

Original papers

Grower-in-the-loop interactive reinforcement learning for greenhouse climate control

Maxiu Xiao^a, Jianglin Lan^b, Jingxin Yu^{a,c,d}, Weihong Ma^e, Qiuju Xie^f, Congcong Sun^a ^{*}

^a Agricultural Biosystems Engineering Group, Wageningen University, 6700 AA Wageningen, The Netherlands

^b James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, United Kingdom

^c National Engineering Research Center for Intelligent Equipment in Agriculture, Beijing, 100097, China

^d Research Center for Intelligent Equipment Technology, Beijing Academy of Agriculture and Forestry Sciences, Beijing, 100097, China

^e Information Technology Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing, 100097, China

^f College of Electrical and Information, Northeast Agricultural University, 150030 Harbin, China



ARTICLE INFO

Keywords:

Interactive reinforcement learning
Grower-in-the-loop learning
Greenhouse climate control
And agri-food production

ABSTRACT

Climate control is crucial for greenhouse production as it directly affects crop growth and resource use. Reinforcement learning (RL) has received increasing attention in this field, but still faces challenges, including limited training efficiency and high reliance on initial learning conditions. Interactive RL, which integrates human (grower) input with the RL agent's learning, offers a potential solution to overcome these challenges. However, interactive RL has not yet been applied to greenhouse climate control. Furthermore, human input is hardly perfect due to the complexity of climate control. The performance of interactive RL may also be limited by imperfect input. Therefore, this paper aims to explore the possibility and performance of applying interactive RL with imperfect inputs into greenhouse climate control, by: (1) developing three representative interactive RL algorithms tailored for greenhouse climate control (reward shaping, policy shaping and control sharing); (2) analyzing how input characteristics are often contradicting, and how the trade-offs between them make grower's inputs difficult to perfect; (3) proposing a neural network-based approach to enhance the robustness of interactive RL agents under limited input availability; (4) conducting a comprehensive evaluation of the three interactive RL algorithms with imperfect inputs in a simulated greenhouse environment. The demonstration shows that interactive RL incorporating imperfect grower inputs has the potential to improve the performance of the RL agent. Interactive RL algorithms that influence action selection, such as policy shaping and control sharing, perform better when dealing with imperfect inputs, achieving 8.4% and 6.8% improvement in profit, respectively. In contrast, reward shaping, an algorithm that manipulates the reward function, is sensitive to imperfect inputs and leads to a 9.4% decrease in profit. This highlights the importance of selecting an appropriate mechanism when incorporating imperfect inputs.

1. Introduction

In response to the threat that climate change poses to global food security, greenhouse production has emerged as a vital approach for mitigating risks and enhancing food production efficiency worldwide (Goddek et al., 2023; Vatistas et al., 2022). Within greenhouse production systems, climate control plays a critical role, as it directly influences plant growth by regulating key environmental factors such as temperature, humidity, and CO₂ concentration. However, climate control is also one of the most significant sources of energy consumption in greenhouse operations (Paris et al., 2022). Looking ahead, greenhouse horticulture faces several pressing challenges, including high energy demands (Wageningen Social and Economic Research, 2025),

as well as a shortage of skilled labor and experienced managers (Christiansen et al., 2020). Therefore, developing optimal and autonomous climate control systems is essential for improving crop productivity while minimizing energy and resource consumption.

Over the past years, various control approaches have been investigated (Van Straten et al., 2010; Robles Algarín et al., 2017; Mahmood et al., 2023). Among them, model predictive control (MPC) has been widely adopted due to its ability to handle the complexity of greenhouse systems (Mahmood et al., 2023; Zhang et al., 2022). MPC relies on a model of the greenhouse system (climate, crop, and economic) to predict its future behavior, and optimizes control strategies accordingly. This means the performance of MPC is heavily dependent on the

* Corresponding author.

E-mail address: congcong.sun@wur.nl (C. Sun).

