applied sciences

MDPI

Article

Promoting the Emergence of Behavior Norms in a Principal–Agent Problem—An Agent-Based Modeling Approach Using Reinforcement Learning

Saeed Harati [1,*], Liliana Perez [1] and Roberto Molowny-Horas [2]

[1] Laboratory of Environmental Geosimulation (LEDGE), Department of Geography, Université de Montréal, 1375 Ave. Thérèse-Lavoie-Roux, Montreal, QC H2V 0B3, Canada; l.perez@umontreal.ca
[2] Centre de Recerca Ecològica i Aplicacions Forestals (CREAF), Bellaterra, E-08193 Cerdanyola de Vallès, Catalonia, Spain; roberto@creaf.uab.es
* Correspondence: saeed.harati.asl@umontreal.ca

Abstract: One of the complexities of social systems is the emergence of behavior norms that are costly for individuals. Study of such complexities is of interest in diverse fields ranging from marketing to sustainability. In this study we built a conceptual Agent-Based Model to simulate interactions between a group of agents and a governing agent, where the governing agent encourages other agents to perform, in exchange for recognition, an action that is beneficial for the governing agent but costly for the individual agents. We equipped the governing agent with six Temporal Difference Reinforcement Learning algorithms to find sequences of decisions that successfully encourage the group of agents to perform the desired action. Our results show that if the individual agents' perceived cost of the action is low, then the desired action can become a trend in the society without the use of learning algorithms by the governing agent. If the perceived cost to individual agents is high, then the desired output may become rare in the space of all possible outcomes but can be found by appropriate algorithms. We found that Double Learning algorithms perform better than other algorithms we used. Through comparison with a baseline, we showed that our algorithms made a substantial difference in the rewards that can be obtained in the simulations.

Keywords: complex systems; emergence; reinforcement learning; temporal difference learning; social status

Citation: Harati, S.; Perez, L.; Molowny-Horas, R. Promoting the Emergence of Behavior Norms in a Principal–Agent Problem—An Agent-Based Modeling Approach Using Reinforcement Learning. *Appl. Sci.* **2021**, *11*, 8368. https://doi.org/10.3390/app11188368

Academic Editor: Paola Pellegrini

Received: 23 July 2021
Accepted: 6 September 2021
Published: 9 September 2021

1. Introduction

One of the challenges of management in general, and sustainable development management in particular, is to gain the support of the individuals who are being managed. The use of incentives can be costly to managers and governments, and the use of authority is not always successful [1–4]. These problems, where a Principal (or several Principals) wishes to make an Agent (or several Agents) behave in a certain way are known as Principal–Agent problems [5].

In this study, we are interested in learning if the Principal can use recognition and the offer of good reputation to promote a new behavioral norm among the Agents. We are particularly interested in the complexities that emerge with the new norm, as the norm influences and is influenced by the decisions of the Agents. In this regard, social science literature describes a focus theory of normative conduct [6], which suggests that in making decisions, individuals consider what others do and what others approve of. We illustrate an implication of this theory in a Principal–Agent setting. We would like to see if the Agents' regard for their image in their society can lead to the emergence of a behavior norm that the Principal desires. Specifically, it is interesting for us to learn if, in absence of social sanctions and other forms of enforcement, good reputation can be a sufficient motivation for Agents to cooperate with the Principal. We would also like to see if the

Principal's intervention can hasten the emergence of this norm. This study is an effort to gain insight into the complexities that arise in an abstract Principal–Agent setting with the added consideration of normative conduct. We are curious about and intrigued by the complexities related to the abstract structure of entities, motivations, and interactions in the above setting.

Norms have been defined in various studies. For example, according to Ross [7], in a society, norms are cultural rules guiding people's behavior. Savarimuthu and Cranefield [8] consider norms as social rules that govern how certain behaviors are encouraged or condemned. In the context of institutions, Ostrom [9] writes that norms show the valuations of the actions of individuals in a society, regardless of the immediate consequences of those actions. In Crawford and Ostrom's view [10], norms are part of institutions, and deviating from them has unknown or undefined consequences. North [11] states that institutions are able to formalize norms into laws, and enforce them legally. Literature reviews report that many of the previous studies associate norms with social sanctions, or with the punishment of individuals who do not follow norms [8,12]. However, the term 'norms' has also been used in studies of the emergence of behavior expectations that do not involve sanctions [8]. Cialdini et al. [6] distinguish two types of norms, which they refer to as 'descriptive' and 'injunctive'. Descriptive norms inform the individual of what others in the society do. Injunctive norms urge the individual to do what others in the society approve of, and to avoid things of which others disapprove. According to Cialdini et al. these two types of norms come from different concepts and different motivations. Therefore, although what people do and what people approve of are often the same, separating these two norms is important in the study of normative influence. In order to avoid confusion regarding social sanctions, we follow the recommendation of Cialdini et al. Injunctive norms are associated with social sanctions, whereas descriptive norms are not. In this study, our interest is in a setting without social sanctions. Therefore, in the rest of this paper we focus on descriptive norms.

We take a complex systems approach to analyze the above problem. Complex systems are structures composed of elements, interactions and dynamics in such a way that they produce novel configurations and demonstrate surprising emerging behavior [13]. Some characteristics for complex systems are nonlinearity, self-organization, going beyond equilibrium, and existence of attractors other than a state of equilibrium, such that the combination of these characteristics can cause the emergence of new patterns in complex systems [14]. Complex systems literature uses the expression 'aggregate complexity' to refer to the interaction of system components that results in holism and synergy, with key attributes of such aggregate complexity being internal relationships, internal structure (subsystems), relationships with the environment, learning and memory, emergence, and change and evolution [15].

One of the approaches used in the study of complex systems is Agent-Based Modeling. An Agent-Based Model (ABM) consists of multiple agents acting upon their individual objectives [16]. ABMs are constructed in a bottom-up manner and allow us to compute the aggregate and large scale results of the interactions of agents with each other and with their environment [17]. As such, ABMs have been used in a wide variety of disciplines. Some examples of these applications include innovation diffusion [18], theory of cooperation [19], automated negotiation [20], recommender systems [21], migration [22], urban segregation [23–25] epidemiology [26], forest ecology [27], and species distribution [28]. ABMs have also been applied in sustainable development studies, such as in urban planning [29], sustainable transportation [30], circular economy [31], and in problems related to the tragedy of the commons [6,32].

ABMs have been extensively used in studies of social norms [8]. Some models apply the Belief–Desire–Intention (BDI) framework in the decisions of their agents [33,34]. In the BDI framework, agents have mental attributes of belief, desire, and intention, which indicate their state in terms of information, motivation, and deliberation for action, respectively [35,36]. Some of the mechanisms employed in agent-based normative simulations

are leadership [37], learning by imitation [38], machine learning and reinforcement learning [39], norm recognition [40], and reputation [41]. In a review of the literature, Hollander and Wu [12] identify areas for research and improvement. Some of those areas are norm creation and ideation, alternatives for social sanction, and the verification and validation of models [12].

Given the above context, our focus in this study is on the creation and emergence of a new descriptive norm in a setting with central leadership, where there are rewards for performing the behavior that the leadership promotes, but there are no sanctions or punishments for the non-performers. The reward in this setting is reputation and recognition as a responsible member of the society. We take an agent-based simulation approach to explore the possibility of norm emergence in such a setting.

Our conceptual framework is as follows: several user agents act upon self-interest, while a governing agent requests the user agents to take a costly action. There is no force in the governing agent's request. If the user agents cooperate with the governing agent, then the governing agent acknowledges them by giving them a 'responsible user' label. The governing agent can choose what action it shall request users to do—an action that is easy for all users but useless for the governing agent, or an action that is difficult for users and desirable for the governing agent. In the former case, the governing agent gives free 'responsible user' labels to all users. In the latter case, the user agents estimate the benefit of having the label. They do so by considering if the label makes them unique in their group, and if being unique in owning the label has any value. Such value is zero at first and increases with the exposure of the group to the label over time. Ultimately, user agents compare their estimated benefit of gaining the label with their own perception of the cost of the action they are asked to do. This way, they decide if they will cooperate with the governing agent. These actions and interactions occur in each time step. The definition of our conceptual framework was inspired by a work of Bone and Dragićević [42], wherein user agents are logging companies in a forest, and in each time-step they consider cooperating with a conservationist agent, though with different interactions and algorithms from our model.

In terms of the Belief–Desire–Intention (BDI) framework [35], our model's user agents' belief is composed of two parts: the information they have about the last known percentage of the users that participated in the costly behavior, and the information they have concerning the number of responsible user labels awarded since the beginning of the run. The user agents' desire is to have a good reputation while avoiding costly decisions. The user agents' intentions are the decisions they make in response to the governing agent's requests.

Reinforcement Learning (RL) algorithms are a group of Machine Learning algorithms that are based on self-evaluation. RL algorithms do not know the correct answer to the problem at hand, but they can learn to improve themselves from the differences between the results of their own efforts [43]. An RL algorithm has a policy that prescribes an action for each state. In this sense the policy is a function. With each action, there comes a reward and a subsequent state. RL algorithms take note of rewards that are gained from various (state, action) pairs, and update their policies in such a way that the sum of rewards weighted by their time-values is maximized [44]. RL algorithms are suitable for the problem of our study, as our model's governing agent searches for a sequence of decisions to maximize a reward, which in our model is the proportion of user agents that cooperate with the governing agent. Because of their relevance to problems involving repeated decision making, RL algorithms have been used in a variety of simulations of social systems [45–47] as well as social–ecological systems [42,48,49]. In a similar fashion, we used RL algorithms in our model.

Within the above framework, our objectives are to answer the following questions:

1. Can the actions of agents in the above setting result in the emergence of a behavior norm in the user agents, such that the user agents compete for social status and cooperate with the governing agent despite the costly action they are asked to do?

2. How can the governing agent find a sequence of choices that facilitates or hastens the emergence of the above behavior norm?

2. Materials and Methods

To answer the questions of this study we adopted an agent-based simulation approach. First, we built a model of interactions of user agents and the governing agent. In the model, we included algorithms for a governing agent to guide the user agents towards the desired norm of behavior. Next, we performed tests on the model, with and without the governing agent's algorithms. To gain insight about the emergence of the intended norm of behavior, we planned model runs without a purposeful intervention from the governing agent. This allowed us to become familiar with the state of possible outcomes of repeated actions of the user agents. Then, we tested the model with purposeful interventions with a governing agent that was equipped with several algorithms. This allowed us to compare different algorithms against each other and identify algorithms and parameters that lead to the emergence of the desired behavior norms faster than other algorithms and parameters. Finally, we ran the model several times with random interventions by the governing agent to construct a baseline for comparison with the best simulations. In this section we describe the design of the model, the algorithms, and the tests of performance of the simulations.

2.1. Overview, Design Principles, Details

The model description follows the ODD (Overview, Design concepts, Details) protocol [50–52], which serves as a standard for communication of information about Agent-Based Models. In addition, the model description is inspired by the ODD + D protocol [53], which is an adaptation of the ODD protocol for describing human decisions in Agent-Based Models.

