ethical sanctions.
Building upon this reference point, I designed two novel implementations to be tested with this agent architecture.
The first, MORL RAWL-E, inspired by [49], splits the Q-network's output into separate individual and ethical reward heads and learns them jointly with multi-objective RL scalarisation. The second, inspired by Rodriguez-Soto et al. [2], is the Ethical-Embed RAWL-E model which keeps the MORL architecture but adds a learnt weight that tries to guarantee the ethical head dominates at action-selection time.
All three models were trained for 2000 episodes and compared on four fairness and sustainability metrics (Gini, minimum experience, social welfare, duration).
Across both environments, the MORL variant consistently out-performed the baseline on fairness without hurting the mean social welfare or the duration. I attribute this to the twin-head network preserving an undiluted ethical gradient during learning.
By contrast, the Ethical-Embed model fell short. The increased ethical weighting often starved agents of the resources needed to stay alive, leading to higher inequality in several cases and shorter, less sustainable episodes. Fitness values for learnt behaviours were higher, showing the weight was indeed large, but numerosity dropped, signalling poor norm uptake.
The experiments validate MORL as a practical route to fairer MAS and the Ethical Embed model casts doubt on an increased ethical weighting in this environment. From this we propose two future directions: swap the ethical embedding framework for thresholded lexicographic ordering to stop the ethical head from eclipsing basic survival, and push MORL RAWL-E into richer domains with multiple, possibly conflicting, ethical dimensions.

Bibliography

[1] J. Woodgate, P. Marshall, and N. Ajmeri, "Operationalising Rawlsian Ethics for Fairness in Norm-Learning Agents," Tech. Rep., 2025. [Online]. Available: www.aaai.org.

[2] M. Rodriguez-Soto, M. Lopez-Sanchez, and J. A. Rodriguez-Aguilar, "Multi-Objective Reinforcement Learning for Designing Ethical Environments," Tech. Rep., 2021.

[3] J. Rawls, *Justice as Fairness: A Restatement*, E. Kelly, Ed. Harvard University Press, 2001, ISBN: 9780674005105. DOI: 10.2307/j.ctv31xf5v0. [Online]. Available: http://www.jstor.org/stable/j.ctv31xf5v0.

[4] *Multi-agent systems: How can research in multi-agent systems help us to address challenging real-world problems?* 2025. [Online]. Available: https://www.turing.ac.uk/research/interest-groups/multi-agent-systems#:~:text=Multi,robot%20factories%2C%20automated.

[5] T. Simonini, *Hugging Face Deep RL Course*, 2018. [Online]. Available: https://huggingface.co/learn/deep-rl-course.

[6] V. Kumar, *How can research in multi-agent systems help us to address challenging real-world problems?* Dec. 2024. [Online]. Available: https://adasci.org/all-you-need-to-know-about-multi-agent-reinforcement-learning/#:~:text=%23%20a.%20Non.

[7] Y. Shoham and K. Leyton-Brown, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations," Tech. Rep. [Online]. Available: http://www.masfoundations.org..

[8] D. M. Roijers, P. Vamplew, S. Whiteson, A. W. Nl, and R. Dazeley, "A Survey of Multi-Objective Sequential Decision-Making," Tech. Rep., 2013, pp. 67–113.

[9] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, "Q-Learning Algorithms: A Comprehensive Classification and Applications," *IEEE Access*, vol. 7, pp. 133 653–133 667, 2019. DOI: 10.1109/ACCESS.2019.2941229.

[10] T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," *J. Artif. Int. Res.*, vol. 13, no. 1, pp. 227–303, Nov. 2000, ISSN: 1076-9757.

[11] R. Kalra, *Understanding Model-Free Reinforcement Learning*, 2024. [Online]. Available: https://medium.com/@kalra.rakshit/understanding-model-free-reinforcement-learning-9958a09f24f8.

[12] D. Kim, *what is Temporal Difference Learning?* Mar. 2023. [Online]. Available: https://medium.com/@chkim345/what-is-temporal-difference-learning-4c4c040613aa.

[13] M. Irodova and R. H. Sloan, "Reinforcement Learning and Function Approximation," in *The Florida AI Research Society*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:815827.

[14] Tashmit, *Epsilon Greedy Algorithm.*

[15] T. Hester, M. Vecerik, O. Pietquin, *et al.*, "Deep Q-learning From Demonstrations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. DOI: 10.1609/aaai.v32i1.11757. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11757.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Playing Atari with deep reinforcement learning," Dec. 2013.

[17] L. Lin, "Reinforcement Learning for Robots Using Neural Networks," 1992. [Online]. Available: https://api.semanticscholar.org/CorpusID:60875658.