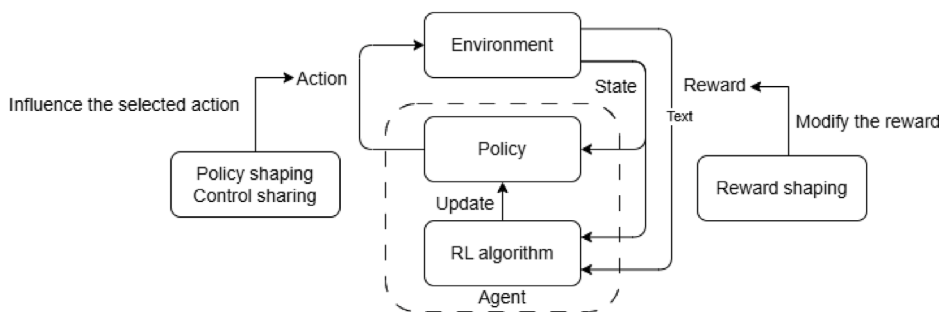


Fig. 1. Interactive RL framework.

accuracy of the model and the certainty of predictions. Studies that focus on dealing with the uncertainty in the model and prediction also exist. Methods like stochastic MPC (Kim and You, 2025) and robust MPC (Mahmood et al., 2023) are applied to solve it. However, MPC for greenhouse climate control still faces limitations, among which, the major one is that MPC is highly dependent on the provided model and has limited adaptability to changes in the greenhouse system without re-tuning the model.

Recently, reinforcement learning (RL) has attracted growing attention in greenhouse climate control. RL is a learning-based control method with adaptive capabilities. This enables RL to perform effectively in complex, error-prone, and uncertain environments (Lin et al., 2020; Meng et al., 2025), which are characteristic of greenhouse systems. As a learning-based, data-driven and adaptive approach, RL can continuously optimize control strategies to adapt to the dynamic system. Several studies have already evaluated RL in simulated greenhouse settings (Ajagekar et al., 2023; Morcego et al., 2023; Mansour et al., 2025), demonstrating its potential for achieving optimal climate control. Other studies have also explored the combination of RL and MPC to mitigate their individual weaknesses and improve overall performance (Msaad et al., 2025; Mallick et al., 2025).

Despite these promising results, RL applications in greenhouse climate management still face significant challenges. Among the most frequently mentioned limitations are low learning efficiency and limited robustness (Mansour et al., 2025). Moreover, current RL approaches often exclude growers from the control loop, failing to leverage their domain knowledge and practical experience, which could enhance learning efficiency. This lack of involvement may also lead to growers' reluctance to adopt RL or other artificial intelligence methods in real-world applications, as they have limited control or understanding of the decision-making process.

Interactive RL (Human-In-The-Loop RL) is a variant of RL that incorporates human input into the training process. In interactive RL, the RL agent learns through both interactions with the environment and human input. Fig. 1 provides a simplified illustration of how human input can be incorporated into the learning process. Methods like reward shaping modify the reward the RL agent receives, while methods like control sharing and policy shaping influence what action is selected by the RL agent.

Interactive RL has been tested on classic control tasks like Pac-Man (Griffith et al., 2013), MountainCar (Knox and Stone, 2012), and CartPole (Knox and Stone, 2012), as well as in more complicated environments like robotic arm control (Luo et al., 2025). These studies have demonstrated that interactive RL can surpass traditional RL in terms of both training efficiency and overall performance. However, to the best of our knowledge, interactive RL has not yet been explored in the context of greenhouse climate control. In addition, inputs in greenhouse climate control are difficult to achieve perfection, which could influence the performance of interactive RL (Kessler Faulkner et al., 2020; Cruz et al., 2018; Bignold et al., 2021).

Given this background, this paper aims to explore the feasibility and performance of applying interactive RL with imperfect inputs to

greenhouse climate control. The main contributions are summarized as follows:

- We developed three representative interactive RL algorithms tailored for greenhouse climate control: policy shaping, which uses grower advice on actions; control sharing, and reward shaping, both of which rely on grower's feedback regarding control performance.
- We analyzed how input characteristics are often contradicting, and how the trade-offs between them make grower's inputs difficult to perfect. We also discuss the potential shortcomings of different methods inputs can be provided.
- We proposed a neural network-based approach to enhance the robustness of interactive RL agents under limited input availability by incorporating additional neural networks to aggregate grower's input.
- We conducted a comprehensive evaluation and comparison of the three interactive RL algorithms in a simulated greenhouse environment, considering the potential variability and characteristics of actual grower input.

The remainder of the paper is organized as follows. Section 2 introduces the simulated greenhouse environment, the traditional RL algorithm (PPO), and an overview of interactive RL. Section 3 presents the characteristics of imperfect inputs and how they are simulated in this paper. Section 4 introduces the proposed interactive RL algorithms. The evaluation results are then presented and discussed in Section 5. Finally, Section 6 draws the conclusion as well as provides future work and recommendations.

2. Introduction of Greenhouse environment and RL

This section begins by introducing the simulated greenhouse environment employed in this paper. It then describes the traditional RL algorithm used, followed by an overview of the interactive RL framework.