2.1.1. Purpose

This ABM is an abstract model of interactions of entities and emergence of a particular social behavior among them. Using this model, we intend to, firstly, obtain an insight into the emergence of social behavior that is costly for individuals, and secondly, examine if such emergence can be facilitated with appropriate learning algorithms.

2.1.2. Entities, State Variables, and Scales

Entities of this model are three classes of agents: several user agents, a governing agent, and a registrar agent. State variables of user agents are named *threshold* and *decision*. Each user agent's *threshold* is a real number between 0 and 1, which is predefined at the beginning of each simulation, remains constant throughout the simulation, and is visible to that user agent alone. User agents' *decisions* are binary variables that change throughout the simulation and are visible to all agent classes upon request. State variables of the governing agent are named *signal*, *state*, *Q*, and *policy*, which change throughout the simulation. Except for *signal*, all other variables of the governing agent are known to itself only. *Signal* is a binary variable. *State* is a two-dimensional variable with non-negative integer values. *Q* is a table with a real value for each *state* and *signal* combination. *Policy* is a table with a real number between 0 and 1 for each *state*. The registrar's state variables are named *nLast* and *nSum*, which are non-zero integers that change throughout the simulation and are visible to all agent classes upon request. User agents and the governing agent are the main entities of the model. Registrar is an auxiliary agent that is meant to make the model easier to understand and serves as a mediator of information. This abstract model does not have a spatial dimension, and time in the model is measured with dimensionless time steps.

2.1.3. Process Overview and Scheduling

Each simulation run consists of a number of episodes. Each episode consists of a number of time-steps. Time in this ABM is modeled as discrete time-steps. In each time-step, the agents act as described in Figure 1.

Governing agent:
Ask **Registrar** to report *nLast* and *nSum*
Produce *signal*
Ask **Registrar** to run a step

Registrar (step() function):
Ask **Governing agent** to report *signal*
If *signal* is 0 then {
give promotional responsible agent labels to all **User agents**
}
else {
ask **User agents** to report their *decisions*
count the number of **User agents** whose *decision* is 1
}
Update *nLast* and *nSum*

User agent:
Ask **Registrar** to report *nLast* and *nSum*
Produce *decision*

Figure 1. Interactions between three classes of agents in one time-step. Class names are highlighted. Variable names are shown in italics.

At the beginning of each new episode of time-steps, the variables *nLast* and *nSum* are set to zero, and user agents forget their memory.

2.1.4. Design Concepts

Basic Principles

In this ABM the governing agent does not enforce its authority over user agents. Rather, it offers them 'responsible user' labels in return for cooperation with the governing agent. User agents see improved social status as the benefit of being recognized as a 'responsible user'. The basis for this idea is the assumption that individuals have a motivation for better social status, and that they may take actions that cost them money if their peers and neighbors do so [54–57].

Emergence

Decisions of user agents are made individually. Emergence of a pattern of such decisions that is costly to the individuals will be an unexpected phenomenon.

Adaptation

The governing agent adapts its Q based on results of each step, and accordingly calculates a new *policy* for its actions.

Objectives

The objective of the governing agent is to increase the ratio of cooperating users when it makes signals of 1. The target cooperation ratio is 0.5 in this model. The governing agent aims to reach a state with target cooperation ratio as soon as possible. The reward for the target state is defined as 0, and the reward for all other states is defined as unity minus the ratio of cooperating users. Therefore, the governing agent's reward at each step is between -1 and 0. User agents react to their perceived conditions by comparing the benefit of the said cooperation against their *thresholds*, which represent the cost to each user of cooperation with the governing agent.

Learning

The governing agent learns to adjust its behavior based on responses that it observes in the user agents. To this end, the governing agent uses RL algorithms. For each RL algorithm there is a separate model. In the RL algorithms, the governing agent stores the value of each action taken at each *state* in its table, Q. From Q it extracts *policy*. *Policy* recommends an action at each *state*. Actions of the governing agent are the *signals* it produces. In the next time-step, based on the outcome of its action, which is the observed ratio of cooperation of user agents, the governing agent updates its Q and repeats this loop.

Prediction

In each time-step, the governing agent predicts the present value of the sequence of future rewards of the actions that it may take. This prediction is made based on the results of previous time-steps, and it is stored in Q. As such, Q is the basis for both learning and prediction in the governing agent.

Sensing

The governing agent and user agents read *nLast* and *nSum* from the registrar. The registrar reads *signal* from the governing agent and *decisions* from user agents.

Interaction

The interaction of the governing agent with user agents is through *signal*. If the governing agent produces a *signal* of 0, it is asking for a task that has no cost to the users, hence giving 'responsible user' labels to all users at no cost. If it produces a *signal* of 1, it is asking the users to cooperate in a task that is costly to them. In this case, the response of each user is its *decision*. If the user produces a *decision* of 0, it is not cooperating with the governing agent. If the user produces a *decision* of 1, it is cooperating with the governing agent despite the costly demand of the governing agent.

Stochasticity

The governing agent produces its *signals* using its *policy*, which is stochastic. In the governing agent's *policy*, the probability of recommendation of each action is the ratio of its estimated value to the sum of estimated values of all possible actions. In addition, *thresholds* of user agents are defined at the beginning of the simulation as random numbers with given mean and standard deviation.

Collectives

Individual *decisions* of each user agent affect future *decisions* of itself and other user agents. Other than that, there is no connection between the user agents.

Observation

Throughout each episode, the governing agent stores in its temporary memory the sequence of rewards that it receives in each time-step. At the beginning of the next episode, this part of its temporary memory is erased. At the end of the final episode of each run, the sequence of rewards is stored in a file as output. The reason for choosing the final episode is that as learning happens throughout the simulation, the governing agent's performance improves in each episode. Therefore, the final episode represents the outcome of learning in the model.

Heterogeneity

User agents are heterogenous in their decision thresholds.

Individual Decision Making

All agents make decisions in the model. In each iteration, the object of decision of the governing agent is to choose between (i) requesting a costly behavior from user agents, and

(ii) requesting an easy behavior from user agents. The governing agent gives recognition labels to cooperating user agents. In iterations where the governing agent requests the easy behavior, all user agents unconditionally cooperate with the governing agent and receive the 'responsible user' labels. In iterations where the governing agent requests the costly behavior, the object of decision of the user agents is to choose between accepting and rejecting the governing agent's request. The objective of the governing agent is to encourage at least half of user agents to perform the costly behavior. Decisions of the governing agent affect decisions of user agents and vice versa. Moreover, decisions of each user agent affect future decisions of itself and other user agents. Within the same time-step, the decision of one user agent does not affect decisions of other user agents. The governing agent's decision policy is probabilistic, and in each state recommends an action. The governing agent is equipped with RL algorithms. User agents do not have learning or optimization capabilities. Instead, the basis of decision making of each user agent is a simple if-statement. User agents calculate the utility of the 'responsible user' label by considering (i) the *uniqueness* that the label will give them, and (ii) the *value* of the label in their agent society. They calculate *uniqueness* based on last known cooperation ratio of user agents with the governing agent; and they calculate *value* based on the number of times the 'responsible user' label has been presented in their agent society since the beginning of the run. When some user agents begin performing the costly action, that behavior might become a norm. This emerging norm influences future decisions of user agents. User agents value being recognized with a 'responsible user' label. There are no social sanctions or other punishments for user agents who do not follow the emerging norm.

2.1.5. Initialization

At the beginning of each run, the registrar's *nLast* and *nSum* are zero. Additionally, the governing agent's *Q* table is filled with random values.

2.1.6. Input Data

The model does not use input data to represent time-varying processes.

2.1.7. Submodels

In RL algorithms, in order to assess policies and find a pathway to improving them, a function is used that allocates a value to each (state, action) pair. In RL literature this function is known as Q [58,59]. In turn, Q is used to update the policy. Different RL algorithms are distinguished in their timing and method of updating Q. We used a class of RL algorithms known as Temporal Difference (TD) learning algorithms. In TD algorithms, learning occurs at each time step. That is, with every action that is taken, its reward and its subsequent state are used to update Q and policy, so that the next action is prescribed with improved knowledge of the behavior of the system [59]. Below, we describe six different TD algorithms which we used. Each of these algorithms defined a submodel in our work. In these descriptions we assume that in state S, the algorithm's policy p prescribes action A. Taking this action results in reward R and subsequent state S′. The next action will be A′. In all cases, the present value of a future earning is calculated using a future discounting rate, γ. Moreover, a learning rate, α, is applied to the correction term before updating Q. All the descriptions and formulas are from Sutton and Barto [59]. For flowcharts of these algorithms, see the Data Availability section.

SARSA

In SARSA, the next action A′ is identified as p(S′). Then, assuming that Q leads the system from pair (S, A) to (S′, A′), the previous assessment of Q is corrected. This correction accounts for the reward R as well as the value of the future pair (S′, A′). Future discounting rate γ is used to calculate the present value of that future pair. Equation (1) summarizes this description:

$$Q(S, A) = Q(S, A) + \alpha\left[R + \gamma \times Q(S', A') - Q(S, A)\right] \quad (1)$$

Q-Learning

Another TD algorithm, known as Q-Learning, looks at Q after taking action A and identifying subsequent state S′. Then, among all pairs (S′, a) that are registered in Q for the new state S′ and all possible actions, the algorithm selects the one with the maximum value, and uses it to correct Q. These operations are summarized in Equation (2):

$$Q(S, A) = Q(S, A) + \alpha\left[R + \gamma \times \max_a\{Q(S', a)\} - Q(S, A)\right] \quad (2)$$

Expected SARSA

Another TD algorithm, known as Expected SARSA, proceeds similar to Q-Learning up to the correction of Q. In that stage, Expected SARSA considers all (S′, a) pairs and calculates their average value. Equation (3) describes this update process:

$$Q(S, A) = Q(S, A) + \alpha[R + \gamma \times \sum_a p(a|S') \times Q(S', a) - Q(S, A)] \quad (3)$$

where the summation is performed over all actions a.

Double Learning Methods

Corresponding to the above three methods, there are more complicated methods that are called Double Learning algorithms, because they involve two Q tables. In each time-step, one of the Q tables is selected randomly and updated using the other one. The following formulas describe this concept. In each set, only one of the two formulas is performed in each time-step. The formulas for update of Q tables of the Double SARSA, Double Q-Learning, and Double Expected SARSA algorithms are as shown in Equation pairs (4) and (5), (6) and (7), (8) and (9), respectively.

