2023

Deep Reinforcement Machine Learning as a Driver of Agent Decision-Making in Agent-Based Models of Coupled Natural and Human Complex Systems

Kevin Allen Andrew
University of Vermont

Deep Reinforcement Machine Learning as a Driver of Agent Decision-Making in Agent-Based Models of Coupled Natural and Human Complex Systems

A Thesis Presented

by

Kevin Allen Andrew

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Computer Science

August, 2023

Defense Date: July 12th, 2023
Thesis Examination Committee:

Asim Zia, Ph.D., Advisor
Scott Hamshaw, Ph.D., Chairperson
Donna Rizzo, Ph.D.
Safwan Wshah, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

Abstract

Agent-based models are becoming increasingly useful in studying the behavior of real-world complex multi-agent systems; however, one of the outstanding challenges in the modeling of coupled natural and human systems is the dearth of techniques for creating agents that are able to learn from their past failures and successes, as well as compounded environmental and social uncertainties. This research has been focused on the integration of traditional agent-based modeling with machine learning methodologies for modeling agent decision-making and its recursive impacts on economic, environmental, and societal outcomes, feeding into the dynamic co-evolution of the coupled natural and human system state variables within simulated worlds, resulting in the development of two models incorporating and exploring the use of deep reinforcement machine learning as a driver for decision-policy making in agent-based models.

The first of these models is a model of agricultural land use and the adoption of agricultural best-management practices by farmers in response to ecological and economic scenarios as a result of municipal regulation and variance in the occurrence of extreme weather events. The primary study area used for the model is a region of the Missiquoi Bay Area of Lake Champlain in Vermont, containing 480 farmer agents corresponding to agricultural land parcels within the region. A parameter sweep and sensitivity analysis on model hyperparameters was conducted to explore the effects of changes to agent calibration and training on agent decision-making and model performance.

The second model expands upon the scope of the first, including forester agents and commercial and residential urban agents within a larger region of the Lake Champlain Basin of Vermont. Additionally, the impacts of agent decision-making take place on the simulated landscape, resulting in gradual land cover change over time. Land cover data from the United States Geological Survey's National Land Cover Database was used for initial parameterization, calibration, and training of the model (years 2001, 2006) and model testing (year 2011).

Results suggest that with appropriate scoping and hyperparameter selection, the integration of deep reinforcement machine learning techniques into the development of agent-based models can increase predictive accuracy in the modeling of real-world phenomena; however, these gains must be weighed against the increased technical complexity of such a model and the associated risk of introducing model error.

Table of Contents

List of Figures . . . iii
List of Tables . . . v

1 Review of Related Work **1**

2 Machine Learning in Multi-Agent Systems **6**
2.1 Methodology . . . 7
2.1.1 Modeling Approach . . . 7
2.1.2 Agent Decision-Making . . . 8
2.1.3 Agent Learning . . . 9
2.1.4 Agent Memory . . . 10
2.2 Experimental Design . . . 11
2.2.1 Model Overview . . . 11
2.2.2 Experimental Setup . . . 21
2.3 Results . . . 23
2.3.1 Model Performance . . . 23
2.3.2 Agent Behavior . . . 25
2.4 Discussion . . . 25

3 Increasing ABM Integration **28**
3.1 Methodology . . . 28
3.2 Experimental Design . . . 29
3.2.1 Model Overview . . . 29
3.2.2 Experimental Setup . . . 43
3.3 Results . . . 46
3.3.1 Model Performance . . . 46
3.4 Discussion . . . 48

A Farm Model ODD+D Document **53**
A.1 Overview . . . 53
A.2 Design Concepts . . . 58
A.3 Details . . . 61

B Land-Cover Model Design Overview **63**
B.1 Overview . . . 63
B.2 Initialization . . . 68

C Result Listings **71**

List of Figures

2.1 Diagram of (a) the actor-critic network layout and (b) the active-target transfer learning air used by agents 8
2.2 Plot of the number of agents that converged to a stable policy for each parameterization of the model 24
2.3 Sample model output showing the change in BMP adoption likelihood from a characteristic initial model state (a) to a characteristic end states for (b) a model run with the parameterization ($g = 0.05$, $\Delta EE = 0$), and (c) a model run with the parameterization ($g = 0.20$, $\Delta EE = 0.2$). The color of each dot represents the likelihood that the agent will adopt BMPs, with green indicating a high likelihood and red indicating a low likelihood. 24
2.4 Distribution of mean BMP adoption rate for uniform population runs of the agricultural land use model, where $g = 0.0$, for (a) $F = 0$, (b) $F = 0.5$, and (c) $F = 1.0$ 26
2.5 Distribution of mean BMP adoption rate for uniform population runs of the agricultural land use model, where $g = 0.05$, for (a) $F = 0$, (b) $F = 0.5$, and (c) $F = 1.0$ 26
2.6 Distribution of mean BMP adoption rate for uniform population runs of the agricultural land use model, where $g = 0.2$, for (a) $F = 0$, (b) $F = 0.5$, and (c) $F = 1.0$ 26
2.7 Distribution of mean BMP adoption rate for mixed population runs of the agricultural land use model, where $g = 0.0$, $\Delta EE = 0$, for (a) $P = 0.25$, (b) $P = 0.5$, and (c) $P = 0.75$ 27

3.1 A plot of the NLCD data for the selected study area showing land-cover in real-year 2001, each 30m by 30m land cell is colored based on its NLCD cover class (Table 3.1), with similar colors being numerically closer classes. 31
3.2 Diagram showing how land cells and their associated properties are represented within the model as a grid of NLCD land cover values (left), and how those values have land use properties mapped onto them 1-1 (right). 32
3.3 Diagram showing how land-use, and consequently land-cover, may change for an example 2-by-3 parcel over the course of a time-step. . 36

3.4 Flowchart showing a high-level overview of model execution between the main training/calibration loop (left), and the final testing runtime loop (right) 44
3.5 Flowchart demonstrating the overall execution of the agent-based model and its coupling with the machine learning processes 45
3.6 NSE index sensitivity for each index showing the variance in model classification accuracy by each metric under different model parameterizations. 46
3.7 Arrow plots showing a comparison of the categorical transitions between the three main land-cover categories between (a) the real world data, and (b) the "best-fit" model, and (c) the "best median" model. Arrow width between NLCD-11 and 2016 is proportional to the number of transitions between the connected categories. 47

A.1 A map of the study area and the locations of the agricultural agents within it. 54
A.2 A diagram of model execution throughout episodic training. Within each time step of the episode farmer agents make decisions and learn, every five time steps the regulator agent acts, and after 40 model years the episode ends and is reset. 58

List of Tables

2.1 Table of the state properties of agricultural agents and their associated data type for agricultural agents in the agricultural model. 13
2.2 A summary of the actions being select by the agricultural agents in the agricultural model. 14
2.3 Table of the state properties of the regulatory agent and their associated data type for the regulatory agent in the agricultural model. . . 17
2.4 A summary of the action factors being used to drive agent decision-making for both types of agent present in the model. 18
2.5 A summary of the state factors being used as input to the agents' ANNs. 19
2.6 Network parameters for the ANNs used by agents in each class for the land cover model 20
2.7 Hyperparameters and their associated values with source or rationale, if applicable, for the agricultural land-use model. 21
2.8 Table listing experimental parameters for uniform population runs . . 23
2.9 Table listing experimental parameters for mixed population runs . . . 23

3.1 A listing of the 15 NLCD cover classes that are present in the dataset for the study area, their NLCD encoding, and their associated cover category within the model. 31
3.2 Land Cell Features 32
3.3 A summary of the state factors being used during decision-making for agricultural agents in the land-cover model. 33
3.4 A summary of the action factors being used to drive agent decision-making for agricultural agents in the land-cover model. 34
3.5 A summary of the state factors being used during decision-making for forestry agents in the land-cover model. 36
3.6 A summary of the action factors being used to drive agent decision-making for forestry agents in the land-cover model. 37
3.7 A summary of the state factors being used during decision-making for commercial agents in the land-cover model. 38
3.8 A summary of the action factors being used to drive agent decision-making for commercial agents in the land-cover model. 38
3.9 A summary of the state factors being used during decision-making for residential agents in the land-cover model. 39
3.10 A summary of the action factors being used to drive agent decision-making for residential agents in the land-cover model. 39

3.11 Network parameters for the ANNs used by agents in each class for the land cover model 40
3.12 Fixed hyperparameters and their associated values for the land-cover change model. 41
3.13 Experimental parameters that were tested in experimental runs of the land-cover transition model 46
3.14 Confusion matrix comparing the resulting categorical transitions from the NLCD data to the modeled transitions for both the best fit, and best performing parameterization. Because the initial cover category for both transitions are definitionally equal, it has been omitted from the header row. The transition $\Delta_{*,O}$ represents a transition outside one of the three major cover categories. 48

A.1 Table of all state properties of agricultural agents in the farmer model 54
A.2 Table of components of actions that farmer agents can take 55
A.3 Properties of Regulatory Agent 57
A.4 Table of components of actions that the regulatory agent can take in the farm model 57
A.5 Table listing information shared between agents in the farm model . . 60
A.6 Table listing initialization of parameters of the agricultural agents in the farm model 62
A.7 Table listing initialization of parameters of the regulatory agent in the farm model 62

B.1 Table of all properties that are shared amongst all human agents in the land-cover transition model. 64
B.2 Table of all state properties of agricultural agents and their associated data type for agricultural agents in the land-cover transition model. . 65
B.3 Table of state properties of forestry agents and their associated data types in the land-cover transition model. 66
B.4 Table of state properties of commercial agents and their associated data types in the land-cover transition model. 66
B.5 Table of state properties of residential agents and their associated data types in the land-cover transition model. 67
B.7 Table listing initialization of parameters of the agricultural agents in the land-cover transition model 69
B.8 Table listing distribution of initial agent salaries and the weight on their probability. 69
B.8 (continued...) 70

C.1 Mean BMP adoption rate for uniform-population runs of the agricultural model for the parameterizations with results plotted in Figure 2.4, Figure 2.5, and Figure 2.6. 71
C.2 Mean BMP adoption rate for mixed-population runs of the agricultural model for the parameterizations with results plotted in Figure 2.7 for each sub-population: (1) local $F = 0$, (2) local $F = 1$, and (3) mixed neighborhood. 73

Chapter 1

Review of Related Work

The use of agent-based models to study the behavior of human agents in complex systems primarily dates back to the early 1970s, with some of the first formal models being Schelling's dynamic model of segregation [1], Reynolds' distributed herding model [2], and Axelrod and Hamilton's model for the iterative prisoner's dilemma [3]. While attempts to rationalize and describe human behavior date to antiquity, these models were among the first to demonstrate how reducing a complex system down to its elementary components and the simple rules that define it allows for its dynamic behaviors to be reliably, and repeatedly, observed and studied.

The study of emergent systematic behavior and large-scale system dynamics, as described in Anderson's *More is Different* [4], would quickly become known as complex systems studies. Over the following decades, interest in the field grew, and the modeling of multi-agent systems became more widespread, resulting in the development of larger, more complex agent-based models.

While early models primarily focused on studying small homogeneous systems, as work continued through the late 1990s and into the new millennium, researchers

began to model the behavior of more heterogeneous agent populations [5] and explore how agents behave when given cooperative, competitive, or organizational tasks [6]. Work from this period began to focus less on solipsistic agents with information only about their independent state and more on how information sharing and networking can affect agent behavior [7].

The number of ways to define the behavior of agents within complex agent-based systems is myriad; however, some of the most common include probabilistic methods and rule-based approaches. For the majority of this project, the behavior of agents is going to be defined by artificial neural networks trained using deep reinforcement machine learning. Agent-based systems have previously incorporated reinforcement learning methods like SARSA and temporal difference learning; however, this project is one of the first to embed this type of neural network into agents within such a large-scale and heterogeneous model.

This specific application of reinforcement machine learning may be new, but its study is almost as old as the field of modern computer science. One of the first recorded mentions of reinforcement learning techniques for the development of artificial intelligence is in Turing's *Computing Machinery and Intelligence* [8], wherein he proposes that one possible way to construct an intelligent machine is to create a "child machine," that, through the application of punishments and rewards, is taught to behave such that "events which shortly preceded the occurrence of a punishment signal are unlikely to be repeated, whereas a reward signal [increases] the probability of repetition of the events which led up to it." Computational learning of this sort was studied more seriously over the following decade, eventually being dubbed 'reinforcement learning' in Minsky's Steps Towards Artificial Intelligence [9]. While many

of the techniques of this era have been supplanted by newer methodologies, some of its key theoretical concepts became mainstays and went on to form the backbone of modern reinforcement learning— perhaps most notably the development of temporal difference learning as described in Samuel's *Some Studies in Machine Learning Using the Game of Checkers* [10].

Progress in the study of reinforcement machine learning saw little development over the following decade; however, a resurgence of interest in artificial intelligence during the 1970s revitalized the field and resulted in many new algorithms. Some of the more influential of these algorithms being the temporal difference learning algorithm [11], the q-learning algorithm [12], and the related SARSA algorithm [13] for decision-policy making.

Notably, Sutton's temporal difference learning algorithm category $TD(\lambda)$, where the historical discounting factor $0 \leq \lambda \leq 1$, is the basis for many of the techniques used in this project. Dayan proved that Sutton's temporal difference learning algorithm family converges for discrete problem spaces [14]; however, the problem remains undecidable for continuous-valued problems, so consideration must be taken for model hyperparameter selection.

Alongside these developments in reinforcement learning, advancements in computing machinery and the production and training of artificial neural networks helped bypass many of the previous limiting factors in the study of artificial intelligence. For example, the best method to correct neural network output had been an open question since their first use. But, the development of algorithms for the backpropagation of network error revolutionized the field [15]. These methods allowed for the creation of networks that were more intricate and generalizable than ever before, and their

increased performance made them a standard with derivatives still used today.

Entering the mid-to-late 1990s, development in artificial intelligence and reinforcement machine learning again began to stall. Problems like vanishing and exploding gradients within the hidden layers of networks, as well as physical limitations on the size and speed of machine memory, made the use and application of deep, large-scale neural networks infeasible for many potential use cases. Progress in the field remained incremental until the mid-2010s when advancements in GPU-enabled computing allowed for faster, more powerful, and more affordable high-performance computing to enter the mainstream.

With this improvement in computing capabilities came several new reinforcement learning methods, including deep reinforcement learning, which makes use of the ability of artificial neural networks to perform function approximation to make the decision-policy for a problem space. By using deep neural networks in this way, the decision-policy table q-learning algorithms use to value decision-making in discrete problem spaces can be replaced with a neural network with a deep q-network architecture for decision-policy making in more continuous problem spaces. Deep Q-Network learning (DQN) is a suitable algorithm for many reinforcement learning tasks, but it's not without its flaws. Overcoming its propensity towards biasing itself from outlier data early in training can be incredibly difficult. [16]

To combat some of the difficulties that can arise from using DQN, several additions and variations to the algorithm have been developed. The addition of policy gradient [17] and action replay [18] to the algorithm can help to smooth the learning curve and encourage additional exploration of the problem space. Additionally, combination algorithms like double deep q-learning (DDQN) [19] and the rainbow algorithm [20]

have been showing promising results; however, they are still fairly young algorithms and haven't been around long enough to do a proper meta-analysis of their reliability and accuracy across problem types.

