Optimizing Crop Management with Reinforcement Learning and Imitation Learning

Ran Tao\textsuperscript{1}, Pan Zhao\textsuperscript{1}, Jing Wu\textsuperscript{1}, Nicolas Martin\textsuperscript{1}, Matthew T. Harrison\textsuperscript{2}, Carla Ferreira\textsuperscript{3}, Zahra Kalantari\textsuperscript{4} and Naira Hovakimyan\textsuperscript{1}

\textsuperscript{1}University of Illinois at Urbana-Champaign
\textsuperscript{2}University of Tasmania
\textsuperscript{3}Stockholm University
\textsuperscript{4}KTH Royal Institute of Technology
rant3@illinois.edu

Abstract

Crop management has a significant impact on crop yield, economic profit, and the environment. Although management guidelines exist, finding the optimal management practices is challenging. Previous work used reinforcement learning (RL) and crop simulators to solve the problem, but the trained policies either have limited performance or are not deployable in the real world. In this paper, we present an intelligent crop management system that optimizes nitrogen fertilization and irrigation simultaneously via RL, imitation learning (IL), and crop simulations using the Decision Support System for Agrotechnology Transfer (DSSAT). We first use deep RL, in particular, deep Q-network, to train management policies that require a large number of state variables from the simulator as observations (denoted as full observation). We then invoke IL to train management policies that only need a few state variables that can be easily obtained or measured in the real world (denoted as partial observation) by mimicking the actions of the RL policies trained under full observation. Simulation experiments using the maize crop in Florida (US) and Zaragoza (Spain) demonstrate that the trained policies from both RL and IL techniques achieved more than 45% improvement in economic profit while causing less environmental impact compared with a baseline method. Most importantly, the IL-trained management policies are directly deployable in the real world as they use readily available information.

1 Introduction

The agricultural industry worldwide faces significant challenges regarding food security and nutrition and sustainable agriculture, which relate to one of the United Nations Sustainable Development Goals (SDGs), Zero Hunger. Humanity must produce food for a population expected to reach 9.6 billion by 2050 and simultaneously reduce environmental impacts, including ecosystem degradation and high greenhouse gas emissions [Searchinger et al., 2019]. There are numerous management factors influencing crop yield and environmental impact, [Wu et al., 2023] among which nitrogen (N) fertilization and irrigation are two of the most significant ones [Reddy et al., 2003]. Based on empirical experience and existing agricultural studies, local best management practices for N fertilization and irrigation exist among farmers, e.g., [Wright et al., 2022; Skhiri and Dechmi, 2012]. However, it remains to be seen whether the current management practices are optimal and whether these strategies perform well in the presence of changes in climate and market conditions. Thus, new methods are urgently needed to help farmers build cost-effective and readily deployable systems [Ara et al., 2021] that provide optimal management policies given a particular condition (including climate, yield price, management cost, etc.) and a target (e.g., maximum economic profit). Since reinforcement learning (RL) has already demonstrated extraordinary ability in solving tasks involving sequential decision making (SDM), e.g., in robotics and games [Kaufmann et al., 2018; Mnih et al., 2015], we see the great potential of RL in optimizing crop management, which is essentially an SDM problem. As numerous interactions between the RL agent and the environment are required for policy training, it is impossible to implement field trial-based methods. Therefore, using crop models to simulate the crop and environment and interact with the RL agent [Palmer et al., 2013; Attonaty et al., 1997] seems the only realistic solution.

Recently, the authors of [Wu et al., 2022] proposed to train management policies for N management using deep RL, the Decision Support System for Agrotechnology Transfer (DSSAT) [Jones et al., 2003], one of the most widely used crop models in the world, and Gym-DSSAT [Romain et al., 2022]. Their trained policies under full observations outperformed a baseline policy by achieving a higher yield or a similar yield with less N fertilizer input. However, there are a few limitations to this study. First, only N fertilization management was optimized. Also, only one reward function was adopted in the training and it is unclear whether their framework works well for various situations with different goals. Most importantly, their trained policies under full observation are not implementable in the real world as they need information not accessible by farmers, such as nitrate leaching and plant N uptake. Although the authors of [Wu et al., 2022] conducted experiments on policy training under partial...
observation, using only easily obtained or measurable state variables in reality, the training results could not outperform the baseline policy, let alone the ones trained under full observation. These limitations motivate the present paper.