2.1. Greenhouse environment

2.1.1. Greenhouse model and modification

This paper investigates a simulated lettuce greenhouse environment under winter weather conditions, as illustrated in Fig. 2, using the system dynamic model from Van Henten (1994). Details of the model are presented in Appendix. Although this model was developed thirty years ago, it remains the leading and pioneering model that has been applied in recent studies on greenhouse climate control (Morcego et al., 2023; Msaad et al., 2025; Mallick et al., 2025). Moreover, this model specializes in simulating greenhouse systems under winter weather conditions, presenting the challenge of balancing crop growth and resource usage.

Modification to the model input is made as follows: inputs (heating and CO₂ injection rate) are converted into setpoints using proportional-integral (PI) control. These two inputs are modified because providing

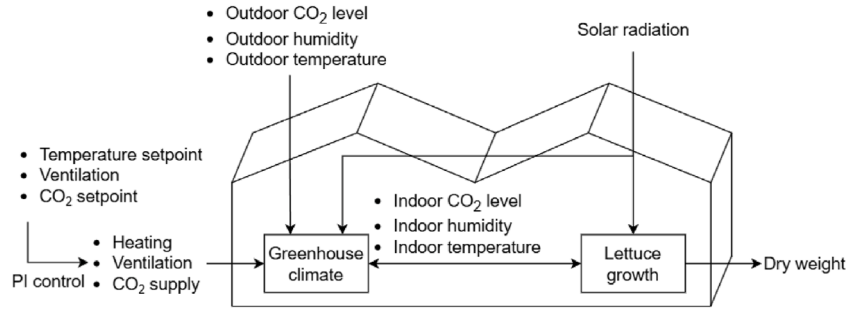


Fig. 2. Schematic diagram of the lettuce greenhouse environment (Van Henten, 1994).

setpoints as input is more intuitive than providing heating and CO₂ injection rate as input. The ventilation rate is not converted, as it passively influences all climate variables. Parameters used in the PI control are presented in the appendix.

2.1.2. Design of environment

To implement the modified model in the environment, the model is discretized using the Runge–Kutta fourth-order method (Higinio and Jesus, 2007) with a step size of 15 min. The state and action space of the greenhouse environment is summarized in Table 1. The state space used in the environment is similar to those used in other studies: time, all measurable climate variables, and crop information (dry weight) (Morcego et al., 2023; Msaad et al., 2025; Mallick et al., 2025). Three new states are also provided to the RL agent: current temperature setpoint, CO₂ setpoints, and ventilation rate. They are added due to modifications to the model, and setpoints are adjusted from their current value during control.

The action space of the environment is discrete, with three dimensions: adjusting temperature setpoints, adjusting CO₂ setpoints, and adjusting ventilation rate. This results in a total of 27 possible actions. The control interval of the environment is one hour, while the actuator settings (heating and CO₂ injection rate) are updated every 15 min using PI control. The action space of the environment is discrete. Using a continuous action space increases the difficulty of simulating feedback, as it requires careful tuning of an acceptable range for good action. This also complicates the comparison of different interactive RL algorithms, since a similar mechanism to the acceptable range is needed to simulate action advice. While continuous actions could be used to evaluate the influence of input accuracy, this is not considered in this paper.

The reward function (object function to be maximized) used is the economic profit of the greenhouse minus a penalty, formulated as follows:

$$\text{reward} = p_{\text{lettuce}} \cdot \Delta_{\text{dryweight}} - (p_{\text{CO}_2} \cdot u_{\text{CO}_2} + p_{\text{heat}} \cdot u_{\text{heat}}) - \text{penalty}, \quad (1)$$

where p_{lettuce} is the price of lettuce (16 Hfl/kg), $\Delta_{\text{dryweight}}$ is the change in lettuce dry weight (kg/m²), p_{CO_2} is the price of CO₂ supply (0.42 Hfl/kg), u_{CO_2} is the usage of CO₂ supply (kg/m²), p_{heat} is the price of heating (6.35×10^{-9} Hfl/J), u_{heat} is the usage of heating (J/m²), penalty is the cost if constraint violated at this step (5.24×10^{-3} Hfl/m²). Note that the price of lettuce, heating, and CO₂ are obtained from (Morcego et al., 2023). The penalty is applied when indoor climate constraints (Van Henten, 1994) (shown in Table 2) are violated. The economic profit is based on the economic profit indicator (EPI) used in similar studies (Morcego et al., 2023; Msaad et al., 2025), and is modified to use dry weight change and resource usage at each step to prevent sparse reward. As for the penalty, a fixed penalty is added whenever any constraint is violated. It is also feasible to dynamically adjust the penalty according to the degree of constraint violated (Morcego et al., 2023; Msaad et al., 2025). A fixed penalty is selected here for simplicity and ease of tuning.