Chapter 2

Machine Learning in Multi-Agent Systems

Traditional ABMs often rely on rule-based or probabilistic decision-making strategies, which sometimes fail to capture higher-order decision-making logic, including reasoning from past experience, dynamic decision-making under uncertainty, and understanding of implicit or emergent external reward/incentive structures. In this regard, one of the outstanding challenges in ABMs of coupled natural and human systems concerns the lack of ABMs that can simulate agents with the ability to learn from their past failures or successes and environmental and social uncertainties. [21]

In this chapter, a method of integrating machine learning into ABMs is presented as a potential solution to this problem using a modeling methodology incorporating elements of deep reinforcement machine learning with classical ABM techniques. This methodology is then applied to a simple ABM of a coupled natural and human system. The results of this application are then discussed.

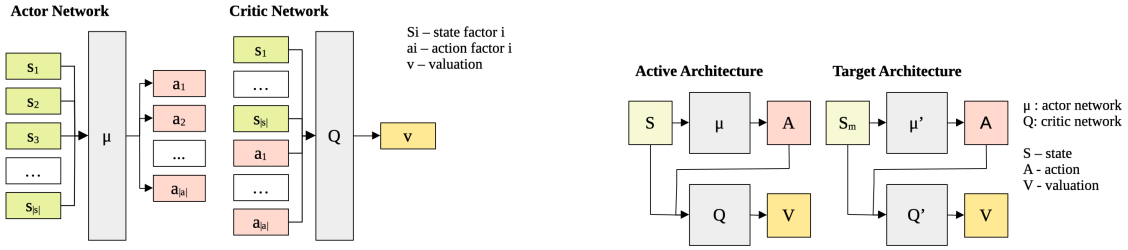
2.1 Methodology

2.1.1 Modeling Approach

The deep reinforcement machine learning techniques that are being incorporated into this modeling methodology are an extension of the deep q-learning methods developed by Hasselt, Guez, and Silver [19], incorporating some of the alterations to action relay and learning convergence as described in the rainbow algorithm developed by Hesel et al. [20] and integrating the episodic training structure into the runtime execution of an agent-based model.

In this regard, each agent has two paired actor-critic neural network architectures — one pair, which is used for 'active' learning, and a 'target' pair which is used for passive learning. The active pair is used to drive agent decision-making within the current simulated model environment in any given time-step. The target pair is updated periodically with the weights of the active pair. This transfer learning is done to prevent the network from overfitting to circumstance and to prevent the networks from diverging during training.

Within each pair, the actor network (μ) is responsible for selecting the next action to take given the current state of the agent and the critic network (Q) is responsible for estimating the value of the current state given the current state and action. The actor network is trained to maximize the value of the critic network, whereas the critic network is trained to minimize the difference between its estimated valuation for each state-action air with the valuation that would be consistent with the rewards received from past events.



(a) Actor-Critic Pair *(b) Active-Target Pair*

Figure 2.1: Diagram of (a) the actor-critic network layout and (b) the active-target transfer learning air used by agents

High-level diagrams of these architectures and how they interact with agent states, $s = \left(s_1, ..., s_{|s|}\right)$, and actions, $a = \left(a_1, ..., a_{|a|}\right)$, as vectorized components to produce value estimations can be seen in Figure 2.1.

2.1.2 Agent Decision-Making

In traditional agent-based models, agents make decisions according to rule-based decision-policies or probabilistic methods. In integrating machine learning, agents instead make decisions according to an internal decision-policy function $\pi(s) = a$ mapping the state of each agent to the potential actions that each agent can take. In this approach, the decision-policy function is being approximated by an artificial neural network (ANN), $\mu : S \rightarrow A$. The input to this ANN is the state of the agent, vectorized as a 1-dimensional array of length W_S. The output of the ANN is a vector of length W_A encoding the action that the ANN has decided the agent should take.

Each time an agent needs to take an action, it passes its current state through the network to generate an action. It performs this action with some probability $1 - \epsilon$. With probability ϵ, the agent will instead take a random action. This random action is used to encourage exploration of the state space and to prevent the agent from

getting stuck in a local optimum.

2.1.3 Agent Learning

The policy evaluation network, Q, is updated using the Q-learning update function (Eq 2.1), where S_t and A_t are the state and selected action at time t, α is the learning rate, and the target Y_t^{DDQN} is defined as a function of the state, action, and received reward value R_t (Eq 2.2).

$$\forall\theta \in \Theta_Q \left[\theta_{t+1} = \theta_t + \alpha \left(Y_t^{DDQN} - Q(S_t, A_t)\right) \nabla_{\theta_t} Q(S_t, A_t)\right] \tag{2.1}$$

$$Y^{DDQN} = R_{t+1} + \gamma Q' \left(S_{t+1}, \text{argmax}_a \left(Q(S_{t+1}, a)\right)\right) \tag{2.2}$$

Because the networks in this system update are training using experience replay, the argmax term present in Equation 2.2 is replaced with the output of the target actor $\mu'(S_{t+1})$ as part of a batch update of batch size B. The actor network μ also updates from the selected experience batch, using policy gradient with regard to the resulting valuations provided by Q.

The target architecture initially has the same weights as the main architecture, at the end of each training episode, weights from the main architecture are copied to the target architecture according to the transfer learning function (Eq 2.3).

$$\forall\theta'_i \in \Theta' \left[\theta'_i \leftarrow \tau\theta_i + (1-\tau)\,\theta'_i, \tau \ll 1\right] \tag{2.3}$$

2.1.4 Agent Memory

Agents in the model store a history of their past experience as a series of state transition records (s_t, a_t, r_t, s_{t+1}). These records are stored in a memory buffer B of fixed length N. When the memory buffer is full, new records overwrite the oldest records in the buffer. The memory buffer is used to train the agent's decision-policy and valuation.

Additionally, agents have a built-in 'forgetfulness factor', F, which has been incorporated in an attempt to capture some of the behavior patterns of human actors with imperfect memory. This factor is a real number between 0 and 1 and is used to linearly scale the amount of noise that is introduced into the memory record as the record ages within a run. An agent with $F = 0$ will have perfect recall of its entire state transition history, whereas an agent with $F = 1$ will have perfect recall of its most recent state transition with actions taken in the distant past being completely forgotten (noise term of equal range as actual term).

for all agents in each time step **do**

With probability ϵ select random action A_t

otherwise select action $A_t \leftarrow \mu(S_t)$

Execute A_t and observe reward R_t and next state S_{t+1}

Push transition record to memory $M[m] \leftarrow (S_t, A_t, R_t, S_{t+1})$

Select random minibatch B of N transitions from memory M

if Agent is forgetful, $F > 0$ **then**

Introduce random noise to transition record proportional to the temporal distance to transition $(m - t)$ and agent forgetfulness F

end if

Perform Q-learning update via action replay

Perform μ update via policy gradient

end for

2.2 Experimental Design

2.2.1 Model Overview

In order to test this modeling methodology for the integration of deep reinforcement machine learning into an agent-based model, an experimental ABM was developed. This ABM is a model of a multi-agent of agricultural decision-makers and how their behavior may change in response to external stimuli. The real-world basis for this model is a study area in the Missisquoi Bay Area of the Lake Champlain Basin of Vermont, and the model is designed to represent the agricultural decision-making processes of farmers in this area — in particular, decisions pertaining to agricultural productivity and the adoption or rejection of agricultural best management practices (BMPs).

Agents

There are two types of agents present in this model — 480 farmer agents, corresponding to the 480 agriculturally-zoned land parcels in the Missisquoi Bay Area, and a single regulatory agent. All agents in the model contain some internal information about their current state and history, a set of state-transition memories used to learn from experience, and a set of neural networks used to drive agent decision-making.

As the agents make decisions over time, they gradually learn the correlation between the actions they take from each state using deep reinforcement machine learning.

The 480 agricultural agents model the behavior of farmers, herders, and other kinds of agricultural land managers within the study area. They make annual decisions about their farming practices, including whether they should change production in one of the four modeled agricultural industries (beef, dairy, corn, and hay) and whether they should implement an agricultural best management practice (BMP) to reduce phosphorus runoff on their land.

The state properties and variables that make up each agricultural agent are listed in Table 2.1. The initialization of these values is detailed in Section A.3.1.

Conditions that factor in as components of a farmer agent's state include the total land area the agent has devoted to cropland or pasture; the productivity of the agent in each of the four modeled agricultural industries along with their associated phosphorus byproduct productivity; an 5-year history of the farm's profitability, storm losses, and BMP usage; and similar historical information from the agent's k-nearest neighboring farmer agents.

This subset of state factors is summarized along with those of the regulatory agent in Table 2.5. For the farmer agents, these break down into a few main groups: information about their own land cover, information about their productivity in the given time-step, a 5-year history of their own experiences, and historical information from their 5-nearest neighbors.

Table 2.1: Table of the state properties of agricultural agents and their associated data type for agricultural agents in the agricultural model.

Name	Description	Data Type
Agent ID	Unique identifier for this agent	`uint`
Agent Status		`enum{3}`
Land Parcel Data		
Crop Land Area A_c	Land devoted to growing crops (sq km)	`float`
Pasture Land Area A_p	Land devoted to grazing animals (sq km)	`float`
Total Land Area A_{tot}	Total land in parcel (sq km)	`float`
Productivity		
Corn p_c	Corn production factor	`float`
Hay p_h	Hay production factor	`float`
Beef p_b	Beef production factor	`float`
Dairy p_d	Dairy production factor	`float`
Phosphorous $p_{p,x}$	Phosphorus production factor	`float`
Cows Owned C	Number of cows on farm	`uint`
Financial History		
Real Net	What was net profit over last 5 years	`float[5]`
Expected Net	What was expected profit for last 5 years	`float[5]`
Extreme Event History	Extreme event presence over past 5 years	`uint[5]`
BMP Usage History B	Did farm use BMP in last 5 years	`bool[5]`
Neighbors	References to neighboring agents	`farmer*[5]`
Neural Networks		
Actor Network μ	Network Weights	`float[l][w]`

Table 2.1: (continued...)

Name	Description	Data Type
Critic Network Q	Network Weights	`float[L][W]`
Target Network μ'	Network Weights	`float[l][w]`
Target Network Q'	Network Weights	`float[L][W]`
Memory Bank		`float[M*R]`
Memory Buffer		`float[B*R]`

The actions that the farmer agents can take are listed in Table 2.2. These actions are divided into two main categories: the action of choosing to adopt or not adopt a BMP for their farm, and adjusting the farm's productivity in one of the four agricultural sectors.

Table 2.2: A summary of the actions being select by the agricultural agents in the agricultural model.

Group	Action
BMP Usage	Adopt BMP
	Don't Adopt BMP
Corn Production	Increase by $[0, S_c^+)$
	Maintain
	Decrease by $[0, S_c^-)$
Hay Production	Increase by $[0, S_h^+)$
	Maintain
	Decrease by $[0, S_h^-)$
Dairy Production	Increase by $[0, S_d^+)$

	Maintain
	Decrease by $[0, S_d^-)$
Beef Production	Increase by $[0, S_b^+)$
	Maintain
	Decrease $[0, S_b^-)$

The production factors are a scalar component of production functions that have been calculated for the region and calibrated for the years 2001–2040. These functions are shown below for the production of corn (Eq 2.4), hay (Eq 2.5), dairy (Eq 2.7), and beef (Eq 2.6), where t is the modeled year.

$$P_c(t) = p_c * A_c^b * 11.433 \log t - 86.826 \tag{2.4}$$

$$P_h(t) = p_h * A_c^b * 1\text{e}{-}32 \exp 0.0358t \tag{2.5}$$

$$P_b(t) = p_b * A_p^b * 2\text{e}{-}20 \exp 0.0234t \tag{2.6}$$

$$P_d(t) = p_d * A_p^b * 2\text{e}{-}9 \exp 0.0114t \tag{2.7}$$

The productivity of the agent is modified by the application of the regulatory agent's regulations G_1 (Eq 2.13) and G_2 (Eq 2.14) and the amount of losses due to extreme weather events (Eq 2.8) as a function of whether the BMP was used and whether the number of extreme events that occurred within the given year exceeds the expected threshold from the weather submodel.

$$
\begin{aligned}
S(B, EE) &= 1, && EE < N \\
S(B, EE) &= 0.1, && EE \geq N, \quad \neg B \\
S(B, EE) &= (0.1 + 0.9B_e), && EE \geq N, \quad B
\end{aligned}
\tag{2.8}
$$

The weather submodel would generate a number of rainfall events for the study area every model year, according to a distribution calibrated according to historical rainfall data for the region from the years 1920–1980. The number of weather events that were necessary to occur to be considered an extreme event year was determined by using peaks-over-threshold for the historical data.

The reward function used for training the policies of the farmer agents (Eq 2.11) is defined by the ratio of the squared realized profits of a time-step (Eq 2.9) and the expected profits at that time-step (Eq 2.10), translated from the range of all possible profits $(P_{\min}, P_{\max})$ to the range $(-1, 1)$.

$$
P_{net}(t) = \sum_x P_x(t) G_1(P_p, B, t) S(B, EE) + G_2(P_p, B, t) \tag{2.9}
$$

$$
P_{exp}(t) = \sum_x P_x(t) G_1(P_p, B, t) + G_2(P_p, B, t) \tag{2.10}
$$

$$
R_f(t) = \frac{P_{net}(t)^2}{P_{exp}(t)} : (P_{\min}, P_{\max}) \to (-1, 1) \tag{2.11}
$$

The one municipal regulatory agent is used to model a municipal government or regulatory agency's behavior managing agricultural practices on the landscape and the local environment and the policies that guide them. This agent acts more slowly than the agricultural agents, once every five time-steps, and decides if/how it should modify its incentive structure — changing its taxation rate, the subsidization given

to an agent adopting a BMP, and the phosphorus runoff threshold at which a penalty is applied. The state properties of the regulatory agent are listed in Table 2.3, and details on their initialization are included in Section A.3.1.

Table 2.3: Table of the state properties of the regulatory agent and their associated data type for the regulatory agent in the agricultural model.