In this paper, we present an intelligent crop management framework, depicted in Figure 1, that generates deployable and adaptable management policies based on RL, imitation learning (IL), crop simulations via DSSAT, and Gym-DSSAT. Our proposed framework combines RL on crop management, from [Wu et al., 2022], and IL on solving the problem of partial observation given an expert policy. We validate the performance of the proposed framework using the maize crop in Florida, US, and Zaragoza, Spain. For both cases, the trained policies outperform a baseline management policy following the convention of the Open AI Gym [Brockman et al., 2016] for russet potatoes. While these works prove the feasibility of deploying RL methods for crop management, the trained policies cannot outperform some baseline methods in the simulation. Recently, a more profound study on applying RL to train crop management policies was conducted by [Wu et al., 2022], which focused on N fertilization for maize in Florida and Iowa. The authors included a larger state space, and the trained policies under full observation achieved better outcomes than a baseline method. A more comprehensive review of research work in RL-based crop management can be found in [Gautron et al., 2022].

2.2 Crop Models for RL
As the real-world farm experiments are laborious, time-consuming, and cost-inefficient, crop models are becoming essential tools for estimating the crop responses to climate change and management practices on crop production [Zhao et al., 2019]. Until now, multiple crop simulation models have been developed for major crops [Bassu et al., 2014; Asseng et al., 2013]. According to [Jones et al., 2017], AP-SIM and DSSAT are the two most widely used crop models. In addition to providing accurate crop production estimates in relation to climate, genotype, soil, and management factors, these two crop models are open-source for the agricultural community and are updated constantly.

For currently available crop models, the users need to predefine the management practice before the start of a simulation. However, RL aims to find optimal policies that decide the management practices in (near) real-time according to the current weather, plant, and soil conditions. Given this concern, efforts have been made to enable real-time communication between an RL agent and the crop environment during a simulation. The authors of [Overweg et al., 2021] developed the CropGym environment for training fertilization policies following the convention of the Open AI Gym [Brockman et al., 2016], an environment widely used for developing and testing RL agents. Another interface based on the SIMPLE crop model for Open AI Gym is proposed in [Zhao et al., 2019] for russet potatoes. While these works prove the feasibility of deploying RL methods for crop management, the simulation models used are over-simplified without detailed information on the crop and environment. In contrast, Gym-DSSAT, based on the widely used DSSAT, is developed for training management policies with RL [Romain et al., 2022]. It enables daily interactions between an RL agent and the simulated crop environment in DSSAT and has been leveraged to optimize N management [Wu et al., 2022].

2.3 Imitation Learning for Policy Training
IL seeks to learn a policy by imitating the behavior of an expert and it has been widely applied in the field of robotics,
especially for autonomous vehicles. The authors of [Zhang and Cho, 2016] applied IL to learn autonomous driving policies using an expert driver as the learning target. The authors of [Zhang et al., 2016] alternatively used the model predictive controller (MPC) as the expert to train a control policy with limited observation. Both of these works indicate the promising ability of IL in learning good policies given an expert. [Zhang et al., 2016] further demonstrates the ability of IL in learning good policies under partial observation, which is usually difficult to achieve with standard RL due to the less available information compared with the full observation case. However, to the best of our knowledge, there has been no reported work on applying IL for crop management.

3 Methods

3.1 MDP Problem Formulation for Crop Management

Nitrogen fertilization and irrigation management are formulated as a finite MDP here. On each day $t$, the agent receives the state of the environment, $s_t$, and chooses the action $a_t$ from the action space $A$ based on some policy $\pi(s_t; \theta_1)$, where $\theta_1$ is the parameter of the policy at current day. $s_t$ contains information related to the weather, plant, and soil at given day from the simulator. $a_t$ consists of the amount of N fertilizer input, $N_t$, and the amount of irrigation water, $W_t$, to be applied on that day. Given $s_t$ and $a_t$, the reward $r_t(s_t, a_t)$ is defined as:

$$r_t(s_t, a_t) = \begin{cases} w_1 Y - w_2 N_t - w_3 W_t - w_4 N_{t,t} & \text{if harvest at } t, \\ -w_2 N_t - w_3 W_t - w_4 N_{t,t} & \text{otherwise}, \end{cases}$$