The weather data used in the environment and episode length are further detailed in Section 5.1.

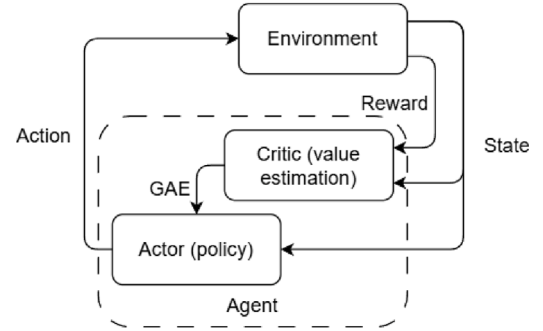


Fig. 3. Actor–critic architecture (Schulman et al., 2017).

2.2. Proximal policy optimization

This paper considers the Proximal Policy Optimization (PPO) (Schulman et al., 2017) as the baseline RL algorithm. PPO is selected because its policy is represented as a probability distribution, which facilitates influencing the RL agent's action selection.

PPO uses the actor–critic architecture shown in Fig. 3, where the critic estimates the value function of the state, and the actor represents the RL agent's policy as a probability distribution. During training, the critic estimates a baseline value for the actor's current policy. Policies that outperform current policy are encouraged, while those that under-perform are discouraged.

The level of encouragement or discouragement is based on the generalized advantage estimation (GAE) A_t , which estimates how much better or worse a policy is compared to the baseline estimation. A_t is computed by

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + (\gamma\lambda)^{T-t-1}\delta_{T-1} \\ = -v_\pi(s_t) + r_t + (\gamma\lambda)r_{t+1} + \dots + (\gamma\lambda)^{T-t}v_\pi(s_T), \quad (2)$$

where the GAE parameter λ combines multi-step δ_t to balance the trade-off between bias and variance. And δ_t is the temporal difference (TD) at time t , defined as

$$\delta_t = r_t + \gamma v_\pi(s_{t+1}) - v_\pi(s_t), \quad (3)$$

where r_t is the reward at time t , $v_\pi(s_t)$ is the value of s_t under policy π , γ is the discount factor.

2.3. Interactive RL backgrounds

2.3.1. Overview of interactive RL

Traditional RL faces many challenges, including low learning efficiency (Dulac-Arnold et al., 2021), which requires the RL agent to have a large amount of interactions with the environment before learning an effective policy. This makes RL hard to apply to real-world problems, where the environment is complicated. Furthermore, human

Table 1
State and action space of the greenhouse environment.

State space		Action space		
Name		Name	Dimension	Action
Outdoor temperature	°C	Temperature setpoint	°C	Temperature setpoint
Outdoor humidity	%	Ventilation rate	mm/s	–2 °C, 0, +2 °C
Outdoor CO ₂	ppm	CO ₂ setpoint	ppm	–0.5 mm/s, 0, +0.5 mm/s
Indoor temperature	°C	Hour of the day		–200 ppm, 0, +200 ppm
Indoor humidity	%	Radiation	W/m ²	
Indoor CO ₂	ppm	Dry weight	kg/m ²	

Table 2
Constraints for indoor climate.

Indoor climate variable	Lower constraint	Upper constraint
Temperature (°C)	6.5	40
CO ₂ level (ppm)	0	1500
Relative humidity (%)	0	90

involvement is typically limited to the design of the reward function in traditional RL. For real-world problems, human knowledge and experience usually exist and can help the RL agent to learn more efficiently. Therefore, interactive RL has been developed to incorporate these knowledge and experience into the RL learning process.

In the development of interactive RL, several basic approaches have been proposed to incorporate different types of input (feedback and advice) in different stages of the RL agent's learning. Details of these approaches are provided in the next section. RL has been successfully applied to simple game-like environments (Pacman, Gridworld) (Griffith et al., 2013; Knox and Stone, 2012). It has also been applied to realistic tasks like robot arm control (Luo et al., 2025) and autonomous driving (Wu et al., 2023a). These studies have demonstrated that interactive RL can outperform traditional RL in both learning efficiency and overall performance.

Besides applying interactive RL to more domains, recent studies on interactive RL also focus on topics like how the input's characteristics influence the performance and how to use the input more efficiently. How the input's characteristics influence the performance is further detailed in Section 3.1. As for improving the efficiency of using human input, human involvement should be less frequent, less costly, and more informative. To achieve this, new functions that estimate human input can be added (Liu et al., 2023), and methods from active learning have been adopted (Kessler Faulkner et al., 2019).