Name		Description	Data Type
Agent ID			`uint`
Aggregate Agent Data			
BMP Adoption			`float[15]`
Extreme Events			`uint[15]`
Financial History			`float[15]`
P Runoff History			`float[15]`
Regulation Change Limit	g		`float`
P Tax Rate	T_p		`float`
P Tax Threshold	P_t		`float`
BMP Subsidy Value	S_b		`float`
Neural Networks			
Actor Network	μ	Network Weights	`float[l][w]`
Critic Network	Q	Network Weights	`float[L][W]`
Target Network	μ'	Network Weights	`float[l][w]`
Target Network	Q'	Network Weights	`float[L][W]`
Memory Bank			`float[M*R]`
Memory Buffer			`float[B*R]`

The components of actions that the regulatory agent can take are listed in Ta-

Table 2.4: A summary of the action factors being used to drive agent decision-making for both types of agent present in the model.

Regulator Agent

Group	Action
Tax Rate	Increase by $[0, T_g^+)$
	Decrease by $[0, T_g^-)$
BMP Subsidy	Provide/Increase
	Remove/Decrease
Phosphorous Threshold$\star$	Scale

ble 2.4 The phosphorus threshold adjustment action is notably implemented differently in that it is a single value which is having the sign taken to determine the direction of the adjustment.

The results of taking these actions is a shift in the regulatory parameters of the system as shown in the set of equations below (Eq 2.12).

$$\begin{array}{c|rcl} a_0 & T_{p,t+1} & = & T_{p,t} + \min_{\geq 0}\left(T_p^+, \mathcal{N}\left(\delta T_p, g+k\right)\right) \\ a_1 & T_{p,t+1} & = & T_{p,t} - \min_{\geq 0}\left(T_p^-, \mathcal{N}\left(\delta T_p, g+k\right)\right) \\ a_2 & S_{b,t+1} & = & T_{p,t} + \min_{\geq 0}\left(S_b^+, \mathcal{N}\left(\delta S_b, g+k\right)\right) \\ a_3 & S_{b,t+1} & = & T_{p,t} - \min_{\geq 0}\left(S_b^-, \mathcal{N}\left(\delta S_b, g+k\right)\right) \\ a_4 & P_{t,t+1} & = & P_{t,t} + \text{signum}\left(a_4\right) \end{array} \tag{2.12}$$

The parameter changes impact the incentive structures provided by the phosphorus taxing function, G_1, shown in Equation 2.13, and the BMP subsidization function, G_2, shown in Equation 2.14.

$$G_1(P, B, t) = \begin{cases} T_p & P_p \geq P_t \\ 1 & P_p < P_t \end{cases} \tag{2.13}$$

Table 2.5: A summary of the state factors being used as input to the agents' ANNs.

Farmer Agent

Group	Description	Detail
Land Cover	Cropland	Normalized Area (sq m)
	Pasture	Normalized Area (sq m)
Productivity	Corn	See A
	Hay	
	Dairy	
	Beef	
History (5-year)	Extreme Event Record	Occurrence
	Financial Record	A
	BMP Adoption Record	
Network Information	Financials	Losses (1-year, 5-year)
	BMP Adoption	Usage (1-year, 5-year)

Regulator Agent

Group	Description	
Aggregate Data	BMP Adoption	
	Financials	Net Profits, Losses
	P Runoff	
History	Extreme Event Record	5-year, 15-year
	BMP Adoption	
	Financials	Net Profits, Losses
	P Runoff	

$$G_2(P, B, t) = \begin{cases} S_b & B \\ 0 & \neg B \end{cases} \tag{2.14}$$

The goal of the regulator agent is to minimize the aggregate phosphorous output and storm loss of all agricultural agents, $R_r = \left\langle W_p \sum_f P_{p,f}(t), W_l \sum_f l_f \right\rangle$, where W_p and W_l are normalizing weights on the component inverse reward signals so that they vary along ranges of similar magnitude.

The parameters for agent learning for both agent types are summarized in Table 2.6.

Table 2.6: Network parameters for the ANNs used by agents in each class for the land cover model

Parameter	Agricultural		Regulatory	
	μ	Q	μ	Q
Input Nodes	15	32	10	15
Inner Layers	4	3	4	3
Inner Nodes	10	16	7	7
Output Nodes	17	1	5	1
Connectivity	— Full —			
Activation Function	— ReLU —			
Output Activation	5-hot	Linear	2-hot + 1-Signum	Linear

Model Hyperparameters

Preliminary model runs were conducted to determine the optimal values for the hyperparameters for machine learning within the model. The learning hyperparameters that were varied in these preliminary runs were the number of training episodes, the number of steps between target network updates, the number of inner layers in the neural networks, the number of neurons in each of those inner layer, the learning rate,

and the batch size. The learning hyperparameters that were held constant were the exploration rate at $\epsilon = 0.1$, the discount factor at $\gamma = 0.99$, the learning transfer rate at $\tau = 0.001$, the number of steps within a training episode at $N = 40$, the relay memory size at $M = 10000$.

Table 2.7: Hyperparameters and their associated values with source or rationale, if applicable, for the agricultural land-use model.

Parameter		Value	Source/Rationale
Learning Rate	α	0.00025	
Exploration Rate	ϵ	0.1	
Discount Factor	γ	0.99	
Transfer Rate	τ	0.001	[19]
Relay Memory Size	M	10000	[19]
Number of Episodes	N	1000	
Number of Steps per Episode	T	40	Economic production function limitation

2.2.2 Experimental Setup

The model was run for a variety of scenarios. All scenarios tested the variables BMP_e, ΔEE, and g. There were two classes of test: tests with agents with uniform memory accuracy (Table 2.8) and tests with agents with heterogeneous memory accuracy (Table 2.9).

BMP Efficacy (B_e) was varied from 0.0 to 1.0 in increments of 0.1. This parameter represents the effectiveness of BMPs in reducing nutrient loading from agricultural fields. A value of 0.0 indicates that BMPs have no effect on nutrient loading, while

a value of 1.0 indicates that BMPs completely eliminate nutrient loading. This parameter was varied in order to determine the effect of BMP efficacy on the behavior of the system.

Change in weather event frequency (ΔEE) was varied from -0.2 to 0.2 in increments of 0.05. This parameter represents the change in the frequency of extreme weather events, such as heavy rainfall, that may be induced by climate change compared to a historical baseline, so in cases where $\Delta EE = 0.2$, the expected number of extreme rainfall events that are going to occur will be 1.2 times the expected number of extreme rainfall events according to the baseline.

The regulation change limiter (g) represents the maximum rate at which the regulatory agent will adjust the regulatory environment. Three values were tested: an aggressive limit ($g = 0.2$), a moderate limit ($g = 0.05$), and a restrictive case ($g = 0$) for testing the model's ability to operate in a static regulatory environment. This value alters the width of the distribution that the regulatory parameters, T_p and S_b, are adjusted by.

The impact of agent memory accuracy was tested for two types of agent populations. In uniform agent populations, all agents had the same memory recall accuracy (F), where F is the probability that a memory will be recalled correctly. In heterogeneous agent populations, agents had different memory recall accuracy, where a proportion of agents (P) had accuracy $F = 1$ and all other agents ($1 - P$) had accuracy $F = 0$.

Table 2.8: Table listing experimental parameters for uniform population runs

Variable		Values
BMP Efficacy	B_e	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Change in Event Frequency	ΔEE	-0.2, -0.15, -0.1, -0.05, 0.0, 0.05, 0.1, 0.15, 0.2
Regulation Change Limiter	g	0, 0.05, 0.2
Forgetfulness Factor	F	0, 0.25, 0.5, 0.75, 1

Table 2.9: Table listing experimental parameters for mixed population runs

Variable		Values
BMP Efficacy	B_e	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Change in Event Frequency	ΔEE	-0.2, -0.15, -0.1, -0.05, 0.0, 0.05, 0.1, 0.15, 0.2
Regulation Change Limiter	g	0, 0.05, 0.2
Population Mixing	P	0.25, 0.5, 0.75

2.3 Results

2.3.1 Model Performance

For each model parameterization, agents were trained for 1000 training episodes. If more than 10% of the agents ($n = 48$) in the model failed to converge to a stable policy within 1000 training episodes, the model was discarded and retrained; however, this occurred in less than 2% of model runs. A plot showing the distribution of number of agents which converged across model parameterizations is shown in Figure 2.2.

In this training, an agent's networks were considered to have converged if after 50 initial training episodes, the net change in the weights of the network during a transfer learning step was less than 10^{-5}. This threshold was chosen to be small enough to ensure that the networks had reached some stable policy, but large enough to avoid overfitting.

Models which were successfully trained and passed through this screening were

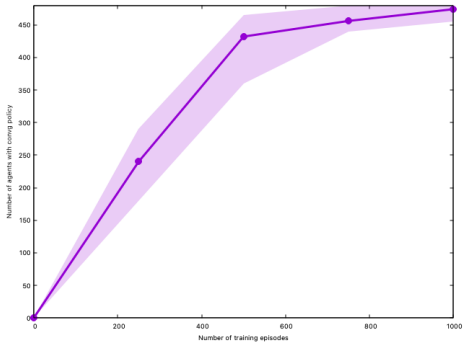
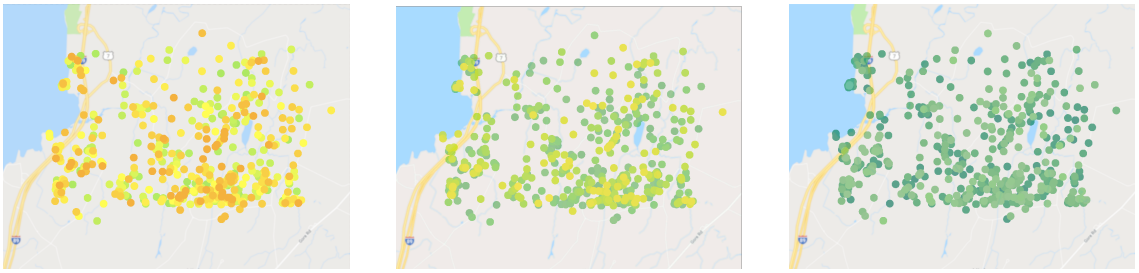


Figure 2.2: Plot of the number of agents that converged to a stable policy for each parameterization of the model



(a) Initial State (y=2001)

(b) End State (y=2040, g=0.05, $\Delta EE = 0$)

(c) End State (y=2040, g=0.2, $\Delta EE = 0.2$)

Figure 2.3: Sample model output showing the change in BMP adoption likelihood from a characteristic initial model state (a) to a characteristic end states for (b) a model run with the parameterization ($g = 0.05$, $\Delta EE = 0$), and (c) a model run with the parameterization ($g = 0.20$, $\Delta EE = 0.2$). The color of each dot represents the likelihood that the agent will adopt BMPs, with green indicating a high likelihood and red indicating a low likelihood.

then run for 40 testing runs. Within this section, some results have been omitted for readability. A listing of results and their associated model parameterizations can be found in Appendix C.

2.3.2 Agent Behavior

Uniform Population Runs

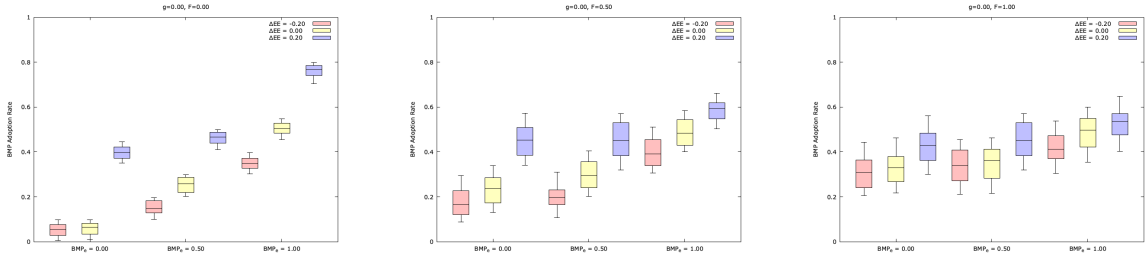
For model parameterizations with uniform agent populations, the proportion of agents which adopted a BMP in each testing model run was recorded and used to generate a distribution of BMP adoption rates for each parameterization. Summaries of the results of these runs are shown in Figure 2.4 for the case where the regulation change threshold (g) was set to 0.0, Figure 2.5 for when g was set to 0.05, and Figure 2.6 for when g was set to 0.2.

Mixed Population Runs

For model parameterizations with mixed agent populations, agents were divided into three groups: group 1, where $F = 0$ for the agent and all neighbors, group 2, where $F = 1$ for the agent and all neighbors, and group 3, for agents with neighbors where $F = 0$ and $F = 1$. The proportion of agents in each group which adopted a BMP in each testing model run was recorded and used to generate a distribution of BMP adoption rates for each parameterization. Results of one set of parameterizations of these runs are shown in Figure 2.7 where $g = 0$, $\Delta EE = 0$.