(1)

where $w_1, w_2, w_3, w_4, Y, N_{t,t}$ are four custom weight factors, yield at harvest, and the amount of nitrate leaching at given day, respectively. $Y$ and $N_{t,t}$ are variables from the state $s_t$. Given the state $s_t$ and a design of the reward function represented by $w_1, w_2, w_3, w_4$, the goal of the agent is to find the optimal policy $\pi(s_t; \theta_1)$ which renders $a_t$ and maximizes the future discounted return, which is defined as $R_t = \sum_{t=1}^{T} \gamma^{t-1} r_t$ representing the sum the reward at current day from applying $a_t$, and discounted future rewards with factor $\gamma$ following this policy.

3.2 Policy Training Using RL

Deep Q-network (DQN), a deep RL algorithm, is selected for agent training to learn the optimal policy that maximizes the future discounted return $R_t$. With DQN, a deep neural network is used to represent the action-value function, also known as the Q function [Mnih et al., 2015], and thus we call the network as a Q-network. The Q function of a policy $\pi$ is defined as $Q^\pi(s, a) = \mathbb{E}[R_t|s_t = s, a_t = a, \pi]$, which measures the expected future discounted return obtained from state $s$ by taking action of $a$ and following policy $\pi$ afterward. We keep updating the parameters of the Q-network to find the optimal Q function, $Q^*(s, a)$, which represents the optimal return that can be gained from state $s$ by taking action $a$ and following the optimal policy afterward. Given $Q^*$ and $s_t$, a greedy policy defined as $a_t^* = \max_{a \in A} Q^*(s_t, a)$ is used by the agent to determine the optimal action. Since the Q function determines the policy of the agent, training of the Q-network is the same as training of the policy. We update the parameter of the Q-network at iteration $i$, denoted by $\theta_i$, by minimizing the loss function:

$$L_i(\theta_i) = \mathbb{E}_{(s, a, r, s')} \left[ r + \gamma \max_{a' \in A} Q(s', a'; \theta_i) - Q(s, a; \theta_i) \right],$$

(2)

where $s, a, r, s'$ and $\gamma$ denote state, action, reward of $s$ and $a$, next state, and discount factor respectively, and $\theta_i^{-}$ denotes the parameters of a previously defined target network. The tuples $(s, a, r, s')$ are randomly chosen from the replay buffer, which is a memory base of previously generated tuples $(s, a, r, s')$ during training. The RL-trained policies under full observation are later used as the expert in IL.

3.3 Crop Simulations with Gym-DSSAT

Similar to [Wu et al., 2022], we leverage Gym-DSSAT [Romain et al., 2022], a Gym interface for DSSAT that enables the agent to interact with the simulated environment (i.e., reading the weather, soil, and crop information and applying management practices) on a daily basis. For more details about DSSAT and Gym-DSSAT, readers can refer to Section 2.2.

3.4 Policy Training Using IL

Imitation learning trains the agent to perform a task by mimicking the behavior of an expert [Osa et al., 2018]. As opposed to learning from scratch by trial-and-error in RL [Hussein et al., 2017], with IL the agent learns a mapping between the observations and desired actions determined by the expert, which simplifies the learning process of complex problems. For the crop management problem, not all state variables from the simulator can be achieved or measured by farmers. Thus, the agent should only utilize state variables that are accessible to farmers during real-world deployment. Given any state $s$, denote $s^o$ as the observable state which contains variables from $s$ that are observable or measurable by farmers. Under partial observation, on each day $t$, the agent receives $s^o_t$. The goal of the agent is to learn an optimal policy $\pi(s^o_t; \theta)$ that outputs an action $a^*_t$ identical to $a_t$, where $a_t$ is the action determined by the expert given $s_t$. Behavior cloning, the simplest form of IL, can be applied to train the policy under partial observation as follows. We first collect demonstrations, state-action pairs $(s, a)$, from the expert policy and store them into a dataset $D$. Then, the policy network of the agent is updated by minimizing the loss function

$$L(\theta) = \sum_{(s, a) \in D} \|\pi(s^o; \theta) - a\|.$$  

(3)

The loss function represents the difference between the output of the policy network with $s^o$ as the input, and the action $a$ determined by the expert policy given $s$.