2.3.2. Interactive RL approaches

In interactive RL, human inputs can be incorporated into the learning process through three approaches (Lin et al., 2020; Arzate Cruz and Igarashi, 2020): reward shaping, policy shaping, and guided exploration. As shown in Table 3, these three approaches incorporate different types of input at different stages.

Reward shaping. This approach modifies the reward function of the RL agent to incorporate additional sources of information. A seminal contribution to this line of work is the TAMER framework proposed by Knox and Stone (2008), which has been widely adopted in interactive RL. In TAMER, the RL agent learns a feedback function $F(s, a)$ that estimates human-provided feedback, rather than learning the Q-function. Building on this foundation, the authors later introduced the DQN-TAMER framework (Knox and Stone, 2010), which incorporates both human feedback and environment-generated rewards. In their study, they evaluated eight different methods for combining these two sources of information. Among these, reward shaping emerged as an intuitive and effective strategy, and can be formally expressed as follows:

$$r'(s, a) = r(s, a) + \beta \cdot F(s, a), \quad (4)$$

where the environment-generated reward $r(s, a)$ is reshaped by adding estimated feedback $F(s, a)$ with a weight factor β .

Policy shaping. This approach influences the action selection process of the RL agent by combining the agent's policy and input's policy (Griffith et al., 2013). This approach can be formulated as:

$$P(s, a) = \begin{cases} 1 - \beta, & a = \pi_{\text{agent}}(s, a) \\ \beta, & a = \frac{\pi_{\text{agent}}(s, a) \cdot \pi_{\text{input}}(s, a)}{\sum_{a' \in \mathcal{A}} \pi_{\text{agent}}(s, a') \cdot \pi_{\text{input}}(s, a')}, \end{cases} \quad (5)$$

where $P(s, a)$ is the final probability of selecting action a at state s , $\pi_{\text{agent}}(s, a)$ is the probability of selecting action a at state s under the RL agent's policy, and $\pi_{\text{input}}(s, a)$ is the probability of selecting action a at state s under the input's policy.

Eq. (5) defines a mixed policy: with probability $1 - \beta$, the RL agent relies on its own policy, and with probability β , the RL agent relies on a mixed policy that combines both the RL agent's and input's policy. Since this requires the probabilities of each action in policy $\pi_{\text{agent}}(s)$ to be presented explicitly, it is typically applied to policy-based algorithms with stochastic policies (PPO, A2C, etc.).

Guided exploration and control sharing. In contrast to policy shaping, guided exploration overrides the action selected by the RL agent instead of reshaping the agent's policy. This makes it applicable to both value-based algorithms and policy-based algorithms. Control sharing is one of the eight approaches evaluated in the DQN-TAMER framework (Knox and Stone, 2010) and belongs to the category of guided exploration. It is formulated as Eq. (6). With probability $1 - \beta$, the agent relies on its own policy, and with probability β , the agent's policy is overridden by a greedy policy that maximizes the estimated feedback.

$$P(a) = \begin{cases} 1 - \beta, & a = \pi_{\text{agent}}(s, a) \\ \beta, & a = \arg \max_a (F(s, a)). \end{cases} \quad (6)$$

3. Grower's input

Although interactive RL has the potential to outperform traditional RL, it relies on high-quality input to do so (Bignold et al., 2021). This section begins by outlining the types of input used in interactive RL and the characteristics these inputs must possess. It then examines how, in the context of greenhouse climate control, these desirable characteristics often conflict with one another, creating trade-offs that make it challenging for growers to provide optimal input. Finally, the section details the method employed in this paper to simulate imperfect inputs.

3.1. Human input in interactive RL

In interactive RL, human inputs can be categorized into two types: feedback and action advice (Arzate Cruz and Igarashi, 2020). Feedback includes both binary and scalar-valued feedback. These two types of feedback relate to the action selected by the RL agent. Binary feedback simply labels the action as "good" or "bad", whereas scalar-valued feedback provides a quantitative evaluation of the action. In contrast, action advice specifies either the optimal action or the probability distribution over all possible actions. Beyond the form of information, the scope of input also differs: feedback pertains only to the selected action, while action advice can provide information about the entire action space. In interactive RL, the quality of input is crucial for

Table 3
The three approaches of interactive RL.