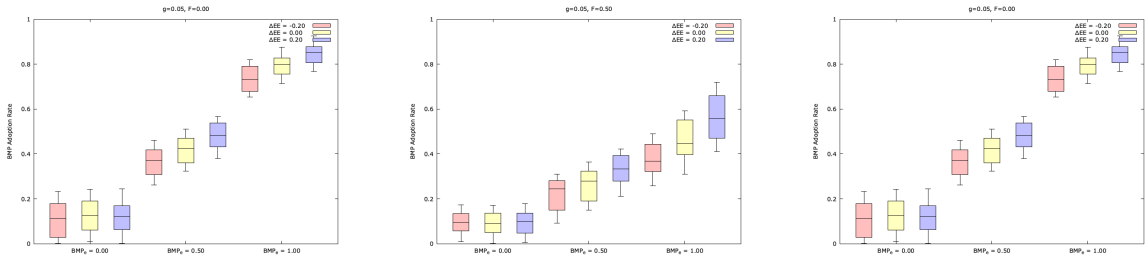
2.4 Discussion

Overall, results seem to indicate that this method of introducing deep reinforcement learning into agent-based modeling does have some viability for driving agent decision-making; however, some components of the experimental testing show how sensitive



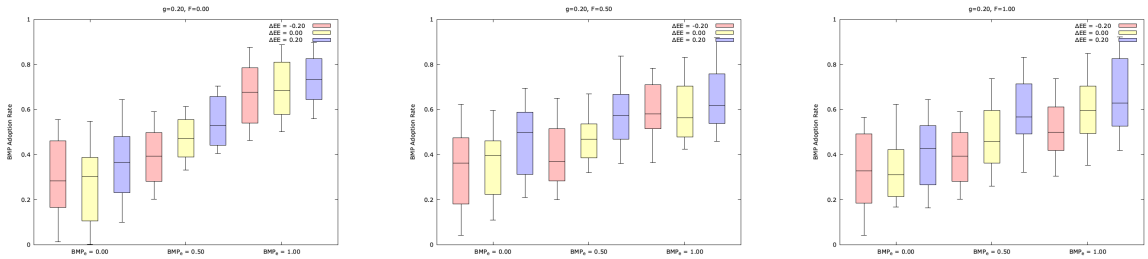
(a) $F = 0$ *(b)* $F = 0.5$ *(c)* $F = 1.0$

Figure 2.4: Distribution of mean BMP adoption rate for uniform population runs of the agricultural land use model, where $g = 0.0$*, for (a)* $F = 0$*, (b)* $F = 0.5$*, and (c)* $F = 1.0$



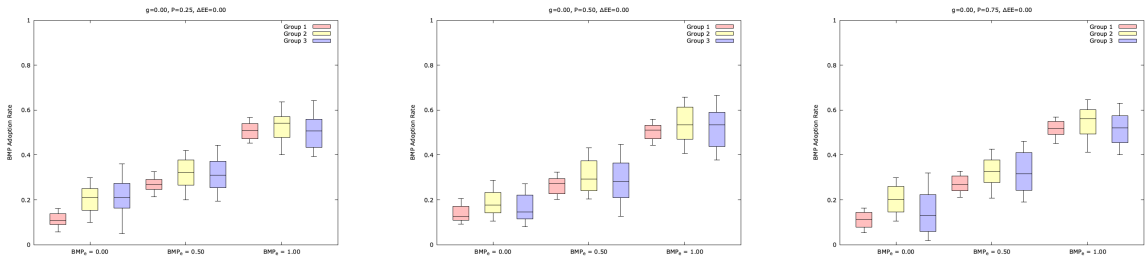
(a) $F = 0$ *(b)* $F = 0.5$ *(c)* $F = 1.0$

Figure 2.5: Distribution of mean BMP adoption rate for uniform population runs of the agricultural land use model, where $g = 0.05$*, for (a)* $F = 0$*, (b)* $F = 0.5$*, and (c)* $F = 1.0$



(a) $F = 0$ *(b)* $F = 0.5$ *(c)* $F = 1.0$

Figure 2.6: Distribution of mean BMP adoption rate for uniform population runs of the agricultural land use model, where $g = 0.2$*, for (a)* $F = 0$*, (b)* $F = 0.5$*, and (c)* $F = 1.0$



(a) $P = 0.25$ *(b)* $P = 0.5$ *(c)* $P = 0.75$

Figure 2.7: Distribution of mean BMP adoption rate for mixed population runs of the agricultural land use model, where $g = 0.0$, $\Delta EE = 0$, *for (a)* $P = 0.25$, *(b)* $P = 0.5$, *and (c)* $P = 0.75$

this kind of model can be be to its parameterization.

The purpose of the regulatory agent in the experimental model was to help incentivize specific agent behaviors, but the experimental parameter being used (g) had such a strong impact on the variability in agent behavior that the results of model runs where $g = 0.2$ had such high variance that it is difficult

Similarly, in the mixed population runs, the experimental method for introducing heterogeneity into the population can introduce variance in observed behaviors, but further testing would be needed to know if it leads to any of the desired emergent behavioral patterns. An experimental study specifically targeting an analysis of the heterogeneity in behaviors seen in real-world populations would be an ideal next-step.

Chapter 3

Increasing ABM Integration

The previous chapter demonstrated how an agent-based model can be implemented with machine learning in order to induce a model of individual behavior. This chapter will go on to show how adding a transfer learning calibration step, in cases where there exists some external metric for determining the ABM's performance as a result of agent behavior, can be integrated into this kind of ABM in order to adjust agent training in a way that increases model performance. This method is applied in the creation of a land-cover transition model of a study area in the Lake Champlain Basin of Vermont.

3.1 Methodology

The methodology used here is similar to the methodology presented in Chapter 2, with the addition of an external calibration step that takes place during model training. Because there is an external method for validating the output of the model for each training episode, a traditional learning step can be incorporated into the transfer learning step as a form of adaptive transfer learning, where there base transfer rate τ

in the transfer learning update (Eq 2.3) is replaced with an adjusted transfer rate τ_{adj} as in Equation 3.1.

$$\forall \theta'_i \in \Theta' \left[\theta'_i \leftarrow \tau_{adj} + (1 - \tau_{adj})\theta'_i\right] \tag{3.1}$$

For a given calibration method, that is, a loss function for overall model performance, the adjusted transfer rate can be set for each transfer update based on the performance of the model in each transfer episode. This allows transfer episodes with a higher metric of model performance to have a greater impact on the weights of the target network architecture, whereas transfer episodes that demonstrated poorer performance would have a lesser impact on the target architecture.

3.2 Experimental Design

3.2.1 Model Overview

In order to test this calibration methodology and an increased degree of integration of deep machine learning techniques into agent-based modeling, a second experimental ABM was developed. This ABM is also a model of a study area within the Lake Champlain Basin of Vermont, but expanded to included systems outside of the agricultural sector, as forestry and urban commercial and residential systems are also included. In this model, agent decision-making not only impacts their economic state, but also impacts how the land-cover of the land associated with each agent develops and changes over time. The land-cover change is modeled as the stochastic byproduct of agent action, wherein, for example, an agricultural agent deciding to increase

its productivity with regard to grazing animals may result in an increase in its beef production factor p_b or a conversion of some unused forested land cells on the agent's property into in-use pasture land cells.

Unlike the model described in Chapter 2, the individual behavior of agents in this model is not its primary area of interest. Because real-world data exists for land-cover, there exists an external, objective method for determining the accuracy of model performance within a training episode. The goal of this model is to explore the potential of using real-world land-cover data and the calibration step and its adjustments to learning transfer to steer agent learning, and consequently decision-making, in the direction of an overall increase in the predictive accuracy of the model.

Representing Land Cover

Real world land cover data for the study area was taken for the United States Geological Survey's National Land Cover Dataset (NLCD) for four years: 2001, 2006, 2011, and 2016. This dataset divides the study area into a matrix of 30m by 30m land cells which have been assigned one of 15 NLCD land cover classes. For the purposes of this model, these 15 classes were divided into 6 cover categories: urban, forested, agricultural, barren, grassland/scrub, and other. These categories and their corresponding NLCD land cover classes are listed in Table 3.1. A plot of the NLCD representation of the study area for NLCD year 2001 is shown in Figure 3.1.

The land cells, as they exist in the NLCD data, provide the initial land cover for the model within the starting year. These land cells have been divided into land parcels, which are the collections of cells that an agent in the model has control over. The properties that constitute the land cells within this model are listed in Table 3.2,

Table 3.1: A listing of the 15 NLCD cover classes that are present in the dataset for the study area, their NLCD encoding, and their associated cover category within the model.

Cover Category	NLCD Cover Class	NLCD Encoding
Urban	Open Space Urban	21
	Low Density Urban	22
	Medium Density Urban	23
	High Density Urban	24
Forested	Deciduous Forest	41
	Evergreen Forest	42
	Mixed Forest	43
Agricultural	Pasture	81
	Crops	82
Barren	Barren	31
Grassland/Scrub	Scrub	52
	Grassland	71
Other	Water	11
	Wetlands Woody	90
	Wetlands Other	95

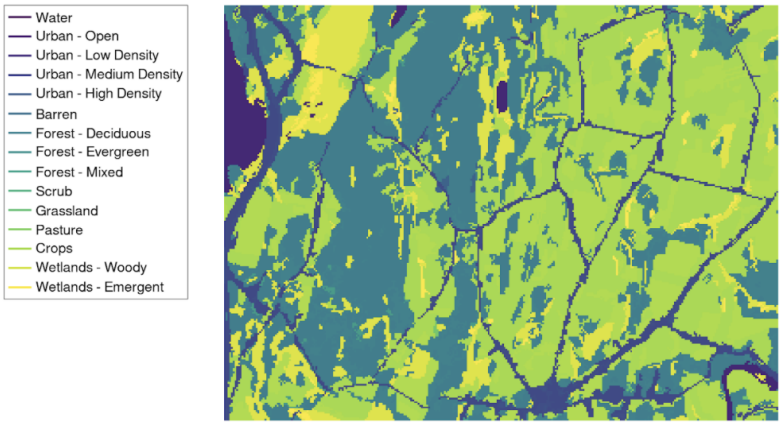


Figure 3.1: A plot of the NLCD data for the selected study area showing land-cover in real-year 2001, each 30m by 30m land cell is colored based on its NLCD cover class (Table 3.1), with similar colors being numerically closer classes.

Table 3.2: Land Cell Features

Parameter	Values
Parcel ID	Parcel that this cell is a part of
Land Cover Category	Agricultural, Forested, Urban, Other
NLCD Cover Class	See Table 3.1
Land Usage Type	Managed/In-Use, Adjacent, Unmanaged

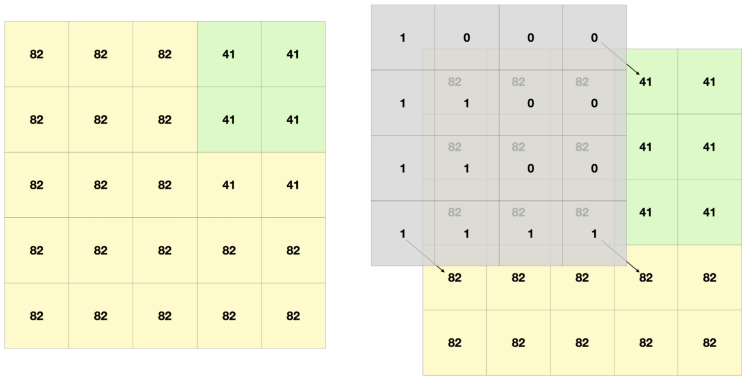


Figure 3.2: Diagram showing how land cells and their associated properties are represented within the model as a grid of NLCD land cover values (left), and how those values have land use properties mapped onto them 1-1 (right).

and a diagram showing how this data is overlaid on the NLCD data is shown in Figure 3.2.

Initial land-use for the cells within the model is generated via a stochastic process. Within the land-use initialization, all cells start with a label of "unmanaged", then a number of cells within each parcel are labeled as "managed", depending on the agent type and weighted towards population centers. Finally, all cells which are "unmanaged" and border a "managed" cell are labeled as "adjacent".

Agents

There are four types of agent present in this model: agricultural agents, forestry agents, commercial agents, and residential agents. There is one agent assigned for

each land-parcel in the model, and the agent type is assigned based on the majority land-cover of the parcel.

Agricultural agents model the behavior of farmers, herders, and other kinds of agricultural land managers within the study area. They make annual decisions about their farming practices, including whether they should change production in one of the four modeled agricultural industries (beef, dairy, corn, and hay) and whether they should implement an agricultural best management practice (BMP) to reduce phosphorous runoff on their land. This agent type has a very similar implementation in this model as was described for the agricultural model in Section 2.2, but with the land-cover change further broken down by land-use. It's modified state factors are listed in Table 3.3, and the actions it can take are listed in Table 3.4.

Table 3.3: A summary of the state factors being used during decision-making for agricultural agents in the land-cover model.

Group		Description	Detail
Land Cover	$c_{c,m}$	Cropland/In-use	Cell Count
	$c_{c,a}$	Cropland/Adjacent	Cell Count
	$c_{p,m}$	Pasture/In-use	Cell Count
	$c_{p,a}$	Pasture/Adjacent	Cell Count
	$c_{a,u}$	Agricultural/Unmaintained	Cell Count
	$c_{o,a}$	Other/Adjacent	Cell Count
Productivity	p_c	Corn	
	p_h	Hay	
	p_d	Dairy	
	p_b	Beef	

Table 3.3: (continued...)

Group	Description	Detail
History	BMP Adoption Record	
	Extreme Event Record	
	Net Profits/Losses	
Network Information	BMP Adoption	
	Net Profits/Losses	

In order to prevent any direct interference with the production functions, an additional land-use decision action was added to the list of actions for the agricultural agent, to explicitly capture the intent to change the scope of land-use. This action had initially been planned for inclusion in the agricultural model described in Chapter 2, but the lack of real-world land-cover data in the associated datasets prevented its implementation.

Table 3.4: A summary of the action factors being used to drive agent decision-making for agricultural agents in the land-cover model.

Group	Action
BMP Usage	Adopt BMP
	Don't Adopt BMP
Corn Production	Increase by $[0, S_c^+)$
	Maintain
	Decrease by $[0, S_c^-)$
Hay Production	Increase by $[0, S_h^+)$
	Maintain

Table 3.4: (continued...)

Group	Action
	Decrease by $[0, S_h^-)$
Dairy Production	Increase by $[0, S_d^+)$
	Maintain
	Decrease by $[0, S_d^-)$
Beef Production	Increase by $[0, S_b^+)$
	Maintain
	Decrease $[0, S_b^-)$
Land-Use Decision	Grow
	Maintain
	Shrink

Here, it behaves as follows. A decision to "maintain" land-use means that no land-use changes will occur. A decision to "grow" land-use means that the agent will search for up to 2 land cells which are of land-use "adjacent" to convert into land-use "maintained", and will adjust its land area calculations to match. A decision to "shrink" land-use means that the agent will attempt the opposite transition, converting up to 2 land cells from land-use "maintained" to land-use "adjacent". In all other agent types, this behavior is implicit to the actions that the agents take and their associated scoping.

Forestry agents model the behavior of loggers and other kinds of forested land managers within the study area. They make annual decisions about their practices and whether to implement an advised management practice (AMP) on their land.

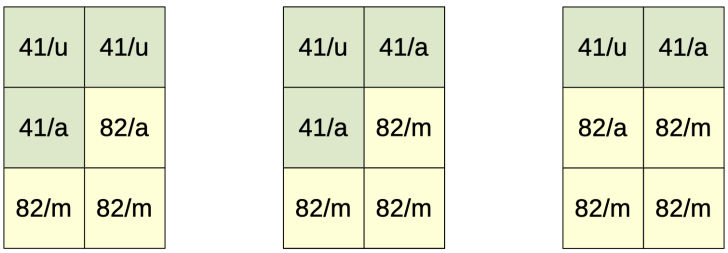


Figure 3.3: Diagram showing how land-use, and consequently land-cover, may change for an example 2-by-3 parcel over the course of a time-step.

The forestry agents are implemented very similarly to the agricultural agents, but the land-cover of interest has been changed to forested land, and the production function has been replaced with a generalized forested productivity function. The state factors used by the forestry agents in their decision-making are listed in Table 3.5, and the actions that these agents can take are listed in Table 3.6.

Table 3.5: A summary of the state factors being used during decision-making for forestry agents in the land-cover model.

Group		Description	Detail
Land Cover	$c_{f,m}$	Forested/In-use	Cell Count
	$c_{f,a}$	Forested/Adjacent	Cell Count
	$c_{f,u}$	Forested/Unmaintained	Cell Count
	$c_{o,a}$	Other/Adjacent	Cell Count
Productivity	p_f	Forestry	
History		AMP Adoption	
		Extreme Event Record	
		Net Profits/Losses	
Network Information		AMP Adoption	

Table 3.5: (continued...)