Double SARSA:

$$Q_1(S, A) = Q_1(S, A) + \alpha\left[R + \gamma \times Q_2(S', A') - Q_1(S, A)\right] \quad (4)$$

$$Q_2(S, A) = Q_2(S, A) + \alpha\left[R + \gamma \times Q_1(S', A') - Q_2(S, A)\right] \quad (5)$$

Double Q-Learning:

$$Q_1(S, A) = Q_1(S, A) + \alpha\left[R + \gamma \times \max_a\{Q_2(S', a)\} - Q_1(S, A)\right] \quad (6)$$

$$Q_2(S, A) = Q_2(S, A) + \alpha\left[R + \gamma \times \max_a\{Q_1(S', a)\} - Q_2(S, A)\right] \quad (7)$$

Double Expected SARSA:

$$Q_1(S, A) = Q_1(S, A) + \alpha[R + \gamma \times \sum_a p(a|S') \times Q_2(S', a) - Q_1(S, A)] \quad (8)$$

$$Q_2(S, A) = Q_2(S, A) + \alpha[R + \gamma \times \sum_a p(a|S') \times Q_1(S', a) - Q_2(S, A)] \quad (9)$$

2.2. Model Parameters

The model includes several parameters, which we have divided into two groups: those that are parameters of the problem, and those that are parameters of the algorithm. Parameters of the problem are the number of agents (n), mean (μ) and standard deviation (σ) of the decision thresholds of the population from which user agents are selected, and future discounting rate (γ). Parameters of the algorithm are the rate of exploration vs. exploitation (ε) and learning rate (α). Appendix A lists these parameters and their values in the simulations. These parameters and values produce 810 different combinations for each of the 6 algorithms. Therefore, a total of 4860 distinct problem and solution/algorithm settings are possible. We produced simulations for each of these settings. In the analysis of the results, we separated the two parameter groups, taking note of 54 combinations of

problem parameters and 15 combinations of ε and α for each of the 6 algorithms, which produced 90 combinations of solution/algorithm parameters.

Among the problem parameters, n, μ and σ are used in the making of user agents. There are 18 possible combinations of values of these parameters. For each of those combinations, we made 50 sets of user agents. Each set was defined by selecting n thresholds from a normally distributed population with mean μ and standard deviation σ. These sets of thresholds were saved and used in all simulations that shared their respective values of n, μ and σ.

In addition to the above parameters, the model has some parameters that we did not vary in simulations. Specifically, the number of training episodes for each run, which was set to 4000; the number of time steps per episode, which was set to 17; and the number of levels of the two-dimensional state variable, which was set to $\lceil n/2 \rceil$, or integer ceiling of half of user agents, for each dimension. The rational for this choice was to enable the governing agent to distinguish states with different levels of cooperating user agents. The target state is when the number of cooperating user agents reaches or exceeds $n/2$.

2.3. Simulation Experiments

2.3.1. Quantifying Model Performance

In each simulation and for each set of parameters, the model trained itself in 4000 episodes. In the final episode of each run, after long sequences of updates and improvements, the model policy and Q were at their best. Therefore, results of the final episode of each run were used to assess that run. We used two measures to quantify model performance. In most of our analyses we calculated the mean of the rewards of the time steps of an episode as the score of that episode. Our rationale for this choice was that it is a measure of cumulative rewards. In another part of our analyses we used the rewards of the final time step as the score of the episode. The rationale for this choice was that it shows the state of the system at the end of the simulation and allows us to answer questions such as to what extent the desired state was achieved.

2.3.2. Space of Outputs

The user agents in our model react to the *signals* produced by the governing agent. In order to better understand the process of the study, we produced the space of all possible outputs of the model, by producing all possible sequences of decisions of the governing agent and feeding those sequences to the user agents. This way, we constructed a large binary tree of all binary strings of length 16. The length of these strings is one less than the length of the episode because in episodes of length 17 the algorithm makes 16 decisions. We kept the size of episodes to this level because for larger episode lengths, the scope of outputs would become exponentially larger and more difficult to manage, from the point of view of computation. Taking note of the score of each of the produced 2^{16} chains, we obtained insight about the space of outputs, which allowed us to realize which scores are rare and what conditions favor higher scores.

2.3.3. Comparison of Simulations with Each Other

The combination of 54 problem parameters and 90 algorithm/solution parameters produced 4860 unique combinations of parameters and algorithms to run the model in. We ran the model 50 times in each of these combinations of parameters and algorithms. We then took note of scores of runs as the mean reward per time-step of the 50 simulations in each run. This produced a dataset that we organized as a matrix with 54 columns and 90 rows. Based on recorded scores, rows and columns of the scores' matrix were analyzed and ordered with hierarchical clustering. We then identified the ranks of the values within each column of the matrix. The ranked matrix showed a comparison of the 90 algorithm/solution parameters against one another. Both matrices were plotted as heatmaps. Using these visualizations enabled us to find groups of simulations with higher scores. This visual finding was confirmed by marginal sums of the matrices. Through this

process we were able to select algorithms that performed better than the others in most of the problem parameter settings.

2.3.4. Comparison of Simulations with a Reference Baseline

In addition to comparing algorithms with each other, we compared the selected algorithms against a baseline. The reason for this comparison was to note what would happen without the algorithms, and so assess the role of the RL algorithms in the achievement of results. To make this basis for comparison, we ran another series of simulations with the same problem settings and the same thresholds for user agents but without RL algorithms for the governing agent. Instead, in these simulations the governing agent produced *signals* randomly in each time-step. We then compared the results of this new model, which we call the baseline, with the selected RL algorithms.

2.4. Implementation

The model was developed and run using Java Repast Simphony 2.7 [60]. Simulation results were analyzed and visualized using R statistical software [61] and its packages ggplot2 [62], scales [63], and signs [64]. For model code and results, see the Data Availability section.

3. Results

3.1. Simulations without RL

Figure 2 shows histograms obtained by simulating the actions of various sets of user agents given all possible sequences of signals by the governing agent. In these simulations, no RL algorithm was used for the governing agent. Rather, all binary sequences of length 16 were generated and tried on the user agents. As such, the results of these simulations depict the space of outcomes of all possible policies. Each sequence of decisions was tried on 100 sets of user agents with similar characteristics. Therefore, each histogram shows the distribution of scores of 6,553,600 episodes. The score of each episode is calculated as the mean reward per time-step of that episode.

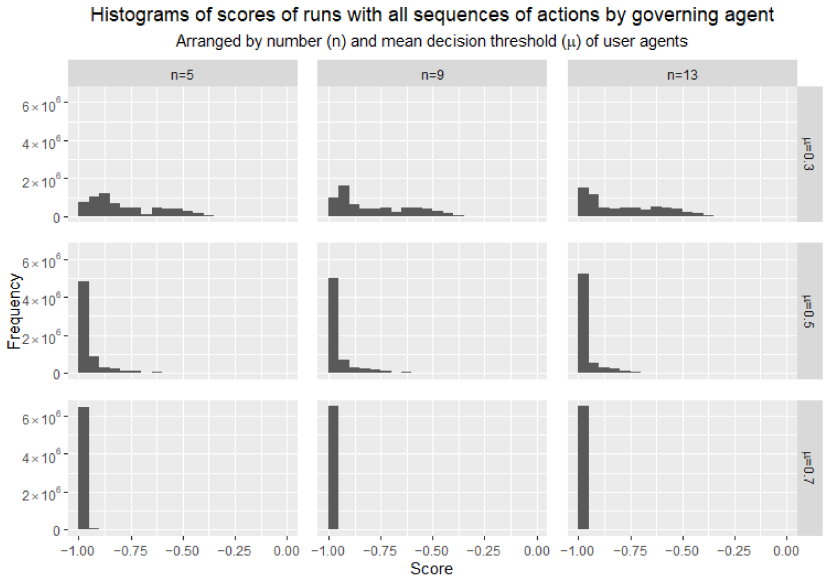

Figure 2. Histograms of overall scores of episodes in reference dataset, ordered by number and mean decision threshold of user agents. Each histogram summarizes 6,553,600 data points. Each data point is an episode of 17 time-steps. The score of each episode is its mean reward per time-step. Rewards are real numbers between −1 and 0, and they are calculated based on the ratio of user agents that cooperate with the governing agent in each time-step.

As shown in Figure 2, the emergence of the desired norm of behavior is highly dependent on the mean cost–benefit decision threshold of the user agents. In simple terms, the more costly the behavior, the less likely it is to become a trend in the society. This is especially evident in the simulations with a mean decision threshold of 0.7 for the user agents, where nearly all episodes ended with the minimum score, and cooperation of the user agents with the governing agent was a rarity. To a lesser extent, this happened in the simulations with a mean decision threshold of 0.5 as well. The histograms of these runs show lower peaks and more dispersed distributions of scores, though their modes are still at the minimum score. On the other hand, in the simulations with a mean decision threshold of 0.3, the scores are distributed more evenly. In two of the three histograms of these simulations, the mode is not at the minimum score. In fact, these histograms show that the number of user agents has an inverse effect on the mode of scores.

3.2. Simulations with RL

Figure 3 shows a heatmap of scores of 4860 sets of simulations. This heatmap is composed of 54 columns and 90 rows. The columns and rows of this figure correspond to problem parameters and algorithm/solution parameters, respectively. Specifically, each column is for one unique combination of number of user agents, mean and standard deviation of decision threshold of the population of user agents, as well as the future discounting rate. Each row is for a unique combination of the RL algorithm, its rate of exploration vs. exploitation, and its learning rate. As such, each column represents a problem, and each row is a solution to that problem. Appendix B includes parameter combinations and their respective codes, which are assigned to columns and rows of the heatmap figures. Each pixel in the heatmap represents the mean score of 50 simulations with the same problem parameters and algorithm/solution parameters. The score of each simulation is the mean reward per time-step of that simulation. Column numbers and row numbers are printed on the margins of the heatmap. The order of columns and rows was determined through hierarchical clustering, using column sums and row sums, respectively. Dendrograms of the hierarchical clustering of columns and of rows are shown in the figure. Larger copies of these dendrograms as well as lists of parameters of rows and columns are available in the Data Availability section.

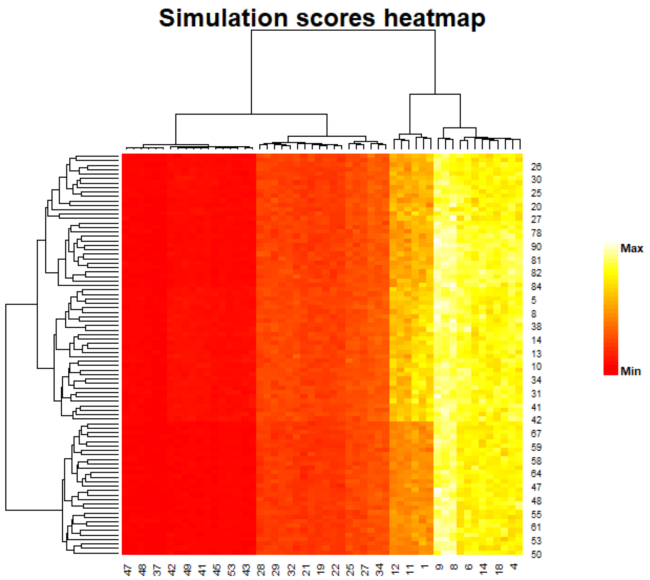

Figure 3. Heatmap of mean scores of simulations. Each row is a unique combination of algorithm settings. Each column is a unique combination of problem parameters. Rows and columns are ordered using hierarchical clustering, as shown in their respective dendrograms. Each pixel represents the mean score of 50 simulations with its respective row and column settings. Simulation scores are mean rewards per time-step.