Approach	Type of Input	Adoption Stage	Example
Reward shaping	Feedback	Reward function	Knox and Stone (2010)
Policy shaping	Action advice	Action selection	Griffith et al. (2013), Cederborg et al. (2015)
Guided exploration	Feedback/Action advice	Action selection	Knox and Stone (2010)

Table 4
Characteristics of human input (Bignold et al., 2021).

Characteristic	Description
Accuracy	How appropriate inputs are to the current situation.
Availability	The availability of human teachers.
Concept drift	The goals or understanding of the environment shift over time.
Reward bias	The teaching style of the human teachers, e.g., encouragement and punishment.
Cognitive bias	The difference between the RL agent and the human teacher's goal.
Knowledge level	Knowledge of the environment or information available.
Latency	Delay in providing inputs.

successive learning (Bignold et al., 2021). The quality of input can be assessed by seven characteristics (Bignold et al., 2021), summarized in Table 4.

3.2. Characteristics of grower's input

In greenhouse climate control, grower's inputs are difficult to achieve perfection due to the trade-offs between these four key characteristics: availability, cognitive bias, latency, and knowledge level.

Availability. It refers to the accessibility and willingness of human experts to provide input. While RL agents are often trained in simulated greenhouse environments to reduce time and costs, this setup demands a large volume of input within a short period. Effective greenhouse climate control requires domain knowledge and experience, making it difficult for an RL expert to provide as good input as real growers. Additionally, given the repetitive and time-consuming nature of providing inputs, growers may find the task abstract, tedious, and exhausting, which diminishes their willingness to participate. As a result, querying growers at every step of the training process is impractical, leading to low input availability. To address this issue, two main strategies are commonly used: applying predefined simple rules or leveraging existing grower knowledge or data through expert systems or imitation learning. However, both approaches may introduce cognitive bias and often lack the depth and nuance of real expert input, thereby reducing the overall knowledge level of the guidance provided.

Cognitive bias. It refers to the misalignment between the RL agent's objective of maximizing final profit and the grower's actual decision-making priorities. Complex and time-consuming decision-making for providing input is required to achieve RL agent's objective. This reduces input availability when frequent grower involvement is needed. To improve availability, existing grower knowledge or data (e.g., expert rules) is often used. But this introduces greater cognitive bias, as the greenhouse the RL agent is trained on might differ from the greenhouse in the data source. Additionally, growers may prioritize crop physiology over profit due to the long production cycles and economic uncertainties in greenhouse operations. This risk-averse approach aims to ensure high yield and quality, indirectly supporting profitability. And even when grower input targets profit, it often lacks the adaptability of dynamic optimization, limiting its effectiveness under uncertain and changing conditions (Van Straten et al., 2010).

Latency. It refers to delays in providing input. Latency is particularly problematic for action advice, as it must be delivered during the RL agent's action selection process. Input with minimal cognitive bias can lead to increased latency. Long growing cycles in greenhouse production mean that the final profit (RL agent's objective) is only known at the end of the cycle, resulting in inherently high latency. To reduce latency, inputs may instead focus on short-term gains or crop physiology. However, this shift introduces greater cognitive bias.

Knowledge level. It refers to the grower's understanding of the greenhouse system and the information available for decision-making. Inputs based on high knowledge levels can reduce cognitive bias but often require more time and effort to provide, which decreases availability and increases latency. Conversely, inputs derived from low knowledge levels, such as predefined rules that set allowable ranges for climate variables, offer high availability but tend to exhibit greater cognitive bias.

3.3. Simulation of growers' imperfect inputs

In this paper, three types of simulated inputs are designed, as detailed in Table 5. Both precise action advice and constraint advice are forms of action advice that provide probabilities for each action, but they differ in knowledge level. As for feedback, only feedback with a high knowledge level is included. Feedback with a low knowledge level is excluded, as using feedback to enforce constraints (ranges) is a common practice in RL.

All three types of simulated inputs are based on pre-extracted knowledge and inherently contain cognitive bias. They are categorized into low and high knowledge levels. Inputs with a low knowledge level specify suitable ranges (constraints) for climate variables, as detailed in Table 6. In contrast, inputs with a high knowledge level provide optimal climate variable values aiming at maximizing current crop growth. These optimal values are pre-calculated and approximated using interpolation to facilitate faster input generation.

Fig. 4 illustrates examples of the three input types, focusing solely on temperature. In constraint advice, actions that keep the temperature within the specified range are assigned equal probabilities of selection. Precise action advice sets the temperature as close as possible to the optimal setpoint. Feedback is provided after the outcome of the action (S_{t+1}) is observed and evaluates whether the resulting temperature is closer to the optimal value compared to the potential effects of alternative actions.