Group	Description	Detail
	Net Profits/Losses	

Table 3.6: A summary of the action factors being used to drive agent decision-making for forestry agents in the land-cover model.

Group	Action
AMP Usage	Adopt AMP
	Don't Adopt AMP
Forestry Decision	Increase by $[0, S_f^+)$
	Maintain
	Decrease by $[0, S_f^-)$

Agricultural and forestry agents are connected to and share information with their 5-nearest neighbors of the same agent type; these networks are static throughout each model run. For both of these agent types, their learning reward is based on their net profitability as was described in Chapter 2.

Commercial agents model the behavior of shops, factories, offices, and other kinds of commercial land-holders within the study area. They make decisions bi/trimonthly about their workforce, including their available jobs and the associated salaries. Byproducts of their actions impact the density and sprawl of urban land cover on the landscape. The state factors that it uses in decision-making are listed in Table 3.7, and the actions they can take are listed in Table 3.8.

Table 3.7: A summary of the state factors being used during decision-making for commercial agents in the land-cover model.

Group	Description
Financial Status	
Employees	Capacity
	Actual
	Utilization

Table 3.8: A summary of the action factors being used to drive agent decision-making for commercial agents in the land-cover model.

Group	Action
Business Capacity	Decrease
	Maintain
	Increase
Employees	Fire
	Maintain
	Hire

Commercial agents use a simple compound reward function for their performance, with a 'living reward' t_l, which grows as the number of time steps the commercial agent has gone without declaring bankruptcy / employee count becoming zero, and the realized utilization U_e of their land, such that the reward $R_c = t_l * U_e$.

Residential agents model the behavior of renters and landowners within the study area. They make two decisions annually: whether to attempt a job change and whether to try to move houses. Household satisfaction, and their reward value R_r, is

valued as a combination of financial stability and mental satisfaction. Each household earns wages provided by a commercial agent — these wages are determined by a stochastic process and can be adjusted by the job over time. The decisions of these agents do not directly impact land cover change on their associated parcel, but land cover can transition within their parcel as a result of the decisions of other agents.

Table 3.9: A summary of the state factors being used during decision-making for residential agents in the land-cover model.

Group	Description
Financial Status	
Length in State	
Household Budget	Monthly
Household Budget	Yearly
Failed Action Count	Consecutive

Table 3.10: A summary of the action factors being used to drive agent decision-making for residential agents in the land-cover model.

Group	Action
Employment	Search for new job
	Keep current job
Housing	Search for new housing
	Keep same housing

The reward function used by residential agents is similar to that used by commercial agents, but the living reward is scaled by the agent's financial status.

Commercial and residential agents exist in a bipartite network with one another.

This network is initialized via a stochastic process and is updated as agents make decisions. This process is detailed in Appendix B, alongside other details of agent initialization.

The learning architecture for agents of each type is listed in Table 3.11. Similarly to the agent-networks described in the model in Chapter 2, all networks are fully-connected, use ReLU activation and He Initialization, and use n-hot output to encode their selected action. Additionally, all target architectures are initially identical to the active network architectures as specified below.

Table 3.11: Network parameters for the ANNs used by agents in each class for the land cover model

Parameter	Agricultural		Forestry		Commercial		Residential	
	μ	Q	μ	Q	μ	Q	μ	Q
Input Nodes	15	32	10	15	4	10	5	9
Inner Layers	4	3	4	3	2	2	2	2
Inner Nodes	10	16	7	7	5	5	4	5
Output Nodes	17	1	5	1	6	1	4	1

Model Hyperparameters

A summary of the fixed hyperparameters across all runs of this model are listed in Table 3.12. Many of the parameters are taken directly from the agricultural model described in Chapter 2, with the economic production functions taking from the corresponding model years and the weather generation submodel using the baseline case of $\Delta EE = 0$.

The length of the training episode was set to 60 time-steps, or 5 model-years, to match the granularity of the real-world NLCD datasets. Preliminary model runs for determining appropriate fixed model hyperparameters were done with the land-cover

data from NLCD year 2001 as input data, comparing the data from NLCD year 2006 with the model output year 2006.

Table 3.12: Fixed hyperparameters and their associated values for the land-cover change model.

Parameter		Value
Learning Rate	α	0.00025
Exploration Rate	ϵ	0.1
Base Transfer Rate	τ	0.001
Transition Memory Size	M	10000
Number of Steps per Episode	T	60

Evaluating Model Performance

For this model, the Nash-Sutcliffe efficiency index (NSE) was used to evaluate the goodness-of-fit of model output at the end of each training episode. The NSE is a measure of the relative magnitude of the residual variance of modeled data compared to the residual variance of the observed data. The value of the index ranges from $-\infty$ to 1, where a score of 1 indicates a perfect fit, a score of 0 indicates that the model's fit is no better than the mean of the observed data, and a score less than 0 indicates that the mean of the observed data is a better predictor than the model.

This index was calculated in relation to three forms of model performance. The first is the ability of the model to appropriately predict the proportional coverage of each land cover type in the target year, NSE_{plc}. The second is the ability of the model to appropriately predict the categorical transitions of land cover types from the start year to the target year, NSE_{cat}. The third is the ability of the model to appropriately

predict the absolute transitions of land cover types from the start year to the target year, NSE_{abs}. These indices are detailed below.

A majority of land cells in the study area do not transition land cover between the start year and target year ($92.6\%, n = 69124$), which would heavily bias any analysis of model performance. Therefore, these NSE indices were only calculated for those cells that transitioned land cover.

The NSE measure of proportional coverage (NSE_{plc}) is shown in Equation 3.2, where P_B represents the observed proportional coverage of land cover category B in the target year, $\hat{P_B}$ represents the simulated proportional coverage of land cover category B in the target year, and $\bar{P_B}$ represents the mean observed proportional coverage of land cover category B in the target year.

$$\text{NSE}_{\text{plc}} = \frac{\sum_B \left(P_B - \hat{P_B}\right)^2}{\sum_B \left(P_B - \bar{P_B}\right)^2} \tag{3.2}$$

The NSE measure of categorical land cover transitions (NSE_{cat}), shown in Equation 3.3, where $\Delta_{A,B}$ represents the number of observed transitions from land cover category A in the starting year to land cover category B in the target year, for the overarching categories shown in Table 3.1, where $\widehat{\Delta_{A,B}}$ represents the number of simulated transitions from A to B, and where $\overline{\Delta_{A,B}}$ represents the mean observed number of transitions from A to B.

$$\text{NSE}_{\text{cat}} = \frac{\sum_{A,B} \left(\Delta_{A,B} - \widehat{\Delta_{A,B}}\right)^2}{\sum_{A,B} \left(\Delta_{A,B} - \overline{\Delta_{A,B}}\right)^2}, A \neq B \tag{3.3}$$

The NSE measure for absolute land cover transition (NSE_{act}), shown in Equation 3.4, is very similar to the calculation of NSE_{cat}, except that it is calculated for

the absolute land cover class of each land cell and not just its categorical class. This index is a stricter metric than the previous two indices, as, for example, a modeled transition from cropland to deciduous forest $\delta_{82,41}$ compared to an observed transition to mixed forest $\delta_{82,43}$, which would be labeled "correct" according to NSE_{cat} would be considered a misclassification according to NSE_{abs}.

$$\text{NSE}_{\text{abs}} = \frac{\sum_{a,b} \left(\Delta_{a,b} - \widehat{\Delta_{a,b}}\right)^2}{\sum_{a,b} \left(\Delta_{a,b} - \overline{\Delta_{a,b}}\right)^2}, a \neq b \tag{3.4}$$

For the purposes of model calibration, as described in Section 3.1, the model was calibrated to maximize the result of NSE_{cat} for the transition from the starting year 2006, to the observed and modeled year 2011.

Execution Overview

A high-level overview of model execution, showing the main training loop with the transfer calibration step and the final testing loop, is shown in Figure 3.4. A more-detailed overview of model execution, specifically focused on the behavior and training and agents is shown in Figure 3.5.

3.2.2 Experimental Setup

In order to avoid some of the high levels of variance seen in the experimental runs of the agricultural model and that model's sensitivity to agent paramterization, these experimental runs were limited in scope to a subset of model hyperparameters that relate to agent memory and foresight, listed in Table 3.13. Each combination of parameterizations was tested for 40 model runs.

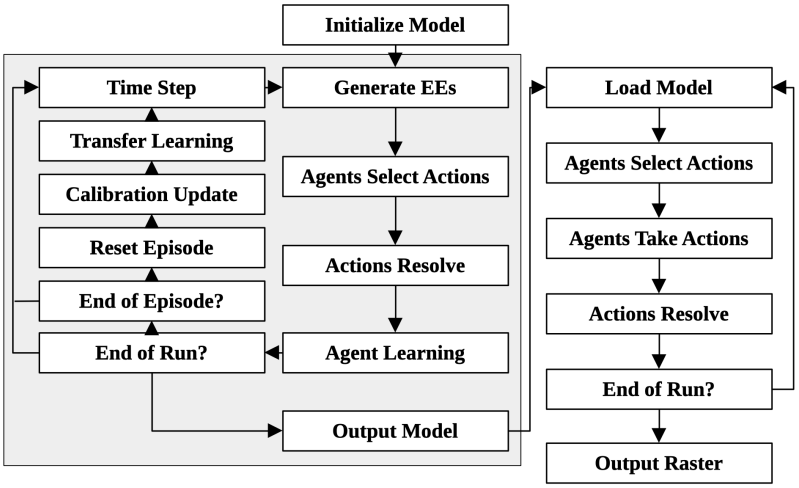


Figure 3.4: Flowchart showing a high-level overview of model execution between the main training/calibration loop (left), and the final testing runtime loop (right)

Replay batch size B was tested for the sizes 8, 16, 32, and 64. This value determines how many state transition records are sampled during the action-replay and policy gradient learning steps. A higher number of records sampled increases the smoothness of the gradient being sampled, at the cost of extra computation.

The discount factor γ was tested for the values $0.5, 0.9$, and 0.99. This factor is used in the Q-learning update (Eq 2.1) to discount the expected rewards at a future time-step. A lower value of γ indicates a lower value of the reward in step $t+1$ than the current reward in step t, which compounds multiplicably with each additional time-step forward.

The recall accuracy F', is the notational inverse of the forgetfulness factor described in Section 2.1.4, where $F' = (1 - F)$ and consequently a recall accuracy of 1 indicates agents with 0 forgetfulness, that is, perfect record recall.

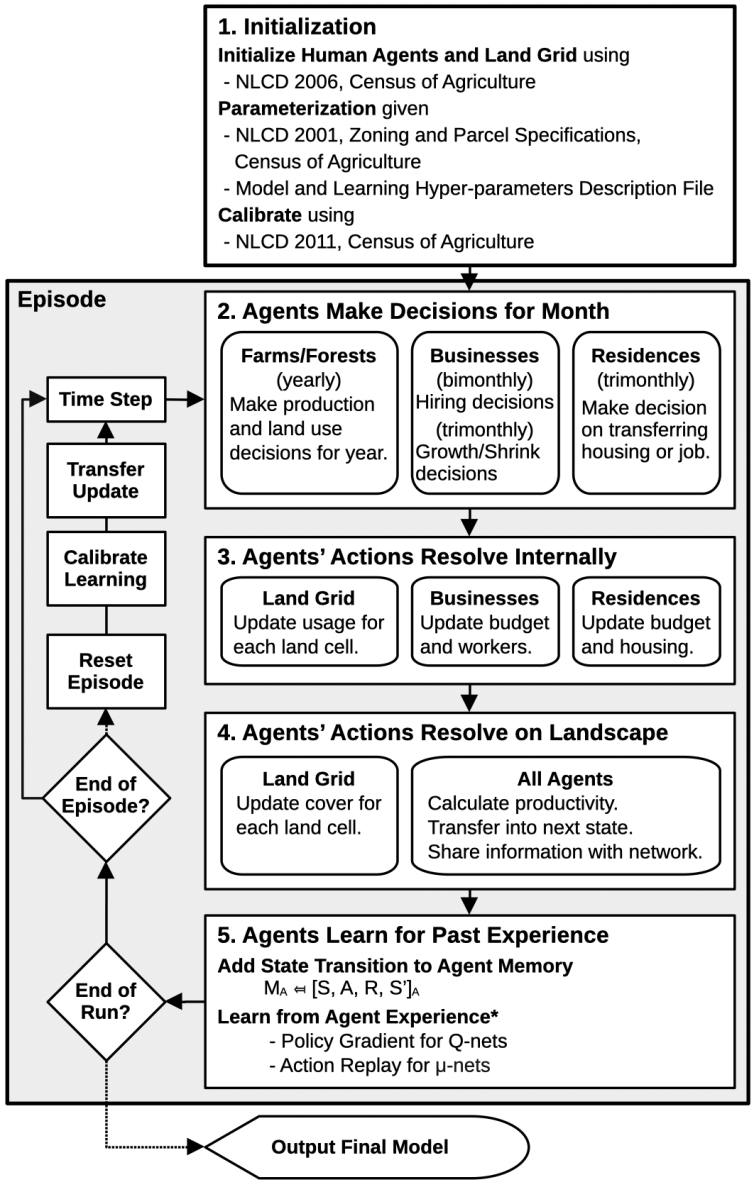
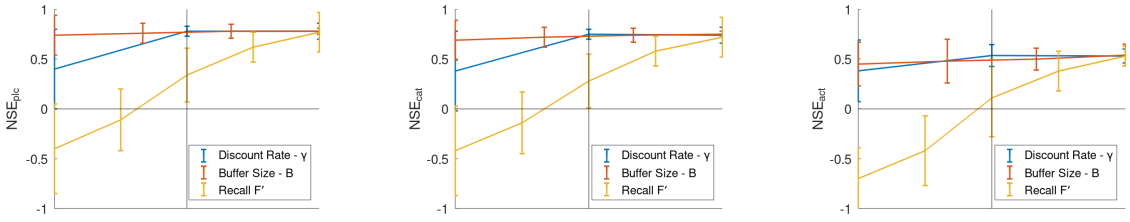


Figure 3.5: Flowchart demonstrating the overall execution of the agent-based model and its coupling with the machine learning processes

Table 3.13: Experimental parameters that were tested in experimental runs of the land-cover transition model

Variable		Values
Batch Size	B	8, 16, 32, 64
Discount Factor	γ	0.5, 0.9, 0.99
Recall Accuracy	F'	0, 0.25, 0.5, 0.75, 1



(a) NSE_{plc} *(b) NSE_{cat}* *(c) NSE_{act}*

Figure 3.6: NSE index sensitivity for each index showing the variance in model classification accuracy by each metric under different model parameterizations.