[18] C. F. Hayes, R. Rădulescu, E. Bargiacchi, *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, p. 26, 2022, ISSN: 1573-7454. DOI: 10.1007/s10458-022-09552-y. [Online]. Available: https://doi.org/10.1007/s10458-022-09552-y.

[19] R. Sutton, "The reward hypothesis,"

[20] N. Criado, E. Argente, and V. Botti, *Open issues for normative multi-agent systems*, 2011. DOI: 10.3233/AIC-2011-0502.

[21] G. Boella, L. van der Torre, and H. Verhagen, "Introduction to normative multiagent systems," *Computational and Mathematical Organization Theory*, vol. 12, no. 2-3 SPEC. ISS. Pp. 71–79, 2006, ISSN: 15729346. DOI: 10.1007/s10588-006-9537-7.

[22] G. Boella and L. van der Torre, "Substantive and procedural norms in normative multiagent systems," *Journal of Applied Logic*, vol. 6, no. 2, pp. 152–171, Jun. 2008, ISSN: 15708683. DOI: 10.1016/j.jal.2007.06.006.

[23] B. Tony, R. Savarimuthu, and S. Cranefield, "A categorization of simulation works on norms," Tech. Rep. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2009/1905.

[24] R. Conte[1], G. Andrighetto[1], M. Campennì[1], and M. Paolucci[1], "Emergent and Immergent Effects in Complex Social Systems," Tech. Rep., 2007. [Online]. Available: http://labss.istc.cnr.it.

[25] G. Boella and L. Van Der Torre, "The Social Delegation Cycle," Tech. Rep.

[26] A. Morris-Martin, M. De Vos, and J. Padget, "Norm emergence in multiagent systems: a viewpoint paper," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 706–749, Nov. 2019, ISSN: 15737454. DOI: 10.1007/s10458-019-09422-0.

[27] B. Savarimuthu, S. Cranefield, M. Purvis, and M. (Purvis, "Role Model Based Mechanism for Norm Emergence in Artificial Agent Societies," in *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, J. Sichman, J. Padget, S. Ossowski, and P. Noriega, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 203–217, ISBN: 978-3-540-79003-7.

[28] M. J. Hoffmann, "Entrepreneurs and norm dynamics: An agent-based model of the norm life cycle," *Penn's Agent-Based Modeling Laboratory (PAMLA). http://www. polisci. upenn. edu/abir/private/-Pamla/Hoffmann norms. doc*, 2003.

[29] C. Cordova, J. Taverner, E. D. Val, and E. Argente, "A systematic review of norm emergence in multi-agent systems," Tech. Rep., 2024.

[30] D. Villatoro, S. Sen, and J. Sabater-Mir, "Topology and Memory Effect on Convention Emergence," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, 2009, pp. 233–240. DOI: 10.1109/WI-IAT.2009.155.

[31] T. Bench-Capon and S. Modgil, "Norms and value based reasoning: justifying compliance and violation," *Artificial Intelligence and Law*, vol. 25, no. 1, pp. 29–64, Mar. 2017, ISSN: 15728382. DOI: 10.1007/s10506-017-9194-9.

[32] H. Hart and L. Greene, *The Concept of Law*. Oxford University Press, 2012.

[33] D. Lewis, *Convention: A Philosophical Study*. John Wiley & Sons, 2008.

[34] S.-T. Tzeng, N. Ajmeri, and M. P. Singh, "Fleur: Social Values Orientation for Robust Norm Emergence," in *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XV*, Springer International Publishing, 2022, pp. 185–200, ISBN: 978-3-031-20845-4.

[35] E. Argente, E. D. Val, D. Pérez-García, and V. Botti, "Normative Emotional Agents: A Viewpoint Paper," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1254–1273, 2022. DOI: 10.1109/TAFFC.2020.3028512.

[36] B. Savarimuthu, R. Arulanandam, and M. Purvis, "Aspects of Active Norm Learning and the Effect of Lying on Norm Emergence in Agent Societies," in *Agents in Principle, Agents in Practice*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 36–50, ISBN: 978-3-642-25044-6.

[37] C. Castelfranchi, "Prescribed mental attitudes in goal-adoption and norm-adoption," Tech. Rep., 1999, pp. 37–50.

[38] J. Allen, R. Fikes, and E. Sandewall, "Modeling Rational Agents within a BDI-Architecture," Tech. Rep., 1991.

[39] S. Hu and H.-f. Leung, "Achieving Coordination in Multi-Agent Systems by Stable Local Conventions under Community Networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4731–4737. DOI: 10.24963/ijcai.2017/659. [Online]. Available: https://doi.org/10.24963/ijcai.2017/659.