4. Interactive RL for Greenhouse climate control

This section begins by explaining how the three types of inputs described in Section 3.3 are incorporated into PPO. It then introduces a neural network-based approach to enhance the robustness of interactive RL agents when input availability is limited. Finally, it describes the proposed interactive RL algorithms.

4.1. Incorporating grower's inputs

Reward shaping, control sharing, and policy shaping are applied in this work to cover both forms of human input (advice and feedback) and the two incorporating stages of input (reward function and action

Table 5
Three types of grower's simulated inputs.

Characteristics	Feedback	Precise action advice	Constraint advice
<i>Input form</i>	Binary (bad action: -1, good action: 1)	Probabilities for action	Probabilities for action
<i>Cognitive bias</i>	Maximize current crop growth	Maximize current crop growth	Maximize current crop growth
<i>Knowledge level</i>	High (consider the optimal values of climate variables)	High (consider the optimal values of climate variables)	Low (consider the suitable range of climate variables)

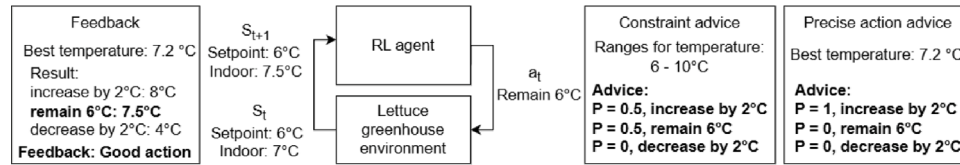


Fig. 4. Examples of simulated grower giving input.

Table 6
Ranges of climate conditions for constraint advice.

Indoor climate variable	Daytime range	Nighttime range
CO2 setpoint (ppm)	400 – 1500	300 – 600
Temperature setpoint (°C)	10 – 25	6 – 10

selection). Reward shaping and policy shaping are selected because they respectively represent one type of input form and one stage to incorporate, as shown in Section 2.3.2. And both of these approaches are intuitive. Control sharing, as a form of guided exploration, is quite similar to policy shaping in PPO, since PPO uses a stochastic policy. Control sharing is still selected because it allows direct influence on action selection through feedback. Although less intuitive, control sharing may be more effective than reward shaping (Knox and Stone, 2010), as reward shaping modifies the reward signal and only affects the policy indirectly.

The reward shaping and control sharing used in this work are adapted from the DQN-TAMER framework (Knox and Stone, 2008, 2010). Reward shaping modifies the reward according to (4), while control sharing influences action selection by overriding the RL agent's policy according to (6). Both approaches employ a weighting factor β to regulate the degree of input involvement. Additionally, an auxiliary neural network (F function) is introduced to estimate feedback in both approaches. The F function takes state variables as input and outputs estimated feedback values for each action.

Policy shaping (Cederborg et al., 2015) used in this work incorporates both types of action advice. It influences action selection according to (5). Additionally, a new neural network, π_{grower} , is introduced to estimate the action advice. π_{grower} takes state variables as input and outputs estimated probability for each action.

4.2. Addressing limited availability

In all three algorithms, an additional network (F function or π_{grower}) is used to estimate the input. This network not only facilitates the control sharing method but also enhances the algorithm's tolerance to limited input availability and high latency. This capability is especially important when inputs must be obtained by querying growers at every step or when providing input demands significant time and effort.

With the additional network, the goal of providing inputs is similar to active learning: maximize the learning efficiency of input estimation with minimal input given. As in active learning, two sampling approaches exist: pool-based and stream-based. This paper only focuses on the pool-based approach for two reasons: First, the pool-based approach has better sample efficiency (Cacciarelli and Kulahci, 2024), which suits the needs of limited input availability better. Second, the stream-based approach requires immediate decisions (one chance per step), increasing complexity and potentially the burden of the grower.