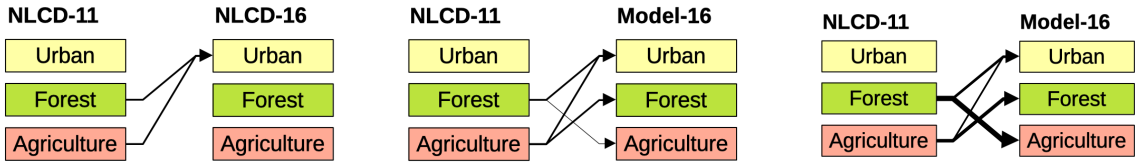
3.3 Results

3.3.1 Model Performance

Model performance under each experimental parameterization was evaluated by comparing the land cover in model year 2016 to the recorded/observed land cover for the study area for real year 2016.

The maximum NSE values seen during testing runs were $\text{NSE}_{\text{plc}} = 0.84$, $\text{NSE}_{\text{cat}} = 0.76$, and $\text{NSE}_{\text{act}} = 0.64$. The sensitivity of these indices to each of the experimental parameters was evaluated by calculated the mean and variance of the indices under each parameterization across all 40 runs. These results have been plotted in Figure 3.6.

In order to better understand the output of the model and where misclassification



(a) Observed Cover Change *(b) "Best Fit" Cover Change* *(c) Best Median Change*

Figure 3.7: Arrow plots showing a comparison of the categorical transitions between the three main land-cover categories between (a) the real world data, and (b) the "best-fit" model, and (c) the "best median" model. Arrow width between NLCD-11 and 2016 is proportional to the number of transitions between the connected categories.

was occurring within testing runs, the land-cover transition matrices were analyzed. Categorical cover change had been used as the calibration function for the model, so land cover transitions were divided into three primary categories: urban U, forested F, and agricultural A, and an other O category for all other land-cover categories; and instances where both the real and modeled data had a matching transition from a land-cover category to itself were removed.

For the purposes of this discussion, the two land-cover transition matrices that will be compared against the actual land-cover transition are the "best fit" model ($\text{NSE}_{\text{cat}} = 0.76$) and the "best median" model, the median performer for the highest scoring parameterization during testing ($\text{NSE}_{\text{cat}} = 0.68$, $\gamma = 0.99$, $B = 64$, $F' = 1$). Arrow plots, summarizing how land-cover transitions compared between these models, are shown in Figure 3.7, with the associated confusion matrices listed in Table 3.14

Within the study area, cells with urban land-cover did not transition into any other category of land-cover during the study period. Forested and agricultural cells which did transition categories, primarily developed into urban land-cover, followed by transitions into other categorizations. These forested and agricultural transitions are the primary source of the classification error within these models and are discussed

Table 3.14: Confusion matrix comparing the resulting categorical transitions from the NLCD data to the modeled transitions for both the best fit, and best performing parameterization. Because the initial cover category for both transitions are definitionally equal, it has been omitted from the header row. The transition $\Delta_{*,O}$ represents a transition outside one of the three major cover categories.

	"Best Fit" 2016				Best Median 2016			
NLCD 2016	$\Delta_{*,U}$	$\Delta_{*,F}$	$\Delta_{*,A}$	$\Delta_{*,O}$	$\Delta_{*,U}$	$\Delta_{*,F}$	$\Delta_{*,A}$	$\Delta_{*,O}$
$\Delta_{U,U}$	—	0	0	0	—	0	0	0
$\Delta_{U,F}$	0	0	0	0	0	0	0	0
$\Delta_{U,A}$	0	0	0	0	0	0	0	0
$\Delta_{U,O}$	0	0	0	0	0	0	0	0
$\Delta_{F,U}$	7	0	0	0	7	0	0	0
$\Delta_{F,F}$	1	—	5	0	0	—	7	0
$\Delta_{F,A}$	0	0	0	0	0	0	0	0
$\Delta_{F,O}$	0	7	0	36	3	5	5	30
$\Delta_{A,U}$	11	0	0	0	11	0	0	0
$\Delta_{A,F}$	0	0	0	0	0	0	0	0
$\Delta_{A,A}$	7	1	—	0	3	14	—	0
$\Delta_{A,O}$	0	4	4	3	1	0	10	0

further is the following section.

3.4 Discussion

The experimental parameters which were tested each had a different degree of impact on model performance. The recall factor F', had the strongest impact on model performance, with values less than 0.75 resulting in model runs that were worse performers than the probabilistic average for NSE_{act}, and with a value of 0.5 resulting in worse performance than average for all 3 indices. In the case of NSE_{act}, there is a noticeable drop in model performance for the case of $\gamma = 0.99$ compared to $\gamma = 0.9$; however, this change is not large enough to be statistically significant when the variance in model performance for the case of $\gamma = 0.9$ is considered.

Looking at the transition plots and confusion matrices for some of the higher accuracy model runs, the largest source of error within the tested models came from land-cover transitions between the agricultural and forested land-cover categories. These transitions, like most land cover transitions between cover categories, were poorly represented within the calibration data, as land-cover transition is a relatively slow process that even the 5-year interval of the NLCD dataset has difficulty capturing. Similarly, the highest performing models tended to overclassify urbanization land-cover changes, which were the most represented cross-category land-cover change in the calibration data set.

The categorical classification into the group other O was the most accurate cross-category predictor — in particular, categorical transition into the Grassland/Scrub cover-category. These transitions are present in the calibration and testing datasets, and frequently occur in proximity to changes in agricultural and forested land-cover changes. It is possible that this is related to the high confusion in the land-cover transitions between agricultural and forested land-cover regions, but that level of testing would require land-cover datasets for similar regions with similar land-cover transition patterns, which were outside of the scope of this modeling effort.

If this work were to be continued, it would be interesting to see how this type of integrated land-cover transition model would perform over a longer period of time or wihtin a larger study area, so the number of and type of land-cover transitions occurring within the calibration dataset would be more representative of the land-cover transitions occurring within the study period.

Bibliography

[1] Thomas C Schelling. Dynamic models of segregation. *Journal of mathematical sociology*, 1(2):143–186, 1971.

[2] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

[3] Robert Axelrod and William D Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

[4] Philip W Anderson. More is different: Broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047):393–396, 1972.

[5] E. Bonabeau. Application of simulation to social sciences. *Hermés Sciences*, pages 451–461, 2000.

[6] Robert Axelrod. The complexity of cooperation: Agent-based models of competition and collaboration. *Princeton Univ. Press*, 1997.

[7] Michael J Prietula, Kathleen M Carley, and Les Gasser. *Simulating organizations: Computational models of institutions and groups*. Mit Press, 1998.

[8] Alan Turing. Computing machinery and intelligence. *Mind*, 59(236):433, 1950.

[9] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[10] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[11] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

[12] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge United Kingdom, 1989.

[13] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering, 1994.

[14] Peter Dayan. The convergence of td (λ) for general λ. *Machine learning*, 8:341–362, 1992.

[15] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[16] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[17] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-Garadi, and Uzoma Rita Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, 105:233–261, 2018.

[18] Dongbin Zhao, Haitao Wang, Kun Shao, and Yuanheng Zhu. Deep reinforcement learning with experience replay based on sarsa. In *2016 IEEE symposium series on computational intelligence (SSCI)*, pages 1–6. IEEE, 2016.

[19] Hado V. Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[20] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] Egemen Sert, Yaneer Bar-Yam, and Alfredo J Morales. Segregation dynamics with reinforcement learning and agent based modeling. *Scientific reports*, 10(1):1–12, 2020.

[22] Carla Gomes, Thomas Dietterich, Christopher Barrett, Jon Conrad, Bistra Dilkina, Stefano Ermon, Fei Fang, Andrew Farnsworth, Alan Fern, Xiaoli Fern, et al. Computational sustainability: Computing for a better world and a sustainable future. *Communications of the ACM*, 62(9):56–65, 2019.

[23] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset

characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[24] Lisa Torrey and Jude Shavlik. Transfer learning. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1:242, 2009.

[25] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[26] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[27] Ladislav Rampasek and Anna Goldenberg. Tensorflow: Biology gateway to deep learning? *Cell systems*, 2(1):12–14, 2016.

[28] Asim Zia, Ciara Low, and Danielle Shaw. Hedonic value of water quality in lake champlain basin: A multi-level modeling approach. *Ecological Economics*, 2017.

[29] Asim Zia, Arne Bomblies, Andrew W Schroth, Christopher Koliba, Peter D F Isles, Yushiou Tsai, Ibrahim N Mohammed, Gabriela Bucini, Patrick J Clemins, Scott Turnbull, Morgan Rodgers, Ahmed Hamed, Brian Beckage, Jonathan Winter, Carol Adair, Gillian L Galford, Donna Rizza, and Judith Van Houten. Coupled impacts of climate and land use change across a river-lake continuum: insights from an integrated assessment model of lake champlain's missisquoi basin, 2000–2040. *Environmental Research Letters*, 11(11), 2016.

[30] Bongghi Hong, Karin E Limburge, Jon D Erickson, John M Gowdy, Audra A Nowosielski, John M Polimeni, and Karen M Stainbrook. Connecting the ecological-economic dots in human-dominated watersheds: Models to link socio-economic activities on the landscape to stream ecosystem health. *Landscape and Urban Planning*, 91:78–87, June 2009.

[31] Ji He, Man Lan, Chew-Lim Tan, Sam-Yuan Sung, and Hwee-Boon Low. Initialization of cluster refinement algorithms: A review and comparative study. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 1, pages 297–302. IEEE, 2004.

[32] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 407–412, 2010.

Appendix A
Farm Model ODD+D Document

This appendix contains an ODD+D design document detailing the specifications of the farmer behavior model described in the Chapter 2.

A.1 Overview

A.1.1 Purpose

This model aims to explore the behavior of farmers within the Missisquoi Bay Area of the Lake Champlain Basin of Vermont. In particular, the goal is to look at how farmers within the region may choose to change their land-use practices and adopt or reject agricultural best management practices (BMPs) and how a government regulator may implement taxes or subsidies on farming practices in an attempt to stymie environmental damages to the ecosystems of Lake Champlain.

This model features 480 farmer agents, corresponding to the agricultural land parcels within the Missisquoi Bay Area, and one municipal regulatory agent. Agents receive input from their environment, including inter-agent communication and stochastic environmental factors (viz., simulated extreme weather events). Agents make decisions as frequently as once per model year, and the decision policies guiding their decision-making are trained using deep reinforcement machine learning.

A.1.2 Entities, State Variables, and Scales

Study Area

The study area being used as the basis of this model is a subsection of the Missisquoi Bay Area of the Lake Champlain Basin of Vermont. Each of the 480 agriculturally-zoned land parcels in the Bay Area is used as the land basis for an agent in the model,

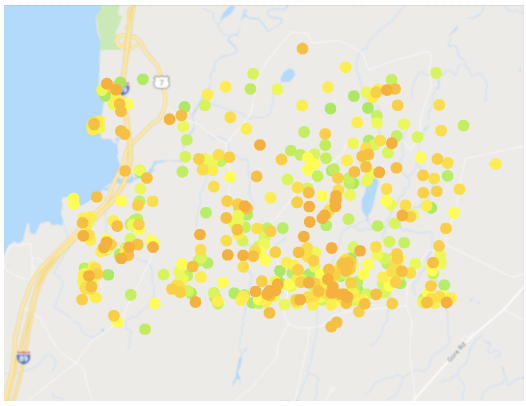


Figure A.1: A map of the study area and the locations of the agricultural agents within it.

with each agricultural agent having control of one parcel of land. A map of the study area and the locations of agents within it is shown in Figure A.1.

Agricultural Agents

Agricultural agents model the behavior of farmers, herders, and other kinds of agricultural land managers within the study area. They make annual decisions about their farming practices, including whether they should change production in one of the four modeled agricultural industries (beef, dairy, corn, and hay) and whether they should implement an agricultural best management practice (BMP) to reduce phosphorus runoff on their land.

The state properties and variables that each agricultural agent has are listed in Table A.1. The initialization of these values is detailed in Section A.3.1.

Table A.1: Table of all state properties of agricultural agents and their associated data type for agricultural agents in this model.

Name	Description	Data Type
Agent ID	Unique identifier for this agent	`uint`
Agent Status		`enum{3}`
Land Parcel Data		
Crop Land Area A_c	Land devoted to growing crops (sq km)	`float`
Pasture Land Area A_p	Land devoted to grazing animals (sq km)	`float`
Total Land Area A_{tot}	Total land in parcel (sq km)	`float`
Productivity		
Corn p_c	Corn production factor	`float`
Hay p_h	Hay production factor	`float`
Beef p_b	Beef production factor	`float`

Table A.1: (continued...)

Name	Description	Data Type
Dairy p_d	Dairy production factor	`float`
Phosphorous $p_{p,x}$	Phosphorus production factor	`float`
Cows Owned C		`uint`
Financial History		
Real Net	What was net profit over last 5 years	`float[5]`
Expected Net	What was expected profit for last 5 years	`float[5]`
Extreme Event History	Extreme event presence over past 5 years	`uint[5]`
BMP Usage History B	Did farm use BMP in last 5 years	`bool[5]`
Neighbors	References to neighboring agents	`farmer*[5]`
Neural Networks		
Actor Network μ	Network Weights	`float[l][w]`
Critic Network Q	Network Weights	`float[L][W]`
Target Network μ'	Network Weights	`float[l][w]`
Target Network Q'	Network Weights	`float[L][W]`
Memory Bank		`float[M*R]`
Memory Buffer		`float[B*R]`

The components of actions that agricultural agents can take are listed in Table A.2.