There are three distinct vertical bands in the heatmap of Figure 3. These correspond to the three tested values for mean decision thresholds of user agents. The left-most vertical band, shown in deep red, corresponds to mean decision threshold of 0.7. The middle band, which shows a variety of red and orange colors, corresponds to mean decision threshold of 0.5, and the right-most band, with the highest variety of colors from orange to white, corresponds to mean threshold of 0.3. The dendrogram of the columns shows that the scores of the two bands on the left are more similar to each other, while the scores of the right-most band are in a different cluster, which is confirmed by the colors of the heatmap.

The colors of the heatmap of Figure 3 are proportional to the values of the pixels of the heatmap, with the lowest value colored red, and the highest value colored white. The visualization in this figure shows that problem parameters have a strong influence in the results. However, our goal is to identify the best solutions to the problems, and from this figure it seems that the differences between the results of various solutions are smaller than the differences between problems. As such, it is not easy to distinguish between different solutions in this figure.

In order to compare different solutions, we prepared Figure 4. This figure is the result of column-ranking of the heatmap of Figure 3. As such, the values in each column of the heatmap of Figure 4 range from 1 to 90, with 1 corresponding to the lowest score and 90 to the highest score in the respective column in the heatmap of Figure 3. In this way, the difference between problem settings is eliminated from Figure 4 and it is only the difference between row values, that is, algorithm/solution parameters, that causes variations in this heatmap. Each of the 54 columns is a test problem. For each test problem, 90 solutions are given, and they are ranked according to their scores. An ideal solution is one that has high ranks in all or most of the tests. In order to more easily understand the figure, the orders of rows and columns are the same as those of Figure 3.

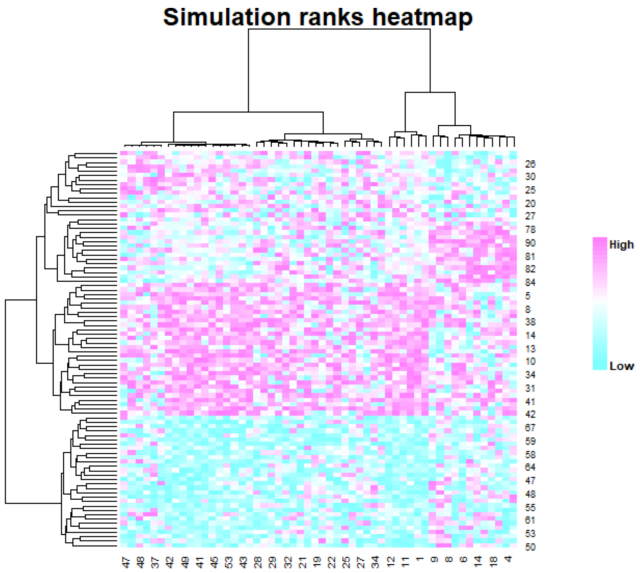

Figure 4. Heatmap of ranks of simulations. Each row is a unique combination of algorithm settings. Each column is a unique combination of problem parameters. For each column, the row with the highest mean score of simulations is given the highest rank. The order of rows and columns in the ranks heatmap is the same as that of the mean scores heatmap.

It can be seen that the heatmap of Figure 4 is divided into three horizontal zones, with the lowest zone having the lowest ranks, and the middle zone having the highest ranks. The lower zone, with lowest ranks, corresponds to the RL algorithms Q-Learning and Expected SARSA. The middle zone, with the highest ranks, corresponds to the RL algorithms Double SARSA and Double Expected SARSA. The upper zone of the figure, which contains solutions with middle ranks, corresponds to the RL algorithms SARSA and

Double Q-Learning. As shown in the dendrograms and confirmed by the colors of the pixels, the scores of the two upper zones are more similar to each other, whereas the lowest zone is in a different cluster.

It is noticeable that within the upper zone of the figure, there is an accumulation of magenta pixels in the bottom-right and in the top-left. We mentioned that the left side of the figure corresponds to problems with mean user agent decision threshold of 0.3, and the right side corresponds to the mean decision threshold of 0.7. We also noted that the latter is a tougher challenge for the RL algorithms because in its space of decisions, rewards are rare. It may seem reasonable to assume that the solutions with higher ranks in the tougher problems are more successful than others. The row numbers of the two groups of solutions show that in this zone, the Double Q-Learning algorithm performs better than the SARSA algorithm.

In all, in Figure 4 the Double Learning algorithms showed superior performance. We looked at the row sums of the heatmap in order to identify the best algorithms with their parameters. The highest-ranking algorithms in the 54 problems were: (1) Double Expected SARSA, with an exploration rate of 0.1 and a learning rate of 0.2; (2) Double Expected SARSA, with an exploration rate of 0.2 and a learning rate of 1.0; and (3) Double SARSA, with an exploration rate of 0.01 and a learning rate of 1.0.

Figure 5 shows the spread of the rewards obtained in various simulations with the RL algorithms Double SARSA and Double Expected SARSA. Each curve in this figure represents the mean rewards of 50 simulations with similar parameters. For each algorithm, 810 different parameter settings were tested. As seen in the figure, the two algorithms show similar variations in results. There are no areas of the plot that are particularly filled with curves of only one of the two algorithms. In this sense, we cannot visually distinguish between the two algorithms. Appendix C includes flowcharts of these two algorithms.

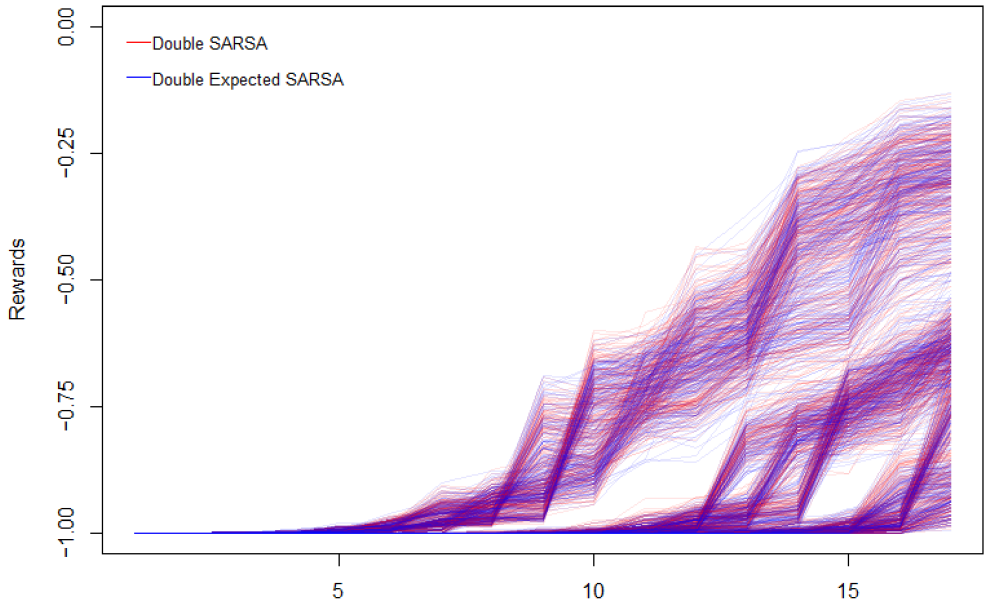

Figure 5. Rewards versus time-steps for Double SARSA and Double Expected SARSA algorithms. For each of the two algorithms, 810 curves are shown. Each curve represents a parameter setting for its respective algorithm. For each parameter setting, 50 simulations were run, their mean score was plotted at each time-step, and a line segment was drawn between the score points of consecutive time steps to produce a curve.

Three strands of curves are visible in the plots of rewards of simulations in time steps. These correspond to the three thresholds for decisions of user agents: the lower the thresholds, the higher the rewards. It is noticeable that in simulations with the mean decision threshold of 0.3, higher rewards emerge between the 5th and 10th time steps. Such

time of emergence of higher rewards is delayed to between the 10th and 15th time steps in simulations with mean decision threshold of 0.5. The rewards of simulations with mean decision threshold of 0.7 emerge later, after the 15th time step. This shows that as the users agents' decision threshold increases, it takes longer times for the RL algorithms to cause the user agents to cooperate with the governing agent.

In Figure 6 we compared the scores of the selected RL algorithms against a baseline. The scores in these histograms are rewards of the 17th time step of 40,500 episodes for each of the RL algorithms and the baseline. In the baseline run, *signals* were produced by the random decisions of the governing agent, and they were given to the user agents. This represents a case where the governing agent does not have an algorithm. Therefore, this case serves as a basis for comparison against the cases where the governing agent does have an algorithm. Recall that the rewards were defined as unity minus cooperation ratio if cooperation ratio is below 0.5, and zero otherwise. The histograms below show this matter, as they include no rewards between -0.5 and 0. The histograms show two peaks of frequencies at the highest and lowest ends of score range. Clearly, in comparison with the baseline, the RL algorithms have lower frequencies of low scores and higher frequencies of high scores.

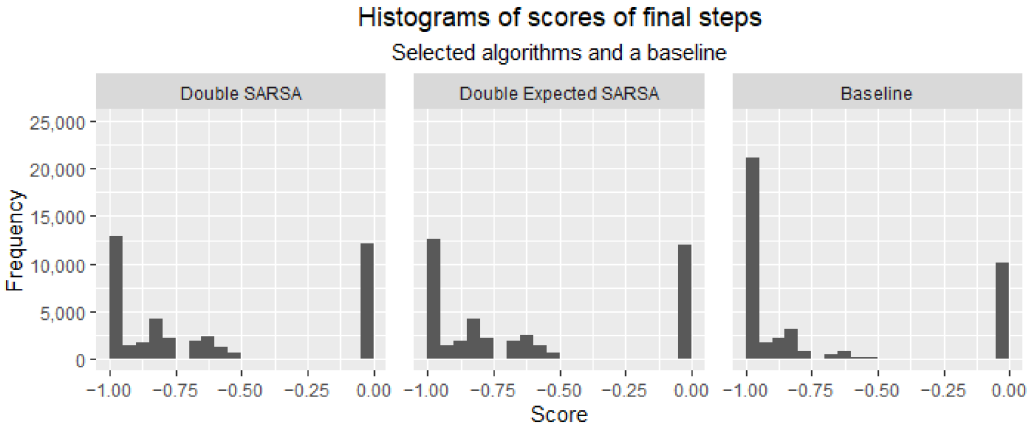

Figure 6. Histograms of scores at the final time step for RL algorithms Double SARSA and Double Expected SARSA as well as a random baseline. Each histogram summarizes 40,500 data points.

The choice of rewards of the 17th step as the score in Figure 6 was in order to show what happens to the group of user agents at the end of the simulation. It indicates to what extent the target state was achieved throughout simulations. The histograms show that compared to the baseline, the RL algorithms were more successful in encouraging the cooperation of the user agents with the governing agent.