4 Experiments and Results

Experiments on the training of $N$ and irrigation management policies for the maize crop under both full observation and partial observation were conducted with two case studies. The first case study uses the simulated environment of
Florida, US in 1982, and the second case study uses the simulated environment of Zaragoza, Spain in 1995. These two case studies aim to validate the feasibility and benefits of the proposed framework, not for immediate deployment. For real-world deployment of our framework, more details can be found in Section 5.2.

For each case study, we used DQN to train the RL agent under full observation, and the trained policies were then used as the experts to train management policies under partial observation using IL. We tested the performance of all the trained policies in simulation and compared them with baseline policies. The baseline policy for the Florida case study follows a maize production guide for farmers in Florida [Wright et al., 2022] written by domain experts, and the baseline policy for the Zaragoza case comes from the survey data on the real-world management practices of maize farmers in Zaragoza [Malik et al., 2019; Skhiri and Dechmi, 2012]. The detailed description of the baseline policies and the state variables of the full observation and partial observation from the simulator are omitted here due to space limits and will be included in the arXiv version of this paper.

### 4.1 Policy Training under Full Observation via RL

DQN was implemented for training the RL agent under full observation. We tested with five different reward functions to demonstrate the adaptability of our framework to different tradeoffs among crop yield, N fertilizer use, irrigation water use, and environmental impact.

#### Implementation Details

The neural network of the Q function was designed to have 3 hidden layers with 256 units in each layer. The discrete action space was set as:

$$A = \{40k \text{ kg ha}^{-1} \text{ N fertilizer} \& 6k \frac{L}{m^2} \text{ Irrigation water}\}$$

where $k = 0, 1, 2, 3, 4$, with a size of 25. This design of the action space includes standard amounts of N fertilizer and irrigation water that farmers can potentially apply in a single day and also provides enough options for finding good policies. The discount factor was set to be 0.99. For updating the neural network, we utilized Pytorch and Adam [Kingma and Ba, 2014] optimizer with an initial learning rate of 1e-5 and a batch size of 640.

For each case study, five different functions for $r_t$ from (1) were used to train the RL agent and for each reward function, we picked one trained policy for evaluation. The parameters used in each reward function (RF) are listed in Table 1. RF 1 represents the economic profit ($/ha) that farmers gain based on the approximate price of maize and cost of N fertilizer and irrigation water from [Mandrini et al., 2022] and [Wright et al., 2022]. RFs 2-4 indicate the economic profit under different situations. To be more specific, RF 2 represents the extreme case when irrigation water is free, RF 3 denotes the extreme case when the price of N fertilizer is free, and RF 4 simulates the situation when the price of N fertilizer is doubled. Compared with RFs 1-4 which consider economic profit only, RF 5 includes the additional term of nitrate leaching, an environmental factor. Nitrate leaching is unfavorable since it’s the major cause of environmental problems including eutrophication of watercourses and soil degradation [Di and Cameron, 2002]. RF 5 is designed with similar weights on yield, N fertilizer usage and irrigation usage, and a much larger weight on nitrate leaching to promote minimal nitrate leaching while obtaining a good economic profit.

#### Policy Evaluation

The evaluation results of the trained policies under full observation for both Florida and Zaragoza cases are shown in Table 2. It is worth mentioning that due to the random initialization of the Q-network and the fact that the Q-network gets updated every episode, the policies evaluated may not represent the best one from training. However, the trained policies we picked are representative enough to demonstrate the ability of RL to optimize crop management and the influence of the reward functions on the training results. According to Table 2, the reward function affects the strategy of the trained policy significantly for both case studies. Trained with RF 2 indicating the zero cost of irrigation water, Trained Policy 2 applies the largest amount of irrigation water while keeping the N input low, leading to the highest yield and the largest cumulative reward according to RF 2. Similarly, with a zero cost of N fertilizer in RF 3, Trained Policy 3 from both case studies applies a large amount of N fertilizer with a small amount of irrigation. With RF 4, indicating a doubled price for N fertilizer compared to RF 1, Trained Policy 4 results in a smaller amount of N input compared to Trained Policy 1. Most importantly, given an RF to compute the cumulative rewards of different trained policies, the largest reward is always achieved by the policy trained with this particular RF for both case studies (e.g., Trained policy 1 achieves the highest cumulative reward with RF 1), except the Florida case under RF 5, where Trained policy 5 achieves a cumulative reward slightly smaller than the largest one from Trained Policy 1 but still much larger than the one from the baseline policy. Compared with the baseline policies, the RL-trained policies achieve a 45% and a 55% increase in terms of profit (RF 1), and almost a 400% and a 30% increase in terms of RF 5 for the Florida case and Zaragoza case respectively. The enormous negative values of the cumulative rewards of Trained Policy 3 from both case studies are the results of the large amounts of N input, which are not punished during training with RF 3. In general, the results above demonstrate the ability of RL to optimize crop management under different criteria, at different locations with various environments. Given a specific target represented by a design of the reward function, we can always apply the RL-based training to find the optimal management policies.