Table A.2: Table of components of actions that farmer agents can take and their associated encoding group for n-hot encoding

Group	Action	Encoding Index
BMP Usage	Adopt BMP	0
	Don't Adopt BMP	1
Corn Production	Increase by $[0, S_c^+)$	2
	Maintain	3
	Decrease by $[0, S_c^-]$	4
Hay Production	Increase by $[0, S_h^+]$	5
	Maintain	6
	Decrease by $[0, S_h^-)$	7
Dairy Production	Increase by $[0, S_d^+)$	8
	Maintain	9
	Decrease by $[0, S_d^-)$	10
Beef Production	Increase by $[0, S_b^+)$	11
	Maintain	12
	Decrease $[0, S_b^-)$	13

The production factors were derived by an underlying economic model and are calibrated from 2001–2040. The equations used are shown for the production of corn (Eq A.1), hay (Eq A.2), dairy (Eq A.4), and beef (Eq A.3), below.

$$P_c(t) = p_c * A_c^b * 11.433 \log t - 86.826 \tag{A.1}$$

$$P_h(t) = p_h * A_c^b * 1\text{e}{-}32 \exp 0.0358t \tag{A.2}$$

$$P_b(t) = p_b * A_p^b * 2\text{e}{-}20 \exp 0.0234t \tag{A.3}$$

$$P_d(t) = p_d * A_p^b * 2\text{e}{-}9 \exp 0.0114t \tag{A.4}$$

The productivity of the agent is modified by the application of the regulatory agent's regulations G_1 and G_2 and the amount of losses due to extreme weather events (Eq A.5) as a function of whether the BMP was used and whether the number of extreme events that occurred within the given year exceeds the expected threshold N.

$$\begin{array}{lcll} S(B, EE) & = & 1, & EE < N \\ S(B, EE) & = & 0.1, & EE \geq N, \quad \neg B \\ S(B, EE) & = & (0.1 + 0.9 B_{eff}), & EE \geq N, \quad B \end{array} \tag{A.5}$$

The reward function uses for training the policies of the farmer agents (Eq A.8) is defined by the ratio of the squared realized profits of a time-step (Eq A.6) and the expected profits at that time-step (Eq A.7), translated from the range of all possible profits $(P_{\min}, P_{\max})$ to the range $(-1, 1)$.

$$P_{net}(t) = \sum_x P_x(t) G_1(P_p, B, t) S(B, EE) + G_2(P_p, B, t) \tag{A.6}$$

$$P_{exp}(t) = \sum_x P_x(t) G_1(P_p, B, t) + G_2(P_p, B, t) \tag{A.7}$$

$$R_f(t) = \frac{P_{net}(t)^2}{P_{exp}(t)} : (P_{\min}, P_{\max}) \to (-1, 1) \tag{A.8}$$

Regulatory Agent

In addition to the numerous agricultural agents, the model contains a single regulatory agent that models the behavior of a municipal government or regulatory agency managing agricultural practices on the landscape and the local environment. Slower acting than the agricultural agents, the regulator agent acts once every 5 model years, optionally modifying its incentive structure.

The state properties of the regulatory agent are listed in Table A.3, and details on their initialization are included in Section A.3.1.

Table A.3: Properties of Regulatory Agent

Name	Description	Data Type
Agent ID	Identifier for this agent	`uint`
Aggregate Agent Data		
BMP Adoption		`float[15]`
Extreme Events		`uint[15]`
Financial History		`float[15]`
P Runoff History		`float[15]`

The components of actions that the regulatory agent can take are listed in Table A.4. The phosphorus threshold adjustment action is notably implemented differently in that it is a single value which is having the sign taken to determine the direction of the adjustment.

Table A.4: Table of components of actions that the regulator agent can take and their associated encoding group

Group	Action	Encoding Index
Tax Rate	Increase by $[0, T^+)$	0
	Decrease by $[0, T^-)$	1
BMP Subsidy	Provide/Increase	2
	Remove/Decrease	3
Phosphorous Threshold$\star$	Scale	4

The goal of the regulator agent is to minimize the aggregate phosphorous output and storm loss of all agricultural agents, $R_r = \left\langle W_p \sum_f P_{p,f}(t), W_l \sum_f l_f \right\rangle$, where W_p and W_l are normalizing weights on the component inverse reward signals so that they vary along ranges of similar magnitude.

A.1.3 Process Overview and Scheduling

Once every model year, starting with the first time step (t=0), the number of extreme weather events for the year is generated from a simple peaks-over-threshold weather event generator. The number of extreme weather events generated for the year impacts the agricultural agents' productivity for that year.

Agricultural agents then act every model year, deciding what action to take and resolving that decision on the landscape. Every 5 model years, the regulator agent reads the productivity and losses from the agricultural agents and decides how/if to alter its incentive strategy for adjusting agricultural agent behavior.

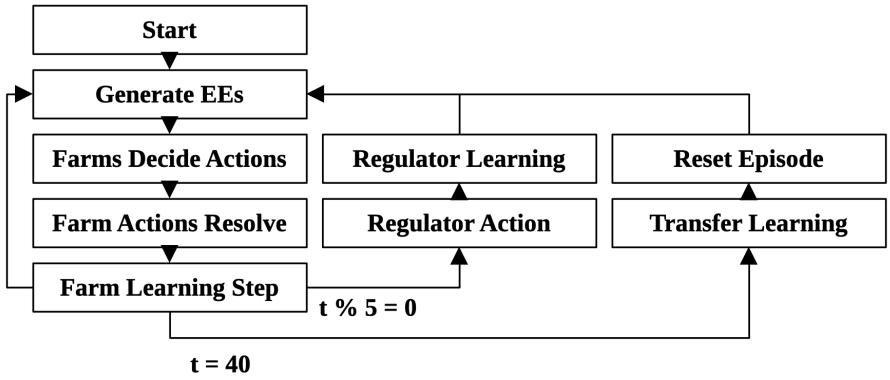


Figure A.2: A diagram of model execution throughout episodic training. Within each time step of the episode farmer agents make decisions and learn, every five time steps the regulator agent acts, and after 40 model years the episode ends and is reset.

An overview of model execution and the looping every model year, every 5 model years, and every training episode is shown in Figure A.2.

A.2 Design Concepts

A.2.1 Basic Principles

This model is an agent-based model where the agents exist on a shared grid representative of the chosen study area. Each agent has control over a portion of this grid as defined by parcel boundaries and can make decisions that can alter the productivity and land-use practices of the land they manage. Agents take actions independently, and their effect on the grid is synchronized at the end of every time step.

Agents create their decision-making policies using deep reinforcement learning, where environmental feedback and internal reward valuation drive the way their internal neural networks learn. Additionally, as they act, state-transitions are stored as memories within the agents, which are used to determine an agent's learning speed and verify that learning progresses in the right direction.

A.2.2 Emergence

There are a few areas of the model where interesting behavioral patterns are more likely to emerge and be observed. The most visible way is how the agricultural agents adopt/reject best management practices on their land. Their choices are also directly

observed and recorded, which allows for the observation of behavioral trends over time and how those trends correlate to factors like the number of extreme weather events that occur per model year.

A.2.3 Adaptation

Agents have a few different adaptive traits, viz. the neural networks used to drive agent decision-making. The networks take in components of the agents' current state and use them to select an action to take, over time encoding the agents' decision-policies and their reward valuation into the networks.

Additionally, the agricultural agents have several features related to agricultural production and BMP usage, which factor into their decision-making and can be adjusted over time.

A.2.4 Objectives

The two agent types have distinct, somewhat adversarial objectives with regard to adapting their decision-making. Both agent types are working towards optimizing their reward functions and associated valuation accuracy.

The goal of the agricultural agents is to maximize their yearly profits. The goal of the regulatory agent is to minimize the aggregate storm loss accrued by the farmer agents and their aggregate phosphorous output.

A.2.5 Learning

All human agents in the model learn and develop their decision-making policies using deep reinforcement machine learning, specifically an adapted version of DDQN. As the agents take actions in various states and transition between them, these state transitions are stored in the agents' memory. After resolving an action, the most recent state transition is compared against state transitions in agent memory to determine the direction and rate of learning within the problem space. The gradient generated from this calculation is then used to adjust the agents' neural networks' weights and steer their decision-policy towards selecting actions that will create state transitions most in line with their objectives.

A.2.6 Prediction

Because of how the internal neural networks work, the prediction of future state-transitions is an implicitly defined process. As each agent makes decisions and expe-

riences state transitions, it updates a valuation network that estimates the value of taking actions from states. This estimated valuation is compared against the actual reward received from taking that action, which is used to update the decision-making network. Because the actual value of taking an action will not be known until the action resolves on the landscape, the prediction is somewhat retroactively validated and used to drive network learning, as opposed to being used to explicitly and actively look forward while decision-making.

A.2.7 Sensing

Agricultural agents are capable of detecting some historical information about storm productivity losses and BMP usage from the five nearest neighboring agents. They also experience extreme weather events as they occur during model years, affecting their yearly productivity. The regulator agent can read information about the state of the agricultural agents as a collective group and can communicate a uniform regulatory policy to the agricultural agents. These data fields are listed in Table A.5.

Table A.5: A listing of information shared between agents in this model and its directionality

Value	From	To	Type
BMP Usage (5-year)	Farmers	5-nearest Neighbors	`bool[5]`
Net Profits/Losses (5-year)	Farmers	5-nearest Neighbors	`float[5]`
BMP Usage	Farmers	Regulator	`bool`
Net Profits/Losses	Farmers	Regulator	`float`
P Output	Farmers	Regulator	`float`
Losses (1-year)			

A.2.8 Interaction

Agents in this model neither communicate nor interact with one another directly; they do however passively share information as outlined in Section A.2.7. Additionally, the regulator agent has the ability to change the reward/incentive structure for the agriculture agents when it acts on 5-year intervals.

A.2.9 Stochasticity

Randomness is used in multiple components of the model. During agent initialization, triangle distributions generate the starting productivity for the agricultural agents, and He network initialization is used to set the initial neural network weights, which are detailed in Section A.3.1.

During training runs of the model, memories are selected as a basis buffer for learning using a uniform distribution. Additionally, in runs with nonzero forgetfulness factors, a degree of uniform stochasticity is added to the accuracy of memory recall proportional to the factor.

A.2.10 Collectives

Agricultural agents share some historical information with their 5-nearest neighbors, as outlined in Section A.2.7, but generally aren't collected in any higher ordered structuring outside of network analysis.

A.2.11 Observation

Several components of the model are tracked during runs and compiled as observational data. Agent decisions, farmer budgets, and overall network training/performance are recorded throughout model runs.

A.3 Details

A.3.1 Initialization

Several of the parameters are initialized according to triangular distributions, seen below. Neural networks had weights initialized according to the He initialization algorithm. [31].

$$\mathrm{Tri}_1(a,b,c) = \begin{cases} a + \sqrt{U(b-a)(c-a)} & \text{for } 0 < U < F(c) \\ b - \sqrt{(1-U)(b-a)(b-c)} & \text{for } F(c) \leq U < 1 \end{cases} \tag{A.9}$$

$$\mathrm{Tri}_2(a,b) = \mathrm{Tri}_1\left(a, b, \frac{a}{b}\right) \tag{A.10}$$

$$\mathrm{Tri}_3(\mu,\delta) = \mathrm{Tri}_1(\mu-\delta, \mu+\delta, \mu) \tag{A.11}$$

Farmer Agents

Many of the initial values used by the agricultural agents are initialized from the input data. The variables which are set during initialization are listed in Table A.6. Many of these values are selected from a triangular distribution which was parameterized according to the calibration of the underlying economic model.

Table A.6: Initialization of Farm Agents

Value		Initialization	Type
Agent ID	f	Assigned sequentially from 0	`uint`
Parcel Data	A_x	Read from input file	`float`
Corn Production (USD)	p_c	$\text{Tri}_2(3.362551, 4259232)$	`float`
Corn Production (P)	$p_{p,c}$	$\text{Tri}_2(2.02\text{e}{-4}, 6.17\text{e}{-4})$	`float`
Hay Production (USD)	p_h	$\text{Tri}_2(0.358672, 0.470757)$	`float`
Hay Production (P)	$p_{p,h}$	$\text{Tri}_2(3.37\text{e}{-5}, 1.12\text{e}{-4})$	`float`
Beef Production (USD)	p_b	$\text{Tri}_2(900.0, 1200.0)$	`float`
Dairy Production (USD)	p_d	$\text{Tri}_2(210.0, 250.0)$	`float`
Cow Production (P)	$p_{p,[bd]}$	$\text{Tri}_2(3.366\text{e}{-4}7.853\text{e}{-4})$	`float`
Cows Owned	C	$\text{Tri}_2(3600, 7800)$	`uint`
Actor-Network Weights	Θ_μ	He()	`float[][]`
Critic-Network Weights	Θ_Q	He()	`float[][]`
Target-Actor Weights	$\Theta_{\mu'}$	Copied from Θ_μ	`float[][]`
Target-Critic Weights	$\Theta_{Q'}$	Copied from Θ_Q	`float[][]`

Regulatory Agent

The regulatory agent is more reactive than proactive, so most of its internal parameters cannot be set until after the agricultural agents have began to act; however, the parameters that are generated during initialization are listed in Table A.7.

Table A.7: Initialization of Regulatory Agent

Value		Initialization	Type
Agent ID	r	Only 1, so set to 0	`uint`
Actor-Network Weights	Θ_μ	He()	`float[][]`
Critic-Network Weights	Θ_Q	He()	`float[][]`
Target-Actor Weights	$\Theta_{\mu'}$	Copied from Θ_μ	`float[][]`
Target-Critic Weights	$\Theta_{Q'}$	Copied from Θ_Q	`float[][]`

Appendix B

Land-Cover Model Design Overview

This appendix contains an overview of the design of the land-cover transition model described in Chapter 3. It generally adheres to the technical principles of the ODD+D format, as shown in Appendix A, but is focused on where the specification provides details that do not appear in the main text of this thesis.

B.1 Overview

B.1.1 Purpose

The purpose of this model is to explore potential changes to land cover as a result of human behavior as it develops in response to projected climatological, economic, and social scenarios within study areas of the Lake Champlain Basin of Vermont. Four types of human agents are present in this model; these agents represent some of the various types of people who make decisions that can change land cover on the landscape. Agents received input from their environment, including inter-agent communication and stochastic environmental factors (f.x. simulated extreme weather events). Agents made decisions as frequently as once per model month, and the decision policy guiding their decision-making was trained using deep reinforcement machine learning.

B.1.2 Entities, State Variables, and Scales

Study Area

The study area being used as the basis of this model is a subsection of the Lake Champlain Basin of Vermont. Each parcel within the study area is treated as an agent within the model. A map of the study area and its initial land-cover is shown in Figure 3.2.

Agents

There are four types of human agents present in this model — agricultural, commercial, residential, and forester. These agents represent various types of landowners/managers within each study area who are able to make decisions that can affect land-use and land-cover. They make these decisions based on both their material state and their perceived mental and financial state. Agent behavior is trained using deep reinforcement machine learning, which provides each agent with a decision-policy that guides their decision-making during test model runs.

The state variables that are present in every human agent in the model are outlined in Table B.1

Table B.1: Table of all properties that are shared amongst all human agents in the land-cover transition model.

Name		Description	Data type
Agent ID		Unique identifier for this agent	`uint`
Parcel ID		Identifier of underlying land cell parcel	`uint`
Agent Status		Class-dependent internal status	`enum`
Neural Networks			
Actor Network	Θ_μ	Network Weights	`float[][]`
Critic Network	Θ_Q	Network Weights	`float[][]`
Target Actor	$\Theta_{\mu'}$	Network Weights	`float[][]`
Target Critic	$\Theta_{Q'}$	Network Weights	`float[][]`

Agricultural Agents

Agricultural agents model the behavior of farmers, herders, and other kinds of agricultural land managers within each study area. They make annual decisions about their farming practices, including whether they should change production in one of the four modeled agricultural industries (beef, dairy, corn, and hay) and whether they should implement an agricultural best management practice (BMP) to reduce

phosphorous runoff on their land. The additional state variables that are found in every agricultural agent in the model are outlined in Table B.2.

Table B.2: Table of all state properties of agricultural agents and their associated data type for agricultural agents in the land-cover transition model.