Figure 7 shows another comparison of the selected algorithms with the baseline. This figure uses the same simulation and baseline scores as Figure 6, but it separates data according to the mean value of user agents' decision threshold. The boxplots of Figure 7 show the spread of rewards gained at the final time-step of the runs, for three values of the mean threshold (μ).

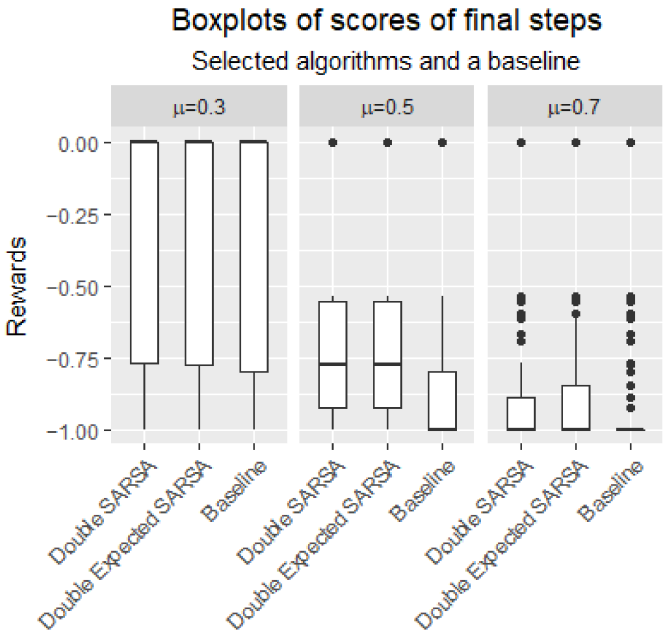

Figure 7. Boxplots of scores at the final time step for RL algorithms Double SARSA and Double Expected SARSA as well as a random baseline, classified by mean value of user agents' decision threshold. Each boxplot shows quartile ranges of scores of 13,500 data points.

Figure 7 reveals several points. Firstly, it is evident in the figure that the results are dependent on the mean decision threshold of the user agents. Secondly, it is noteworthy that at the lower threshold value ($\mu = 0.3$) half of the runs with RL algorithms, as well as the random baseline, reach the target state and obtain full reward at the final time-step. Lastly, at other threshold values ($\mu = 0.5$ and $\mu = 0.7$) the RL algorithms scored distinctly higher than the baseline. The observation that the median of the baseline dataset is the highest possible score when $\mu = 0.3$ indicates that at the lower threshold, the existence of the mechanism of recognition of 'responsible users' leads to emergence of a norm of behavior in which the user agents cooperate with the governing agent. Conversely, the observation that the median of the baseline dataset is the lowest possible score when $\mu = 0.5$ and $\mu = 0.7$ shows that in these cases it is a challenge for the RL algorithms to find a sequence of decisions for the governing agent to create the desired norm of behavior among user agents. These cases show the superior performance of the RL algorithms in comparison with the baseline.

4. Discussion

We started our work with a curiosity about the ability of the governing agent to use reputation as a mechanism for guiding user agents to perform the desired behavior. To that end, we explored the space of possible outcomes of interactions of user agents that consider their reputation. We also equipped the governing agent with a learning algorithm to find a successful policy for its actions. In our abstract study, we consider a policy successful if it leads to the participation of user agents in the desired behavior, such that the percentage of participating users is higher than what could be achieved by random actions. Our criterion for success was inspired by the definition of descriptive norms by Cialdini et al. [6], which inform us of what others do in the society. This is also in accordance with an interpretation of norms mentioned by Therborn [65], in which norms inform us about the distribution of things (in our case: behaviors) and indicate what is 'normal'.

Our simulation of the space of all possible decisions of the governing agent (Figure 2) showed that the emergence of the desired norm of behavior among user agents is possible, though it can be rare, depending on the parameters of the problem. We chose the length of the simulation episodes considering computation hardware limits. Substantial

computational power and memory are required in the processing and internal verification of this stage, as the space of decisions grows exponentially with the number of time steps. Nevertheless, through the simulations we were able to identify information about the process being studied.

Moreover, we showed that RL algorithms could hasten and facilitate the emergence of norms of group behavior that are costly to the individuals. In particular, comparison of the results of RL algorithms with the baseline (Figure 7) showed that in some cases, the algorithms were able to reach results that were rare in their problem settings. On the other hand, the comparison with baseline also showed that in problem settings where the chance of the emergence of the desired behavior is high, a random baseline could reach results comparable with RL algorithms. As such, we can say that if the user agents perceive a low cost for the requested behavior, then having in place a structure in which user agents who performed that behavior are recognized and introduced to the group as 'responsible users' can lead to the diffusion of that behavior in the group and emergence of a new behavior norm. If, on the other hand, the perceived cost of performing that behavior is high for the user agents, then the mere existence of the recognition structure is not enough for diffusion of that behavior in the group. In these cases, a governing agent equipped with an appropriate algorithm may be able to guide the group of user agents towards the desired behavior.

Our initial inspiration for this study comes from our field of work—sustainability—where governments are interested in encouraging individuals to adopt environmentally responsible behavior [66]. The research presented in this paper is part of a larger project aimed at understanding the complexities of a system that is composed of social and ecological parts. Such social-ecological systems involve interactions of subsystems that are, in turn, complex [67,68]. In the present study, we were able to select algorithms to use in the construction of a social-ecological model in future. We also identified parameters to use for those algorithms.

In abstraction, our model's governing agent aims to encourage our model's user agents to do something that the user agents perceive is costly for them. This is as if the governing agent was trying to sell something—in our model's case, a 'responsible user' label—to the user agents, where the user agents are not initially convinced that it is worth the price. A field of study that deals with similar problems is marketing. In fact, ABMs are applied in marketing research and are known to be useful because of their cross-scale capabilities: they build individual agents and capture results that emerge in the scale of the society [69]. A similar point has been mentioned in the literature of innovation diffusion [18,70]. In addition, it has been noted that an individual's decision to purchase a product depends on the quality of the product and the social influence the individual receives from their peers [70]. Similarly, our user agents are influenced by their society. Our model's user agents each perform a cost–benefit analysis. They assess the benefit of performing a task that is costly for them. Such a benefit is social respect. Then, the agents compare that benefit with a threshold, which represents their perception of the cost of the task. Our model's agents, however, do not receive a product in return for the cost that they pay. As such, they compare the cost only with their estimate of the value of the social status that they may gain if they pay the price for it. In a related work, Antinyan et al. [71] built an ABM in which each agent compares its status with the mean status of others in their social network, and decides accordingly to spend a budget to improve its own status. In a similar fashion, our user agents consider the mean status of their group in their decision to take a costly action for improving their own status. In a different study, Shafiei et al. [72] built an ABM of market share of electric vehicles and stated that visibility of a new subject can help it become a trend in the society. Similarly, our user agents consider a measure of visibility of the new trend in their group, and the governing agent's actions increase visibility of the 'responsible user' label. In another ABM study about promotional activities in marketing and sales of products, Delre et al. [73] concluded that timing of promotional activities has an important role in the success of a sales campaign, and inappropriate timing

may cause the sales of the product to fail. In our study, decisions of the governing agent are indeed about giving free promotional 'responsible user' labels to all user agents. The RL algorithms give the governing agent a policy that prescribes when promotional labels should be given for free. The comparison of the performance of the RL algorithms with the random baseline showed that the timing of promotional offers of the label, which the algorithms prescribed, was influential in achieving results.

Our model is an abstract model that simulates interaction of agents in a hypothetical setting. There is always a concern about such abstract models and whether they are useful, as they are not connected to the real world. Moreover, without connection to the real world, questions arise about the validity of the model and the relevance of its results. Below, we address these issues.

Depending on the model's purpose, ABMs can be classified in two different types: predictive and explanatory [74]. The aim of predictive models is to extrapolate trends, evaluate scenarios and predict future states, whereas the aim of explanatory models, in terms of Castle and Crooks, is 'to explore theory and generate hypotheses' [74]. These different purposes justify different approaches. Predictive models try to be detailed enough to make a precise enough replicate of the real world, while explanatory models often involve simplifying assumptions that reduce the real world to abstractions [74,75]. There have been many cases where abstract models have led to better understanding of phenomena and theories. For example, Adam Smith developed a theory in which markets emerge as the result of actions of individuals pursuing their own interest [76]. Centuries later, Gavin analyzed an abstract ABM based on Smith's work and put Smith's theory to test with it, to find whether self-interest actions of individuals will result in increased utility overall [77]. For another example, Axelrod developed an abstract ABM of hypothetical agents interacting with each other in a game of repeated Prisoner's Dilemma [78] and based on that abstract model he made substantial contributions to the theory of cooperation. Another example is Schelling's segregation model [79] in which he simulated spatial patterns of distribution of ethnic groups in a hypothetical urban environment. Our model, too, is abstract and aims to provide insights about emergence of certain behaviors in groups of agents. Our model is not intended to represent a real-world system. Rather, it is meant to show whether it is possible that behaviors that are costly to individuals emerge and become a norm in a group, given a mechanism of recognition of agents who perform such a behavior.

In addition to verifying our model at several stages of development, we compared several algorithms with each other in our model assessment effort (Figure 4). These comparisons shed light on the simulations and allowed us to distinguish more powerful algorithms and identify some sets of parameters with which the algorithms perform well in various tests. We also assessed our algorithms against a baseline (Figures 6 and 7) and showed that the identified algorithms make a difference in comparison to a case where those algorithms are not used. Moreover, by constructing the space of outputs of all possible sequences of actions (Figure 2) we gained an insight into the results that can be reached, and the rarity of our desired state. Through this integrated approach we put our hypothetical ABM to test, verified it, and learned about its power and its limits.

In two literature reviews, Savarimuthu and Cranefield [8] and Hollander and Wu [12] noted that many authors associate norms with social sanctions and enforcement. Axelrod [80] states that in simulations, sanctions facilitate the emergence of norms because an agent's calculation of its utility is affected by the negative score of the sanctions that it might face, if it does not follow the norm. Therefore, it seems that in a setting without sanctions, norms are less likely to emerge than in a similar setting with sanctions. Our study involved a setting without sanctions, and the desired behavior still emerged among the user agents. This indicates two points: firstly, the offer of good reputation is a mechanism that contributes to the emergence of a new norm, even in the absence of sanctions; and secondly, the governing agent's RL algorithm allows it to effectively use the reputation mechanism and promote the desired behavior. These points address a question that Hollander and Wu [12] raise in their literature review, about possible alternatives to social sanctions.