<table>
<thead>
<tr>
<th>Note</th>
<th>RF 1</th>
<th>RF 2</th>
<th>RF 3</th>
<th>RF 4</th>
<th>RF 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economic profit</td>
<td>0.158</td>
<td>0.158</td>
<td>0.158</td>
<td>0.158</td>
<td>0.158</td>
</tr>
<tr>
<td>Free water</td>
<td>0.79</td>
<td>0.79</td>
<td>0</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Free N fertilizer</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Doubled N price</td>
<td>0.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>With N Leaching</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$w_1 (Y)$</th>
<th>$w_2 (N_t)$</th>
<th>$w_3 (W_t)$</th>
<th>$w_4 (N_{t,1})$</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF 1</td>
<td>0.158</td>
<td>0.79</td>
<td>1.1</td>
<td>0</td>
</tr>
<tr>
<td>RF 2</td>
<td>0.158</td>
<td>0.79</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RF 3</td>
<td>0.158</td>
<td>0</td>
<td>1.1</td>
<td>0</td>
</tr>
<tr>
<td>RF 4</td>
<td>0.158</td>
<td>1.58</td>
<td>1.1</td>
<td>0</td>
</tr>
<tr>
<td>RF 5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Weights used in each reward function (RF) defined by (1)
In the preceding studies, we optimized N fertilization and irrigation simultaneously and the results were compared with baseline policies recommended by domain experts. To examine the benefits of joint optimization of the N and irrigation management, we include additional experiments in this section where we trained management policies for only N fertilization or irrigation with the other practice following the baseline methods, e.g., training an N management policy while following the baseline policy for irrigation management. Experiments were conducted using RF 1 and RF 5 for the Florida case study. The results are shown in Table 3, which clearly demonstrates the benefits of joint optimization of N fertilization and irrigation management.

4.2 Policy Training under Partial Observation via IL

For RL-based policy training in Section 4.1, we included a large number of state variables from DSSAT as the input to the RL agent. However, most of the state variables used, including nitrate leaching and daily N denitrification, cannot be readily obtained or even measured by farmers with accessible instruments in reality. As a result, there is no way to implement those trained policies on a real farm. A policy that only utilizes state variables that can be easily obtained or measured (partial observation) is necessary for real-world deployment. In this section, we present a method based on IL to train management policies that only use readily available information for decision-making, in which the RL-trained policies under full observation in Section 4.1 are used as experts to be mimicked. All the state variables needed by the IL-trained policies can be easily accessed or measured by farmers in the real world, like the weather information, soil water content, which can be measured by a soil moisture meter, and maize growing stage, which can be easily identified by an experienced farmer.

Limited Performance of RL under Partial Observation

Under partial observation, we first experimented with RL-based policy training, using the same setup as used in the full-observation case (Section 4.1), except that the state space used here was smaller. Although we tried different hyperparameters, including dimensions of the neural network, learning rates, and batch sizes, all trained policies converged to a single one that applies zero N fertilizer and zero irrigation water every day. A typical training curve is shown in Figure 2, from which one can see that the cumulative reward under partial observation converges to a much smaller value compared with the full observation case. Thus, it is much more challenging to find a good policy under partial observation than under full observation. The unsatisfactory training results from RL under partial observation motivate us to leverage IL, a straightforward approach for policy training under partial observation given an expert. There are other commonly used techniques to deal with partial observation in RL like including the history of observations and actions or using predictive rewards [Muskardin et al., 2022]. However, they may require much more effort and work than applying IL.