Name		Description	Data Type
Land Parcel Data (sq km)			
Crop Land Area	A_c	Land devoted to growing crops	`float`
Pasture Land Area	A_p	Land devoted to grazing animals	`float`
Total Land Area	A_{tot}	Total land in parcel	`float`
Land Cover (Cell Count)			
	$c_{c,m}$	Cropland/In-use	`uint`
	$c_{c,a}$	Cropland/Adjacent	`uint`
	$c_{p,m}$	Pasture/In-use	`uint`
	$c_{p,a}$	Pasture/Adjacent	`uint`
	$c_{a,u}$	Agricultural/Unmaintained	`uint`
	$c_{o,a}$	Other/Adjacent Cell Count	`uint`
Productivity			
Corn	p_c	Corn production factor	`float`
Hay	p_h	Hay production factor	`float`
Beef	p_b	Beef production factor	`float`
Dairy	p_d	Dairy production factor	`float`
Phosphorous	$p_{p,x}$	Phosphorus production factors	`float[3]`
Cows Owned			`uint`
Financial History (5-year)			
Realized Net		Net yearly production	`float[5]`
Expected Net		Expected yearly production	`float[5]`
Extreme Event History			`bool[5]`
BMP Usage History		Did farm use BMP in last 5 years	`bool[5]`
Neighbors		References to neighboring agents	`uint[5]`

Forestry Agents

Forestry agents model the behavior of loggers and other kinds of forested land managers within the study area. They make annual decisions about their practices and whether to implement an advised management practice (AMP) on their land. The forestry agents are implemented very similarly to the agricultural agents, but the land-cover of interest has been changed to forested land, and the production function has been replaced with a generalized forested productivity function. The additional state factors used by the forestry agents in their decision-making are listed in Table B.3.

Table B.3: Table of state properties of forestry agents and their associated data types in the land-cover transition model.

Name		Description	Data Type
Land Parcel Data (sq km)			
Forested Land Area	A_p	Total forested land in parcel	`float`
Total Land Area	A_{tot}	Total land in parcel	`float`
Land Cover (Cell Count)			
	$c_{f,m}$	Forested/In-use	`uint`
	$c_{f,a}$	Forested/Adjacent	`uint`
	$c_{f,u}$	Forested/Unmaintained	`uint`
	$c_{o,a}$	Other/Adjacent Cell Count	`uint`
Productivity			
General	p_f	Relative production factor	`float`
Phosphorous	p_p	Relative production factor	`float`
Financial History (5-year)			
Realized Net		Net yearly production	`float[5]`
Expected Net		Expected yearly production	`float[5]`
Extreme Event History			`bool[5]`
AMP Usage History		Did agent use AMP in last 5 years	`bool[5]`
Neighbors		References to neighboring agents	`uint[5]`

Commercial Agents

Commercial agents model the behavior of shops, factories, offices, and other kinds of commercial land-holders within the study area. They make decisions bi/trimonthly about their workforce, including their available jobs and the associated salaries. Byproducts of their actions impact the density and sprawl of urban land cover on the landscape. The additional state factors that it uses in decision-making are listed in Table B.4.

Table B.4: Table of state properties of commercial agents and their associated data types in the land-cover transition model.

Name	Description	Data Type
Days Operational	Number of days operating	`uint`
Employee Count	Current number of employed agents	`uint`
Employee Capacity	Maximum number of employed agents	`uint`
Employee IDs	IDs of employed agents	`uint*`
Employee Salaries	Salaries of employed agents	`float*`
Total Pay	Total salaried paid to all employees	`float`

Name	Description	Data Type
Budget	Total monthly budget	`float`

Residential Agents

Residential agents model the behavior of renters and landowners within the study area. They make two decisions annually: whether to attempt a job change and whether to try to move houses. Household satisfaction, and their reward value R_r, is valued as a combination of financial stability and mental satisfaction. Each household earns wages provided by a commercial agent — these wages are determined by a stochastic process and can be adjusted by the job over time. The decisions of these agents do not directly impact land cover change on their associated parcel, but land cover can transition within their parcel as a result of the decisions of other agents.

Table B.5: Table of state properties of residential agents and their associated data types in the land-cover transition model.

Name	Description	Data Type
Employer ID	ID of current employer	`uint`
Housing Costs	Total monthly cost of living	`float`
Salary	Total monthly income from employer	`float`
Monthly Budget	Net income over past 1 month	`float`
Yearly Budget	Net income over past 12 months	`float[12]`
Time in State	Number of time steps with current 'mood'	`uint`
Failed Action Count	Number of consecutive failed actions	`uint`

B.1.3 Process Overview and Scheduling

Each time-step within the model is representative of one modeled month. Each model year (12 time-steps), the number of extreme weather events for the year are generated and the agricultural and forestry agents act.

The commercial agent has two primary decision-vectors, the decision to increase or decrease overall productivity is resolved trimonthly (3 time-steps), while the decision to hire/fire staff is resolved bimonthly (2 time-steps). At the end of each model year (12 time-steps), the commercial agent will increase the salary of all employed residential agents by 10%.

A commercial agent which loses its last employee, by firing or quitting, will enter the "bankrupt" state at the end of the occurent time-step, and will not be replaced with a "new" commercial agent until 3 time-steps have passed. This new agent has a period of 3–6 time-steps to acquire its first employee before it can bankrupt again.

The residential agent acts monthly (1 time-step); however, its actions are dependant on the behavior of other agents in the model — f.x. a residential agent cannot move into an occupied house regardless of its price. If a residential agent intends to leaves its job, house, or the entire system, this action cannot be blocked, it is only additive actions which face this restriction.

A new residential agent may only enter the system if there is an available residential parcel with an occupation status of "vacant" or "for-sale" and the total number of residential agents within the system does not match nor exceed the current residential capacity. This new residential agent will be added to the system at the start of the following time-step.

B.2 Initialization

B.2.1 Agent Connections

Agricultural and forestry agents are all connected to their 5-nearest agents of the same type. Since these agents do not move on the landscape, these networks are constant throughout each model run.

Commercial and residential agents exist within a bipartite network. During model initialization, each commercial agent starts with an employment capacity of 10. At model start, 90% of residential agents are assigned jobs selected with uniform likelihood from that available commercial capacity. The initial employment capacity within the selected study area ensures that there will always be more initial capacity than 90% of the initial residential capacity. These networks are dynamic and change as agents make decisions throughout each model run.

B.2.2 Initial Agent State

Many of these values are selected from triangular distributions which was parameterized according to the calibration of the underlying economic model. Triangular distributions, as described in Section A.3.1, are used for the initialization of many parameters. Neural networks had weights initialized according to the He initialization algorithm. [31].

Value		Initialization	Type
Agent ID		Assigned sequentially from 0	`uint`
Parcel ID		Read from Input Data	`uint`
Actor-Network Weights	Θ_μ	He()	`float[][]`
Critic-Network Weights	Θ_Q	He()	`float[][]`

Value		Initialization	Type
Target-Actor Weights	$\Theta_{\mu'}$	Copied from Θ_{μ}	`float[][]`
Target-Critic Weights	$\Theta_{Q'}$	Copied from Θ_{Q}	`float[][]`

Agricultural Agents

Many of the initial values used by the agricultural agents are initialized from the input data. The variables which are set during initialization are listed in Table B.7.

Table B.7: Table listing initialization of parameters of the agricultural agents in the land-cover transition model

Value		Initialization	Type
Parcel Data	A_x	Read from input file	`float`
Corn Production (USD)	p_c	$\text{Tri}_2(3.362551, 4259232)$	`float`
Corn Production (P)	$p_{p,c}$	$\text{Tri}_2(2.02\text{e}{-4}, 6.17\text{e}{-4})$	`float`
Hay Production (USD)	p_h	$\text{Tri}_2(0.358672, 0.470757)$	`float`
Hay Production (P)	$p_{p,h}$	$\text{Tri}_2(3.37\text{e}{-5}, 1.12\text{e}{-4})$	`float`
Beef Production (USD)	p_b	$\text{Tri}_2(900.0, 1200.0)$	`float`
Dairy Production (USD)	p_d	$\text{Tri}_2(210.0, 250.0)$	`float`
Cow Production (P)	$p_{p,[bd]}$	$\text{Tri}_2(3.366\text{e}{-4}, 7.853\text{e}{-4})$	`float`
Cows Owned	C	$\text{Tri}_2(3600, 7800)$	`uint`

Residential and Commercial Agents

Initial agent salaries are selected from the weighted categorical distribution described in Table B.8 on the range $(250, 1000)$. The initial rent/mortgage price for each urban land parcel is selected from a uniform distribution on the range $(400, 1100)$.

Table B.8: Table listing distribution of initial agent salaries and the weight on their probability.

Weight	Initial Salary
10	250
20	300
30	350
50	400
70	450
85	500
90	550
85	600

Table B.8: (continued...)

Weight	Initial Salary
70	650
60	700
50	750
50	800
50	850
50	900
30	950
20	1000

Appendix C
Result Listings

This appendix contains listings of model results which would be difficult to display alongside the main text.

Table C.1: Mean BMP adoption rate for uniform-population runs of the agricultural model for the parameterizations with results plotted in Figure 2.4, Figure 2.5, and Figure 2.6.

g	F	BMP_e	ΔEE	Adoption Rate
0.0	0.0	0.0	-0.2	0.053
0.0	0.0	0.0	0.0	0.058
0.0	0.0	0.0	0.2	0.040
0.0	0.0	0.5	-0.2	0.150
0.0	0.0	0.5	0.0	0.252
0.0	0.0	0.5	0.2	0.461
0.0	0.0	1.0	-0.2	0.349
0.0	0.0	1.0	0.0	0.505
0.0	0.0	1.0	0.2	0.759
0.0	0.5	0.0	-0.2	0.173
0.0	0.5	0.0	0.0	0.231
0.0	0.5	0.0	0.2	0.452
0.0	0.5	0.5	-0.2	0.202
0.0	0.5	0.5	0.0	0.300
0.0	0.5	0.5	0.2	0.450
0.0	0.5	1.0	-0.2	0.399
0.0	0.5	1.0	0.0	0.488
0.0	0.5	1.0	0.2	0.589
0.0	1.0	0.0	-0.2	0.310
0.0	1.0	0.0	0.0	0.333
0.0	1.0	0.0	0.2	0.427

Table C.1: (continued...)

g	F	BMP_e	ΔEE	Adoption Rate
0.0	1.0	0.5	-0.2	0.339
0.0	1.0	0.5	0.0	0.349
0.0	1.0	0.5	0.2	0.450
0.0	1.0	1.0	-0.2	0.411
0.0	1.0	1.0	0.0	0.491
0.0	1.0	1.0	0.2	0.529
0.05	0.0	0.0	-0.2	0.106
0.05	0.0	0.0	0.0	0.124
0.05	0.0	0.0	0.2	0.121
0.05	0.0	0.5	-0.2	0.364
0.05	0.0	0.5	0.0	0.418
0.05	0.0	0.5	0.2	0.478
0.05	0.0	1.0	-0.2	0.733
0.05	0.0	1.0	0.0	0.793
0.05	0.0	1.0	0.2	0.847
0.05	0.5	0.0	-0.2	0.092
0.05	0.5	0.0	0.0	0.088
0.05	0.5	0.0	0.2	0.094
0.05	0.5	0.5	-0.2	0.219
0.05	0.5	0.5	0.0	0.265
0.05	0.5	0.5	0.2	0.332
0.05	0.5	1.0	-0.2	0.378
0.05	0.5	1.0	0.0	0.463
0.05	0.5	1.0	0.2	0.566
0.2	0.0	0.0	-0.2	0.304
0.2	0.0	0.0	0.0	0.276
0.2	0.0	0.0	0.2	0.361
0.2	0.0	0.5	-0.2	0.396
0.2	0.0	0.5	0.0	0.473
0.2	0.0	0.5	0.2	0.548
0.2	0.0	1.0	-0.2	0.667
0.2	0.0	1.0	0.0	0.695
0.2	0.0	1.0	0.2	0.733
0.2	0.5	0.0	-0.2	0.333
0.2	0.5	0.0	0.0	0.356
0.2	0.5	0.0	0.2	0.454
0.2	0.5	0.5	-0.2	0.396

Table C.1: (continued...)

g	F	BMP_e	ΔEE	Adoption Rate
0.2	0.5	0.5	0.0	0.467
0.2	0.5	0.5	0.2	0.574
0.2	0.5	1.0	-0.2	0.5957
0.2	0.5	1.0	0.0	0.586
0.2	0.5	1.0	0.2	0.641
0.2	1.0	0.0	-0.2	0.324
0.2	1.0	0.0	0.0	0.340
0.2	1.0	0.0	0.2	0.405
0.2	1.0	0.5	-0.2	0.396
0.2	1.0	0.5	0.0	0.478
0.2	1.0	0.5	0.2	0.590
0.2	1.0	1.0	-0.2	0.520
0.2	1.0	1.0	0.0	0.608
0.2	1.0	1.0	0.2	0.661

Figure 2.7

Table C.2: Mean BMP adoption rate for mixed-population runs of the agricultural model for the parameterizations with results plotted in Figure 2.7 for each sub-population: (1) local $F = 0$, (2) local $F = 1$, and (3) mixed neighborhood.

g	Group	P	BMP_e	ΔEE	Adoption Rate
0.0	1	0.25	0.0	0.0	0.111
0.0	2	0.25	0.0	0.0	0.202
0.0	3	0.25	0.0	0.0	0.209
0.0	1	0.25	0.5	0.0	0.270
0.0	2	0.25	0.5	0.0	0.322
0.0	3	0.25	0.5	0.0	0.313
0.0	1	0.25	1.0	0.0	0.507
0.0	2	0.25	1.0	0.0	0.529
0.0	3	0.25	1.0	0.0	0.508
0.0	1	0.5	0.0	0.0	0.140
0.0	2	0.5	0.0	0.0	0.186
0.0	3	0.5	0.0	0.0	0.166
0.0	1	0.5	0.5	0.0	0.265
0.0	2	0.5	0.5	0.0	0.302
0.0	3	0.5	0.5	0.0	0.289
0.0	1	0.5	1.0	0.0	0.504

Table C.2: (continued)

0.0	2	0.5	1.0	0.0	0.537
0.0	3	0.5	1.0	0.0	0.522
0.0	1	0.75	0.0	0.0	0.111
0.0	2	0.75	0.0	0.0	0.199
0.0	3	0.75	0.0	0.0	0.149
0.0	1	0.75	0.5	0.0	0.271
0.0	2	0.75	0.5	0.0	0.323
0.0	3	0.75	0.5	0.0	0.321
0.0	1	0.75	1.0	0.0	0.518
0.0	2	0.75	1.0	0.0	0.546
0.0	3	0.75	1.0	0.0	0.518