Our governing agent performs the role of centralized leadership [37] in the emergence of a new norm in its society. The new norm is the manifestation of decisions of user agents to cooperate with the governing agent. These decisions are dependent on the user agents' thresholds for assessment of the utility of their choices. In principle, if the governing agent knew the mean value of decision thresholds of user agents, then the governing agent could adjust its actions accordingly and have an efficient policy. The governing agent could do this by repeatedly giving promotional labels to all user agents and increasing their utility, until their utility reached their decision thresholds. Then, the governing agent could ask for the costly behavior, and the user agents would find that the utility of being recognized as a responsible user is worth more than the cost of the requested behavior, so they would cooperate with the governing agent. This is in accordance with Axelrod's [80] explanation of how the desire for good reputation can lead to the emergence of a norm. However, the challenge for our governing agent is that it does not know the decision thresholds of user agents. To better understand this challenge, suppose that the governing agent underestimates the mean decision threshold of user agents. In this case, before giving sufficient promotional labels and increasing the utility of the desired behavior in the user agents, the governing agent asks for the costly behavior. As a result, the unprepared user agents do not cooperate with the governing agent. As another result, the promotional activity of the governing agent is delayed by its untimely request, and the emergence of the norm will be postponed. Now suppose another case, where the governing agent overestimates the mean decision threshold of user agents. In this case, the governing agent continues increasing the utility of the desired behavior in the user agents, without realizing when they are ready to cooperate with the governing agent. As such, the governing agent loses time in unnecessary promotional activity and does not ask user agents to perform the desired behavior. This case, as well, results in postponed emergence of the norm. The success of the governing agent, therefore, depends on the proper timing of its activities. Our governing agent's RL algorithm adjusted itself by occasionally making costly requests and checking the response of the user agents. In this way, the RL algorithm achieved higher user agent cooperation rates than a random baseline, as evident in Figure 7.

The scope of this study is the evolution of the simulated society of the governing and user agents, from a situation where no user agents cooperate with the governing agent till a situation where the desired proportion of user agents voluntarily cooperate with the governing agent and perform the costly behavior that the governing agent requests. As such, our work addresses another question raised by Hollander and Wu [12] regarding the early stages of norm creation and ideation. What happens afterwards is beyond the scope of this study. Nevertheless, it is worth noting as an implication that when many user agents voluntarily perform the behavior that the governing agent requests, the society of user agents may develop a tendency to take that behavior for granted and sanction those who do not participate in that behavior [80]. As another implication, the governing agent may introduce new laws to enforce the newly emerged norm [8]. These can be subjects of future works.

As another idea for future work, our model can be used in the study of complex systems that involve our case of Principal–Agent setting in conjunction with another phenomenon. For example, in environmental management there is typically a governing agent or entity with demands from users of an environmental resource. Such social interactions can be simulated in our model. The environmental resource, in turn, is subject to the laws of nature. If there exists a model of natural changes in the environmental resource, then by coupling that model with the model described in the present study, it is possible to simulate the changes in that social-ecological system.

5. Conclusions

In this study we developed an abstract ABM of the interactions of a principal and several agents in a hypothetical context where the principal offers the agents recognition and good reputation in return for their cooperation in a behavior that is costly to the agents.

Our simulation results showed that in such a setting, the emergence of the desired behavior as a norm is possible. If the agents perceive that the cost of the behavior is low, then emergence of that norm is possible even without guidance of the agents by the principal. If the perceived cost of the behavior is not low, then cases of emergence of that norm are rare. However, we demonstrated through comparison with a random baseline that RL algorithms can effectively guide the agents towards adopting the said behavior. Among the six TD RL algorithms that we tried—namely, SARSA, Q-Learning, Expected SARSA, Double SARSA, Double Q-Learning, and Double Expected SARSA—we noted that Double Learning algorithms obtained better results in the setting of this study. We conclude that with a proper learning algorithm it is possible to create norms of costly behaviors, by using recognition as a reward for participation in the behavior, even in the absence of social sanction and enforcement.

Author Contributions: Conceptualization, S.H., L.P. and R.M.-H.; methodology, S.H., L.P. and R.M.-H.; software, S.H.; validation, S.H.; formal analysis, S.H., L.P. and R.M.-H.; investigation, S.H., L.P. and R.M.-H.; resources, L.P.; data curation, S.H.; writing—original draft preparation, S.H., L.P. and R.M.-H.; writing—review and editing, S.H., L.P. and R.M.-H.; visualization, S.H.; supervision, L.P. and R.M.-H.; project administration, L.P.; funding acquisition, L.P. and R.M.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada through the Discovery Grant number RGPIN/05396-2016 awarded to L.P., R.M.-H. received financial support from the European Union's Seventh Framework Programme through NEWFOREST programme number PIRSES-GA-2013-612645.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Code of the model is available at https://github.com/s-harati/model-Cooperation (accessed on 23 July 2021). Datasets of results of simulations as well as flowcharts of the RL algorithms used are available at https://osf.io/jyqu7/ (accessed on 23 July 2021).

Acknowledgments: We are thankful to two anonymous reviewers who read the manuscript and provided helpful feedback towards its improvement. We are also thankful to the Université de Montréal's International Affairs Office (IAO) for their financial support through the International Partnership Development program, which allowed the collaboration between researchers from UdeM and CREAF.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Model Parameters

Table A1. Model parameters.

Parameter Group	Parameter	Symbol	Values
Problem parameters	Number of user agents	n	5, 9, 13
	User agents mean threshold	μ	0.3, 0.5, 0.7
	User agents' threshold standard deviation	σ	0.06, 0.08
	Future discounting rate	γ	0.1, 0.5, 0.9
Algorithm parameters	Exploration rate	ε	0.01, 0.1, 0.2
	Learning rate	α	0.2, 0.4, 0.6, 0.8, 1

Appendix B. Parameter Combination Codes

Table A2. Codes and combinations of problem parameters. These codes correspond to column numbers of the heatmap figures in the text. μ and σ are the mean and standard deviation of User agent thresholds, respectively. n is the number of user agents. γ is the future discount rate.

Code	μ	σ	n	γ	Code	μ	σ	n	γ	Code	μ	σ	n	γ
1	0.3	0.06	5	0.1	19	0.5	0.06	5	0.1	37	0.7	0.06	5	0.1
2	0.3	0.06	5	0.5	20	0.5	0.06	5	0.5	38	0.7	0.06	5	0.5
3	0.3	0.06	5	0.9	21	0.5	0.06	5	0.9	39	0.7	0.06	5	0.9
4	0.3	0.06	9	0.1	22	0.5	0.06	9	0.1	40	0.7	0.06	9	0.1
5	0.3	0.06	9	0.5	23	0.5	0.06	9	0.5	41	0.7	0.06	9	0.5
6	0.3	0.06	9	0.9	24	0.5	0.06	9	0.9	42	0.7	0.06	9	0.9
7	0.3	0.06	13	0.1	25	0.5	0.06	13	0.1	43	0.7	0.06	13	0.1
8	0.3	0.06	13	0.5	26	0.5	0.06	13	0.5	44	0.7	0.06	13	0.5
9	0.3	0.06	13	0.9	27	0.5	0.06	13	0.9	45	0.7	0.06	13	0.9
10	0.3	0.08	5	0.1	28	0.5	0.08	5	0.1	46	0.7	0.08	5	0.1
11	0.3	0.08	5	0.5	29	0.5	0.08	5	0.5	47	0.7	0.08	5	0.5
12	0.3	0.08	5	0.9	30	0.5	0.08	5	0.9	48	0.7	0.08	5	0.9
13	0.3	0.08	9	0.1	31	0.5	0.08	9	0.1	49	0.7	0.08	9	0.1
14	0.3	0.08	9	0.5	32	0.5	0.08	9	0.5	50	0.7	0.08	9	0.5
15	0.3	0.08	9	0.9	33	0.5	0.08	9	0.9	51	0.7	0.08	9	0.9
16	0.3	0.08	13	0.1	34	0.5	0.08	13	0.1	52	0.7	0.08	13	0.1
17	0.3	0.08	13	0.5	35	0.5	0.08	13	0.5	53	0.7	0.08	13	0.5
18	0.3	0.08	13	0.9	36	0.5	0.08	13	0.9	54	0.7	0.08	13	0.9

Table A3. Codes and combinations of algorithm settings. These codes correspond to row numbers of the heatmap figures in the text. Algorithms are Double Expected SARSA (DXS), Double Q-Learning (DQ), Double SARSA (DS), Expected SARSA (ES), Q-Learning (Q), and SARSA (S). ε is the exploration rate. α is the learning rate.

Code	Algorithm	ε	α	Code	Algorithm	ε	α	Code	Algorithm	ε	α
1	DXS	0.01	0.2	31	DS	0.01	0.2	61	Q	0.01	0.2
2	DXS	0.01	0.4	32	DS	0.01	0.4	62	Q	0.01	0.4
3	DXS	0.01	0.6	33	DS	0.01	0.6	63	Q	0.01	0.6
4	DXS	0.01	0.8	34	DS	0.01	0.8	64	Q	0.01	0.8
5	DXS	0.01	1.0	35	DS	0.01	1.0	65	Q	0.01	1.0
6	DXS	0.10	0.2	36	DS	0.10	0.2	66	Q	0.10	0.2
7	DXS	0.10	0.4	37	DS	0.10	0.4	67	Q	0.10	0.4
8	DXS	0.10	0.6	38	DS	0.10	0.6	68	Q	0.10	0.6
9	DXS	0.10	0.8	39	DS	0.10	0.8	69	Q	0.10	0.8
10	DXS	0.10	1.0	40	DS	0.10	1.0	70	Q	0.10	1.0
11	DXS	0.20	0.2	41	DS	0.20	0.2	71	Q	0.20	0.2
12	DXS	0.20	0.4	42	DS	0.20	0.4	72	Q	0.20	0.4
13	DXS	0.20	0.6	43	DS	0.20	0.6	73	Q	0.20	0.6
14	DXS	0.20	0.8	44	DS	0.20	0.8	74	Q	0.20	0.8
15	DXS	0.20	1.0	45	DS	0.20	1.0	75	Q	0.20	1.0
16	DQ	0.01	0.2	46	ES	0.01	0.2	76	S	0.01	0.2
17	DQ	0.01	0.4	47	ES	0.01	0.4	77	S	0.01	0.4
18	DQ	0.01	0.6	48	ES	0.01	0.6	78	S	0.01	0.6
19	DQ	0.01	0.8	49	ES	0.01	0.8	79	S	0.01	0.8
20	DQ	0.01	1.0	50	ES	0.01	1.0	80	S	0.01	1.0
21	DQ	0.10	0.2	51	ES	0.10	0.2	81	S	0.10	0.2
22	DQ	0.10	0.4	52	ES	0.10	0.4	82	S	0.10	0.4
23	DQ	0.10	0.6	53	ES	0.10	0.6	83	S	0.10	0.6
24	DQ	0.10	0.8	54	ES	0.10	0.8	84	S	0.10	0.8
25	DQ	0.10	1.0	55	ES	0.10	1.0	85	S	0.10	1.0
26	DQ	0.20	0.2	56	ES	0.20	0.2	86	S	0.20	0.2
27	DQ	0.20	0.4	57	ES	0.20	0.4	87	S	0.20	0.4
28	DQ	0.20	0.6	58	ES	0.20	0.6	88	S	0.20	0.6
29	DQ	0.20	0.8	59	ES	0.20	0.8	89	S	0.20	0.8
30	DQ	0.20	1.0	60	ES	0.20	1.0	90	S	0.20	1.0

Appendix C. Flowcharts of Selected Algorithms

Among the algorithms used in this study, Double SARSA and Double Expected SARSA performed better than the others. Their flowcharts are shown in Figures A1 and A2, respectively. For larger images of these flowcharts and the flowcharts of other algorithms used in the study, see the Data Availability section.