Table 2: Evaluation results of the trained policies under full observation and the baseline policy for both case studies. $N_i$ represents the nitrate leaching amount. Trained Policy x indicates the policy trained using reward function (RF) x, and the values in the column of 'RF x' represent the cumulative rewards of each trained policy calculated with RF x. Details of each reward function can be found in Table 1. For each reward function design, the largest cumulative reward value is shown in bold.

<table>
<thead>
<tr>
<th>Florida Case</th>
<th>N Input (kg/ha)</th>
<th>Irrigation (L/m²)</th>
<th>$N_i$ (kg/ha)</th>
<th>Yield (kg/ha)</th>
<th>RF 1</th>
<th>RF 2</th>
<th>RF 3</th>
<th>RF 4</th>
<th>RF 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Policy</td>
<td>360</td>
<td>394</td>
<td>213</td>
<td>10772</td>
<td>984</td>
<td>1417</td>
<td>1269</td>
<td>700</td>
<td>338</td>
</tr>
<tr>
<td>Trained Policy 1</td>
<td>200</td>
<td>120</td>
<td>36</td>
<td>10852</td>
<td><strong>1425</strong></td>
<td>1557</td>
<td>1538</td>
<td>1267</td>
<td><strong>1673</strong></td>
</tr>
<tr>
<td>Trained Policy 2</td>
<td>200</td>
<td>732</td>
<td>59</td>
<td>11244</td>
<td>813</td>
<td><strong>1619</strong></td>
<td>971</td>
<td>655</td>
<td>1020</td>
</tr>
<tr>
<td>Trained Policy 3</td>
<td>19920</td>
<td>108</td>
<td>6205</td>
<td>10865</td>
<td>-1.4e4</td>
<td>-1.4e4</td>
<td><strong>1598</strong></td>
<td>-3.0e4</td>
<td>-4.9e4</td>
</tr>
<tr>
<td>Trained Policy 4</td>
<td>160</td>
<td>102</td>
<td>35</td>
<td>10358</td>
<td>1398</td>
<td>1510</td>
<td>1524</td>
<td><strong>1272</strong></td>
<td>1635</td>
</tr>
<tr>
<td>Trained Policy 5</td>
<td>200</td>
<td>138</td>
<td>39</td>
<td>10926</td>
<td>1417</td>
<td>1568</td>
<td>1575</td>
<td>1259</td>
<td>1651</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zaragoza Case</th>
<th>N Input (kg/ha)</th>
<th>Irrigation (L/m²)</th>
<th>$N_i$ (kg/ha)</th>
<th>Yield (kg/ha)</th>
<th>RF 1</th>
<th>RF 2</th>
<th>RF 3</th>
<th>RF 4</th>
<th>RF 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Policy</td>
<td>250</td>
<td>752</td>
<td>4</td>
<td>10990</td>
<td>712</td>
<td>1539</td>
<td>909</td>
<td>514</td>
<td>1176</td>
</tr>
<tr>
<td>Trained Policy 1</td>
<td>240</td>
<td>330</td>
<td>0</td>
<td>10477</td>
<td><strong>1103</strong></td>
<td>1466</td>
<td>1292</td>
<td>913</td>
<td>1525</td>
</tr>
<tr>
<td>Trained Policy 2</td>
<td>200</td>
<td>1068</td>
<td>6</td>
<td>10923</td>
<td>393</td>
<td><strong>1568</strong></td>
<td>551</td>
<td>235</td>
<td>888</td>
</tr>
<tr>
<td>Trained Policy 3</td>
<td>10640</td>
<td>324</td>
<td>0</td>
<td>10626</td>
<td>-7083</td>
<td>-6727</td>
<td><strong>1323</strong></td>
<td>-1.5e4</td>
<td>-8839</td>
</tr>
<tr>
<td>Trained Policy 4</td>
<td>120</td>
<td>336</td>
<td>0</td>
<td>9601</td>
<td>1053</td>
<td>1422</td>
<td>1147</td>
<td><strong>958</strong></td>
<td>1464</td>
</tr>
<tr>
<td>Trained Policy 5</td>
<td>200</td>
<td>390</td>
<td>0</td>
<td>10589</td>
<td>1086</td>
<td>1515</td>
<td>1244</td>
<td>928</td>
<td><strong>1528</strong></td>
</tr>
</tbody>
</table>

Table 3: Performance comparison of the trained policies on both N fertilization and irrigation with the trained policies on either N fertilization or irrigation. The largest values are shown in bold.