[40] M. Campenni, G. Andrighetto, F. Cecconi, and R. Conte, "Normal = Normative? The Role of Intelligent Agents in Norm Innovation," *Mind & Society*, vol. 8, pp. 153–172, Apr. 2009. DOI: 10.1007/s11299-009-0063-4.

[41] B. T. R. Savarimuthu and S. Cranefield, "Norm creation, spreading and emergence: A survey of simulation models of norms in multi-agent systems," *Multiagent and Grid Systems*, vol. 7, no. 1, pp. 21–54, 2011. DOI: 10.3233/MGS-2011-0167. [Online]. Available: https://journals.sagepub.com/doi/abs/10.3233/MGS-2011-0167.

[42] A. Weiner, *Contestation and Constitution of Norms in Global International Relations*. Cambridge University Press, 2018.

[43] R. Boyd and P. J. Richerson, *Culture and the Evolutionary Process*.

[44] J. S. Santos, J. O. Zahn, E. A. Silvestre, V. T. Silva, and W. W. Vasconcelos, "Detection and resolution of normative conflicts in multi-agent systems: a literature survey," *Autonomous Agents and Multi-Agent Systems*, vol. 31, no. 6, pp. 1236–1282, 2017, ISSN: 1573-7454. DOI: 10.1007/s10458-017-9362-z. [Online]. Available: https://doi.org/10.1007/s10458-017-9362-z.

[45] H. Ashrafian, "Engineering a social contract: Rawlsian distributive justice through algorithmic game theory and artificial intelligence," *AI and Ethics*, vol. 3, no. 4, pp. 1447–1454, 2023, ISSN: 2730-5961. DOI: 10.1007/s43681-022-00253-6. [Online]. Available: https://doi.org/10.1007/s43681-022-00253-6.

[46] Arneson, Richard and Bidadanure, Juliana and Axelsen, and David, "Egalitarianism," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, Eds., Spring 2025, Metaphysics Research Lab, Stanford University, 2025. [Online]. Available: https://seop.illc.uva.nl/entries/egalitarianism/#Bib.

[47] C. Zhang and J. A. Shah, "Fairness in Multi-Agent Sequential Decision-Making," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5556d1d6ca0d004accf36cc2db73e736-Paper.pdf.

[48] M. Zimmer, U. Siddique, and P. Weng, "Learning Fair Policies in Decentralized Cooperative Multi-Agent Reinforcement Learning," in *International Conference on Machine Learning*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229298110.

[49] T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim, "A multi-objective deep reinforcement learning framework," *Engineering Applications of Artificial Intelligence*, vol. 96, p. 103 915, 2020, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2020.103915. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197620302475.

[50] V. Charles, T. Gherman, and J. C. Paliza, "The Gini Index: A Modern Measure of Inequality," Tech. Rep., 2022. [Online]. Available: https://worldpopulationreview.com/country-rankings/gini-coefficient-by-country.

[51] S. Dong, C. Li, S. Yang, B. An, W. Li, and Y. Gao, "Egoism, utilitarianism and egalitarianism in multi-agent reinforcement learning," *Neural Networks*, vol. 178, Oct. 2024, ISSN: 18792782. DOI: 10.1016/j.neunet.2024.106544.

[52] M. Crawshaw, *Multi-Task Learning with Deep Neural Networks: A Survey*, Apr. 2020. DOI: 10.48550/arXiv.2009.09796.

[53] J. Blandin and I. Kash, "Group Fairness in Reinforcement Learning via Multi-Objective Rewards," *Transactions on Machine Learning Research*, 2024.

[54] T. Deschamps, R. Chaput, and L. Matignon, "Multi-objective reinforcement learning: an ethical perspective," Tech. Rep., 2024. [Online]. Available: https://orcid.org/0009-0000-1877-1469,.

Appendix A

Appendix A: AI Prompts/Tools

A.0.1 Understanding Concepts

I used ChatGPT to help me understand certain concepts I was unfamiliar with or found explanations in papers to be too high-level. I used this for the concepts of Deep Q-learning, Multi-objective reinforcement learning and ethical embedding. An example prompt is: "Can you explain how an ethical-optimal policy can be found from the partial convex hull?"

A.0.2 Understanding Code

I used ChatGPT to help me understand parts of the code that I was unfamiliar with. I used this for the codebase from Woodgate et al. [1] and Rodriguez-Soto et al. [2]. An example prompt is "Can you explain how norms are stored and replaced in the norms module of the RAWL-E paper?"

A.0.3 Other tools

I also used the Grammarly web extension to provide an AI spelling and grammar checker for my dissertation. This did not require any prompt.