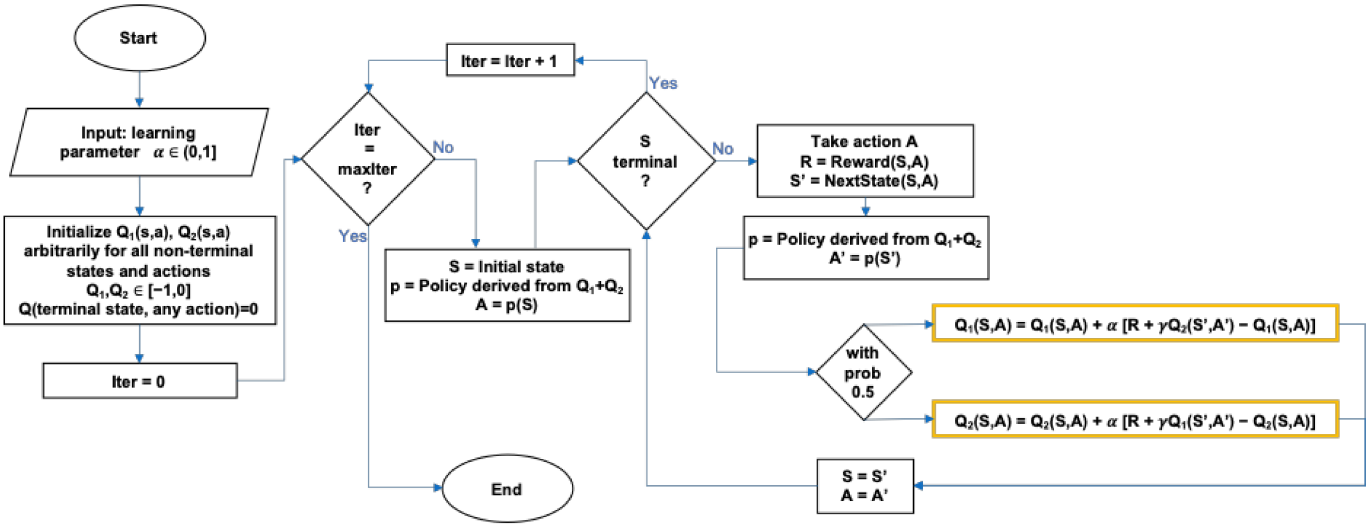

Figure A1. Double SARSA flowchart.

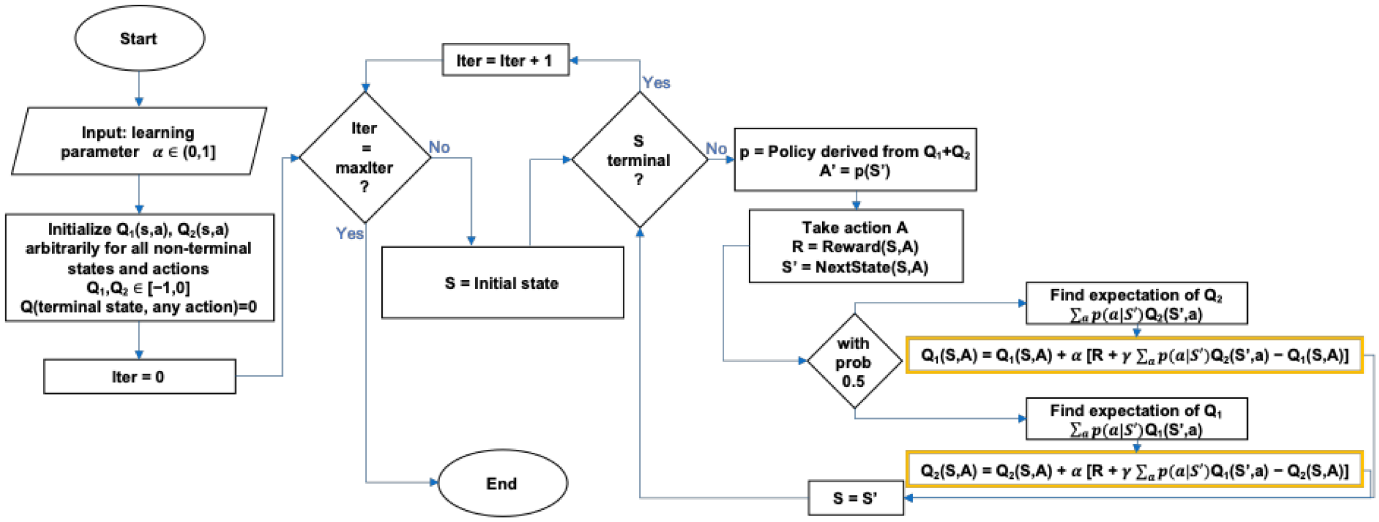

Figure A2. Double Expected SARSA flowchart.

References

1. Wittemyer, G.; Daballen, D.; Douglas-Hamilton, I. Rising ivory prices threaten elephants. *Nature* **2011**, *476*, 282–283. [CrossRef]
2. Feeny, D.; Berkes, F.; McCay, B.J.; Acheson, J.M. The Tragedy of the Commons: Twenty-Two Years Later. *Hum. Ecol.* **1990**, *18*, 1–19. [CrossRef] [PubMed]
3. Blundell, A.G.; Gullison, R.E. Poor regulatory capacity limits the ability of science to influence the management of mahogany. *For. Policy Econ.* **2003**, *5*, 395–405. [CrossRef]
4. Wagner, W. Commons Ignorance: The Failure of Environmental Law to Produce Needed Information on Health and the Environment. *Duke Law J.* **2004**, *53*, 1619–1746.
5. Braun, D.; Guston, D.H. Principal-agent theory and research policy: An introduction. *Sci. Public Policy* **2003**, *30*, 302–308. [CrossRef]
6. Cialdini, R.B.; Reno, R.R.; Kallgren, C.A. A Focus Theory of Normative Conduct: Recycling the Concept of Norms to Reduce Littering in Public Places. *J. Pers. Soc. Psychol.* **1990**, *58*, 1015–1026. [CrossRef]

7. Ross, H.L. *Perspectives on Social Order*; McGraw-Hill: New York, NY, USA, 1973.
8. Savarimuthu, B.T.R.; Cranefield, S. Norm creation, spreading and emergence: A survey of simulation models of norms in multi-agent systems. *Multiagent Grid Syst.* **2011**, *7*, 21–54. [CrossRef]
9. Ostrom, E. *Governing the Commons: The Evolution of Institutions for Collective Action*; Cambridge University Press: Cambridge, UK, 1990.
10. Crawford, S.E.; Ostrom, E. A grammar of institutions. *Am. Polit. Sci. Rev.* **1995**, *89*, 582–600. [CrossRef]
11. North, D.C. *Institutions, Institutional Change and Economic Performance*; Cambridge University Press: Cambridge, UK, 1990; ISBN 9780521397346.
12. Hollander, C.D.; Wu, A.S. The Current State of Normative Agent-Based Systems. *J. Artif. Soc. Soc. Simul.* **2011**, *14*. [CrossRef]
13. Batty, M.; Torrens, P.M. Modelling and prediction in a complex world. *Futures* **2005**, *37*, 745–766. [CrossRef]
14. Goldstein, J. Emergence as a Construct: History and Issues. *Emergence* **1999**, *1*, 49–72. [CrossRef]
15. Manson, S.M. Simplifying complexity: A review of complexity theory. *Geoforum* **2001**, *32*, 405–414. [CrossRef]
16. An, L. Modeling human decisions in coupled human and natural systems: Review of agent-based models. *Ecol. Modell.* **2012**, *229*, 25–36. [CrossRef]
17. Crooks, A.; Castle, C.; Batty, M. Key challenges in agent-based modelling for geo-spatial simulation. *Comput. Environ. Urban Syst.* **2008**, *32*, 417–430. [CrossRef]
18. Kiesling, E.; Günther, M.; Stummer, C.; Wakolbinger, L.M. Agent-based simulation of innovation diffusion: A review. *Cent. Eur. J. Oper. Res.* **2012**, *20*, 183–230. [CrossRef]
19. Axelrod, R. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*; Princeton University Press: Princeton, NJ, USA, 1997; Volume 3.
20. Sanchez-Anguix, V.; Tunalı, O.; Aydoğan, R.; Julian, V. Can Social Agents Efficiently Perform in Automated Negotiation? *Appl. Sci.* **2021**, *11*, 6022. [CrossRef]
21. Amato, F.; Moscato, F.; Moscato, V.; Pascale, F.; Picariello, A. An agent-based approach for recommending cultural tours. *Pattern Recognit. Lett.* **2020**, *131*, 341–347. [CrossRef]
22. Arnoux Hebert, G.; Perez, L.; Harati, S. An Agent-Based Model to Identify Migration Pathways of Refugees: The Case of Syria. In *Agent-Based Models and Complexity Science in the Age of Geospatial Big Data, Advances in Geographic Information Science*; Perez, L., Kim, E.-K., Sengupta, R., Eds.; Springer International Publishing: Basel, Switzerland, 2018; pp. 45–58. ISBN 978-3-319-65992-3.
23. Anderson, T.; Leung, A.; Perez, L.; Dragićević, S. Investigating the Effects of Panethnicity in Geospatial Models of Segregation. *Appl. Spat. Anal. Policy* **2021**, *14*, 273–295. [CrossRef]
24. Anderson, T.; Leung, A.; Dragicevic, S.; Perez, L. Modeling the geospatial dynamics of residential segregation in three Canadian cities: An agent-based approach. *Trans. GIS* **2021**, *25*, 948–967. [CrossRef]
25. Perez, L.; Dragicevic, S.; Gaudreau, J. A geospatial agent-based model of the spatial urban dynamics of immigrant population: A study of the island of Montreal, Canada. *PLoS ONE* **2019**, *14*, e0219188. [CrossRef] [PubMed]
26. Perez, L.; Dragicevic, S. An agent-based approach for modeling dynamics of contagious disease spread. *Int. J. Health Geogr.* **2009**, *8*, 50. [CrossRef] [PubMed]
27. Perez, L.; Dragicevic, S. Modeling mountain pine beetle infestation with an agent-based approach at two spatial scales. *Environ. Model. Softw.* **2010**, *25*, 223–236. [CrossRef]
28. Gaudreau, J.; Perez, L.; Harati, S. Towards Modelling Future Trends of Quebec's Boreal Birds' Species Distribution under Climate Change. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 335. [CrossRef]
29. Li, X.; Liu, X. Embedding sustainable development strategies in agent-based models for use as a planning tool. *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 21–45. [CrossRef]
30. Maggi, E.; Vallino, E. Price-based and motivation-based policies for sustainable urban commuting: An agent-based model. *Res. Transp. Bus. Manag.* **2021**, *39*, 100588. [CrossRef]
31. Yazan, D.M.; Fraccascia, L. Sustainable operations of industrial symbiosis: An enterprise input-output model integrated by agent-based simulation. *Int. J. Prod. Res.* **2020**, *58*, 392–414. [CrossRef]
32. Bristow, M.; Fang, L.; Hipel, K.W. Agent-Based Modeling of Competitive and Cooperative Behavior Under Conflict. *IEEE Trans. Syst. Man, Cybern. Syst.* **2014**, *44*, 834–850. [CrossRef]
33. Afshar Sedigh, A.H.; Purvis, M.K.; Savarimuthu, B.T.R.; Frantz, C.K.; Purvis, M.A. Impact of Different Belief Facets on Agents' Decision–A Refined Cognitive Architecture to Model the Interaction Between Organisations' Institutional Characteristics and Agents' Behaviour. In *Proceedings of the Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*; Aler Tubella, A., Cranefield, S., Frantz, C., Meneguzzi, F., Vasconcelos, W., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 133–155.
34. Afshar Sedigh, A.H.; Purvis, M.K.; Savarimuthu, B.T.R.; Purvis, M.A.; Frantz, C.K. Impact of Meta-roles on the Evolution of Organisational Institutions. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*; Springer: Cham, Switzerland, 2021; pp. 66–80.
35. Rao, A.; Georgeff, M. BDI Agents: From Theory to Practice. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, CA, USA, 12–14 June 1995.
36. Bratman, M. *Intention, Plans, and Practical Reason*; Harvard University Press: Cambridge, MA, USA, 1987.
37. Boman, M. Norms in artificial decision making. *Artif. Intell. Law* **1999**, *7*, 17–35. [CrossRef]