<table>
<thead>
<tr>
<th>Policy Evaluated</th>
<th>RF 1</th>
<th>RF 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline N Fertilization + Baseline Irrigation</td>
<td>984</td>
<td>338</td>
</tr>
<tr>
<td>Training N Fertilization + Training Irrigation</td>
<td><strong>1425</strong></td>
<td><strong>1673</strong></td>
</tr>
<tr>
<td>Baseline N Fertilization + Training Irrigation</td>
<td>1351</td>
<td>849</td>
</tr>
<tr>
<td>Training N Fertilization + Baseline Irrigation</td>
<td>1062</td>
<td>1217</td>
</tr>
</tbody>
</table>
Note that the policy to be trained here has a continuous action space, while RL-trained policies have a discrete action space. A continuous action space may lead to unrealistic results because the trained policy may decide to take actions frequently, applying a small amount of fertilizer and irrigation every day, which is impractical for farmers to follow. Thus, we evaluated the IL-trained policies by rounding the outputs of the network to the closest value in the action space $\mathcal{A}$ (4) used in RL training. The evaluation results of the trained policies from both case studies are shown in Table 4. According to Table 4, for the Florida case, the IL-Trained Policy 1 under partial observation achieves extremely similar results compared with the RL-Trained Policy 1, with only a negligible decrease of 1.4% in the cumulative reward, and the IL-Trained Policy 5 achieves exactly the same results as the RL-Trained Policy 5. For the Zaragoza case, both the trained policies under partial observation achieve a slightly larger cumulative reward than the RL-trained policies, which can be explained by the nature of IL (learning of the mapping between states and preferred actions) and further by the random initialization of the Q-network as we previously explained.

In conclusion, IL can successfully find optimal crop management policies under partial observation that achieve a much better outcome than the baseline policy in different locations with different goals.

### 5 Path to Deployment

It is plausible that the trained management policies, which work well in the DSSAT-simulated environment, may perform poorly in the real world, due to the uncertainty with weather and the mismatch between the crop models used to train the policies and the real cropping system. This is the well-known sim-to-real gap associated with transferring RL policies trained in simulators to the real world.

#### 5.1 Closing the Sim-To-Real Gap

To improve the robustness of the trained management policies, we will leverage domain and dynamics randomization.
techniques, proposed to mitigate the sim-to-real gap [Tobin et al., 2017; Peng et al., 2018]. More specifically, we will perturb selected key parameters of the model and randomize weather conditions when training the policy, which could “force” the trained policies to be robust against model and weather uncertainties. As our work focuses on presenting the framework of RL and IL on crop management and validating its performance, in the current paper we do not include experiments on these ideas, and we plan to have follow-up study addressing the robustness issue.

5.2 Deployment on a Real Farm

The IL-trained management policies for the partial observation case can be readily deployed on a real farm as all state variables used by the policies can be easily observed or measured with sensors. Soil and climate data corresponding to the farm for field tests need to be collected to configure the crop model within DSSAT, and shall be used for policy training. After training, on each day, given the current soil and weather information, the trained policies will make management decisions, i.e., how much N fertilizers and water to apply, and farmers or decision makers can then follow these decisions to apply the management practices. After harvest, based on the amount of water and fertilizer used, and the crop yield, the performance of the management policies can be evaluated. We are collaborating with the Department of crop science at the University of Illinois at Urbana-Champaign to conduct field experiments in the future to verify the efficacy of the proposed framework.

5.3 Policy Evaluation with Measurement Noises

When applying the IL-trained policies in reality, farmers need to obtain those observable states from the weather forecast and the measurement of the soil moisture. The collected data may not be the same as the actual values of the environment due to the inaccuracy of the forecast and the sensors. Considering this mismatch, we tested the IL-Trained Policy 1 from the Florida case study with measurement noises to mimic its performance in the real world. The details are as follows. On each day in simulation, we obtain the real state of the environment from the simulator and add random measurement noise to one or some key observable state variables. It is worth mentioning that values of the measurement noises are determined based on real-world accuracy data of the weather forecast and common soil moisture meters on market [Cobos and Devices, 2010; Floehr, 2010; Zhang et al., 2018; Heinemann et al., 2006]. For each measurement noise added, we evaluated the policy 400 times and recorded the average cumulative reward as well as the standard deviation. The results are shown in Table 5 and reveal that temperature and rainfall are the state variables that influence the performance most, while the rest have tiny effects. In addition, even with relatively large noises on several variables, the trained policy still achieves an average cumulative reward of 1185, which is 15.8% lower than the value without noise, but still much larger than the reward of the baseline policy. Thus, the IL-trained policies are able to achieve relatively good results compared with the baseline method even in the presence of measurement noises that mimic real-world scenarios.

<table>
<thead>
<tr>
<th>Variables with noise and noise level</th>
<th>RF 1 (STD)</th>
<th>Decrease (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Policy</td>
<td>984.4 (N/A)</td>
<td>N/A</td>
</tr>
<tr>
<td>No Noise</td>
<td>1407.7 (N/A)</td>
<td>0</td>
</tr>
<tr>
<td>Soil water content [+0.02]</td>
<td>1407.7 (0)</td>
<td>0</td>
</tr>
<tr>
<td>Soil water content [+0.05]</td>
<td>1405.0 (2.3)</td>
<td>0.2</td>
</tr>
<tr>
<td>Temperature [+1]</td>
<td>1392.0 (112.2)</td>
<td>1.1</td>
</tr>
<tr>
<td>Temperature [+2]</td>
<td>1136.2 (487.5)</td>
<td>13.9</td>
</tr>
<tr>
<td>Solar Radiation [+2%]</td>
<td>1408.3 (0.9)</td>
<td>0</td>
</tr>
<tr>
<td>Solar Radiation [+10%]</td>
<td>1408.0 (15.6)</td>
<td>0</td>
</tr>
<tr>
<td>Rain Fall [90% Accuracy]</td>
<td>1357.8 (233.1)</td>
<td>3.5</td>
</tr>
<tr>
<td>Leaf Area Index [+10%]</td>
<td>1404.0 (1.1)</td>
<td>0.3</td>
</tr>
<tr>
<td>Leaf Area Index [+20%]</td>
<td>1400.0 (5.1)</td>
<td>0.5</td>
</tr>
<tr>
<td>Soil water content [+0.02] + Temperature [+2]</td>
<td>1185 (421.5)</td>
<td>15.8</td>
</tr>
<tr>
<td>+ Solar Radiation [+2%]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Rain Fall [90% Accuracy]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ Leaf Area Index [+20%]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Performance of the IL-trained Policy 1 under measurement noises evaluated with RF 1. Noise levels are included in the square bracket in either strict value or percentage. The decrease (%) is calculated with respect to RF 1 of No Noise.

6 Conclusion

Finding the optimal crop management policy for N fertilization and irrigation is vital to achieving maximum yield while minimizing the management cost and environmental impact. In this paper, we present a framework for finding optimal management policies with deep reinforcement learning (RL), imitation learning (IL), and crop simulations based on DSSAT. Experiments are conducted for the maize crop in Florida, US, and Zaragoza, Spain, where both fertilization and irrigation are necessary for crop growth. Under full observation, i.e., with access to a large number of variables from the simulator, deep Q-network (DQN) is used to train management policies. Under partial observation, i.e., with access to a limited number of state variables that are observable or measurable in the real world, we use IL to train management policies by mimicking the behaviors of the RL-trained policies under full observation. Given variations in reward function designs, the RL-trained policies have different strategies during the decision-making to achieve maximum rewards, which indicates the adaptability of our proposed framework. Using IL, the trained policies under partial observation have almost identical decisions compared to the RL-trained policies under full observation, and all trained policies under both full and partial observations achieve better results compared with production guidelines for the maize crop in these two locations. Furthermore, even in the presence of measurement noises on the observable state variables, the IL-trained policies achieve much better results than the baseline methods, which paves the way for real-world deployment of our framework as they only need readily accessible information. Our future work includes improving the robustness of the trained policies in the presence of model mismatch and uncertain weather conditions. Since the IL-trained policies under partial observation are readily deployable, we are planning to test the efficacy of the proposed framework on a real farm.
Acknowledgements

This work was supported by the C3.ai Digital Transformation Institute and NSF under the RI grant #2133656.

References