38. Lindström, B.; Jangard, S.; Selbing, I.; Olsson, A. The role of a "common is moral" heuristic in the stability and change of moral norms. *J. Exp. Psychol. Gen.* **2018**, *147*, 228–242. [CrossRef]
39. Shoham, Y.; Tennenholtz, M. Emergent conventions in multi-agent systems: Initial experimental results and observations. In Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning (KR), San Mateo, CA, USA, 25–29 October 1992; Morgan Kaufmann: Burlington, MA, USA; pp. 225–231.
40. Andrighetto, G.; Campennì, M.; Cecconi, F.; Conte, R. The Complex Loop of Norm Emergence: A Simulation Model. In *Simulating Interacting Agents and Social Phenomena*; Springer: Japan, Tokyo, 2010; pp. 19–35.
41. Hales, D. Group reputation supports beneficent norms. *J. Artif. Soc. Soc. Simul.* **2002**, *5*, 1–4.
42. Bone, C.; Dragićević, S. Simulation and validation of a reinforcement learning agent-based model for multi-stakeholder forest management. *Comput. Environ. Urban Syst.* **2010**, *34*, 162–174. [CrossRef]
43. Alpaydin, E. Reinforcement Learning. In *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2014; pp. 517–545.
44. Canese, L.; Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; Spanò, S. Multi-Agent Reinforcement Learning: A Review of Challenges and Applications. *Appl. Sci.* **2021**, *11*, 4948. [CrossRef]
45. Angourakis, A.; Santos, J.I.; Galán, J.M.; Balbo, A.L. Food for all: An agent-based model to explore the emergence and implications of cooperation for food storage. *Environ. Archaeol.* **2015**, *20*, 349–363. [CrossRef]
46. Okdinawati, L.; Simatupang, T.M.; Sunitiyoso, Y. Multi-agent Reinforcement Learning for Collaborative Transportation Management (CTM). In *Agent-Based Approaches in Economics and Social Complex Systems IX*; Springer: Singapore, 2017; pp. 123–136.
47. Chan, C.K.; Steiglitz, K. *An Agent-Based Model of a Minimal Economy*; Princeton University: Princeton, NJ, USA, 2008.
48. Rasch, S.; Heckelei, T.; Oomen, R.; Naumann, C. Cooperation and collapse in a communal livestock production SES model – A case from South Africa. *Environ. Model. Softw.* **2016**, *75*, 402–413. [CrossRef]
49. Bohensky, E. Learning Dilemmas in a Social-Ecological System: An Agent-Based Modeling Exploration. *J. Artif. Soc. Soc. Simul.* **2014**, *17*, 1–2. [CrossRef]
50. Grimm, V.; Berger, U.; DeAngelis, D.L.; Polhill, J.G.; Giske, J.; Railsback, S.F. The ODD protocol: A review and first update. *Ecol. Modell.* **2010**, *221*, 2760–2768. [CrossRef]
51. Grimm, V.; Berger, U.; Bastiansen, F.; Eliassen, S.; Ginot, V.; Giske, J.; Goss-Custard, J.; Grand, T.; Heinz, S.K.; Huse, G.; et al. A standard protocol for describing individual-based and agent-based models. *Ecol. Modell.* **2006**, *198*, 115–126. [CrossRef]
52. Grimm, V.; Railsback, S.F.; Vincenot, C.E.; Berger, U.; Gallagher, C.; DeAngelis, D.L.; Edmonds, B.; Ge, J.; Giske, J.; Groeneveld, J.; et al. The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. *J. Artif. Soc. Soc. Simul.* **2020**, *23*, 1–7. [CrossRef]
53. Müller, B.; Bohn, F.; Dreßler, G.; Groeneveld, J.; Klassert, C.; Martin, R.; Schlüter, M.; Schulze, J.; Weise, H.; Schwarz, N. Describing human decisions in agent-based models – ODD + D, an extension of the ODD protocol. *Environ. Model. Softw.* **2013**, *48*, 37–48. [CrossRef]
54. Anderson, C.; Hildreth, J.A.D.; Howland, L. Is the desire for status a fundamental human motive? A review of the empirical literature. *Psychol. Bull.* **2015**, *141*, 574–601. [CrossRef]
55. Tascioglu, M.; Eastman, J.K.; Iyer, R. The impact of the motivation for status on consumers' perceptions of retailer sustainability: The moderating impact of collectivism and materialism. *J. Consum. Mark.* **2017**, *34*, 292–305. [CrossRef]
56. Nolan, J.M.; Schultz, P.W.; Cialdini, R.B.; Goldstein, N.J.; Griskevicius, V. Normative Social Influence is Underdetected. *Personal. Soc. Psychol. Bull.* **2008**, *34*, 913–923. [CrossRef] [PubMed]
57. Lazaric, N.; Le Guel, F.; Belin, J.; Oltra, V.; Lavaud, S.; Douai, A. Determinants of sustainable consumption in France: The importance of social influence and environmental values. *J. Evol. Econ.* **2020**, *30*, 1337–1366. [CrossRef]
58. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement Learning: A Survey. *J. Artif. Intell. Res.* **1996**, 237–285. [CrossRef]
59. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018; ISBN 9780262039246.
60. North, M.J.; Collier, N.T.; Ozik, J.; Tatara, E.R.; Macal, C.M.; Bragen, M.; Sydelko, P. Complex adaptive systems modeling with Repast Simphony. *Complex Adapt. Syst. Model.* **2013**, *1*, 3. [CrossRef]
61. R Core Team. *R: A Language and Environment for Statistical Computing*; R Core Team: Vienna, Austria, 2019.
62. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*; Springer: New York, NY, USA, 2016; ISBN 978-3-319-24-277-4.
63. Wickham, H.; Seidel, D. *R Package "Scales": Scale Functions for Visualization*; R Core Team: Vienna, Austria, 2020.
64. Wolfe, B.E. *R Package "Signs": Insert Proper Minus Signs*; R Core Team: Vienna, Austria, 2020.
65. Therborn, G. Back to Norms! on the Scope and Dynamics of Norms and Normative Action. *Curr. Sociol.* **2002**, *50*, 863–880. [CrossRef]
66. Barr, S. Strategies for sustainability: Citizens and responsible environmental behaviour. *Area* **2003**, *35*, 227–240. [CrossRef]
67. Liu, J.; Dietz, T.; Carpenter, S.R.; Alberti, M.; Folke, C.; Moran, E.; Pell, A.N.; Deadman, P.; Kratz, T.; Lubchenco, J.; et al. Complexity of Coupled Human and Natural Systems. *Science* **2007**, *317*, 1513–1516. [CrossRef]
68. Ostrom, E. A General Framework for Analyzing Sustainability of Social-Ecological Systems. *Science* **2009**, *325*, 419–422. [CrossRef]
69. Rand, W.; Rust, R.T. Agent-based modeling in marketing: Guidelines for rigor. *Int. J. Res. Mark.* **2011**, *28*, 181–193. [CrossRef]
70. Delre, S.A.; Jager, W.; Bijmolt, T.H.A.; Janssen, M.A. Will it spread or not? the effects of social influences and network topology on innovation diffusion. *J. Prod. Innov. Manag.* **2010**, *27*, 267–282. [CrossRef]

71. Antinyan, A.; Horváth, G.; Jia, M. Social status competition and the impact of income inequality in evolving social networks: An agent-based model. *J. Behav. Exp. Econ.* **2019**, *79*, 53–69. [CrossRef]
72. Shafiei, E.; Thorkelsson, H.; Ásgeirsson, E.I.; Davidsdottir, B.; Raberto, M.; Stefansson, H. An agent-based modeling approach to predict the evolution of market share of electric vehicles: A case study from Iceland. *Technol. Forecast. Soc. Change* **2012**, *79*, 1638–1653. [CrossRef]
73. Delre, S.A.; Jager, W.; Bijmolt, T.H.A.; Janssen, M.A. Targeting and timing promotional activities: An agent-based model for the takeoff of new products. *J. Bus. Res.* **2007**, *60*, 826–835. [CrossRef]
74. Castle, C.; Crooks, A. *Principles and Concepts of Agent-Based Modelling for Developing Geospatial Simulations*; Centre for Advanced Spatial Analysis (UCL): London, UK, 2006.
75. Livet, P.; Phan, D.; Sanders, L. Diversité et complémentarité des modèles multi-agents en sciences sociales. *Rev. Fr. Sociol.* **2014**, *55*, 689. [CrossRef]
76. Smith, A. *An Inquiry Into the Nature and Causes of the Wealth of Nations*; Campbell, R.H., Skinner, A.S., Eds.; Liberty Fund: Indianapolis, IN, USA. (first published 1776); 1982; ISBN 978-0-86597-008-3.
77. Gavin, M. An agent-based computational approach to "the Adam Smith problem". *Hist. Soc. Res.* **2018**, *43*, 308–336. [CrossRef]
78. Axelrod, R. The Emergence of Cooperation among Egoists. *Am. Polit. Sci. Rev.* **1981**, *75*, 306–318. [CrossRef]
79. Schelling, T.C. Dynamic models of segregation. *J. Math. Sociol.* **1971**, *1*, 143–186. [CrossRef]
80. Axelrod, R. An evolutionary approach to norms. *Am. Polit. Sci. Rev.* **1986**, *80*, 1095–1111. [CrossRef]