Algorithm 1 The proposed interactive RL algorithm

```

1: Initialize PPO agent
2: Initialize  $F$  function ( $\pi_{\text{grower}}$ ),  $\pi_{\text{error}}$  (for the selective strategy), and buffer  $D_{\text{input}}$ 
3: Set hyperparameters:  $\beta$ ,  $n$ , and  $N$ 
4: while not reach total timesteps do
5:   while not reach update interval do
6:     if using policy shaping then
7:       Interact with the environment using policy from (5) with estimated action advice
8:     else if using control sharing then
9:       Interact with the environment using policy from (6) with estimated feedback
10:    else if using reward shaping then
11:      Interact with the environment using the agent's policy and modify the reward using (4) with estimated feedback
12:    end if
13:  end while
14:  Select  $n$  steps from the rollout buffer to provide inputs (random or  $n$  steps with the maximum estimated error)
15:  Add selected steps and provided input to  $D_{\text{input}}$ 
16:  Update the PPO agent
17:  Update the  $F$  function ( $\pi_{\text{grower}}$ ) and  $\pi_{\text{error}}$  (if exists) using  $D_{\text{input}}$  for  $N$  iterations
18: end while

```

As in active learning, it is essential to efficiently select which steps to provide input. Within pool-based approaches, two selection strategies are compared in this paper: a random strategy and a selective strategy. The random strategy samples input steps randomly, while the selective strategy is inspired by SafeDagger (Zhang and Cho, 2016). In the selective strategy, a new neural network π_{error} is introduced to estimate the discrepancy ($error$) between the provided input and the estimation of F function or π_{grower} . Inputs are provided only to steps with high estimated $error$.

4.3. Interactive RL implementation

The implemented interactive RL algorithms are detailed in Algorithm 1 and Fig. 5. Modifications to the original PPO framework for incorporating grower inputs and addressing limited input availability are emphasized in bold in Algorithm 1. In addition to the introduction of auxiliary neural networks, a new buffer, D_{input} , is implemented to store data (state, action, input, and $error$) for updating these auxiliary networks. Unlike the rollout buffer used in PPO, D_{input} is bigger and

Table 7
Hyperparameters for all RL agents.

Hyperparameters	Values	Hyperparameters	Values
Learning rate	1×10^{-4}	Actor and critic size	512 \times 4
Step numbers per update	2048	Gamma (discount factor)	0.97
Batch size	256	Total training steps	500,000

Table 8
Hyperparameters used in interactive RL agents.

Hyperparameters	Values
Sizes of π_{grower} and π_{error} networks	256 \times 3
Number of inputs n	2048/1024/512/256/128
Initial input weight factor β	0.5/0.2/0.1/0.05
Number of iterations N	500
Entropy coefficient	1×10^{-2} (no inputs, reward shaping, policy shaping(constraint))/ 1×10^{-3} (policy shaping(precise), control sharing)
Learning rate for π_{grower} network	1×10^{-4} (policy shaping(precise))/ 1×10^{-3} (reward shaping, control sharing)
Learning rate for π_{error} network	1×10^{-3}

Table 9
Average cumulative reward on the test environment.

Methods	$\beta=0.05$	$\beta=0.1$	$\beta=0.2$	$\beta=0.5$
baseline (Original PPO)			1.91	
PPO with feedback only			0.71	
Policy shaping (RL agent input)	2.01	1.94	1.96	1.94
Policy shaping (precise)	2.00	1.95	2.07	1.98
Policy shaping (constraint)	1.92	1.97	1.95	1.87
Control sharing	1.97	2.01	2.04	1.89
Reward shaping	1.73	1.68	1.69	1.72

easily balances human input with the RL agent's interaction with the environment, and enables easy experimental comparison.

In this work, β is linearly decreased. Studies (Knox and Stone, 2010, 2012) show that decreasing β along the learning progress leads to good performance of the RL agent. But in these studies, β is decreased exponentially by multiplying it by a factor less than one at the end of each episode. The exponential decrease is replaced with a linear decrease, as the RL agent is trained on more episodes, and β should not decrease too fast.

5.3. Performance of interactive RL

Table 9 shows the average cumulative reward of interactive RL algorithms in the test environment. In all experiments, the availability of inputs is full. Additionally, two more simulations are added: original PPO using only simulated feedback as the reward function, and policy shaping using a trained RL agent to provide input. Original PPO using only simulated feedback as the reward function reduces the test cumulative reward to only 0.71 Hfl/m². This is expected and indicates that the simulated inputs are far from perfect. In policy shaping using a trained RL agent to provide input, the RL agent with the best test performance (2.21 Hfl/m²) is used. This approach improves the performance across all β values. The policy shaping using precise action advice performs best in this environment at $\beta=0.1$, achieving 8.4% improvement. All algorithms except reward shaping improve the cumulative reward, indicating that reward shaping is more sensitive to the quality of input. It is also noticeable that when using $\beta=0.5$, control sharing and policy shaping with constraint advice perform worse than the baseline.

Fig. 6 further compares the performance of interactive RL algorithms to the baseline. In this figure, only β with the highest performance improvement is selected. Except for reward shaping, all interactive RL algorithms maintain a similar harvested lettuce dry

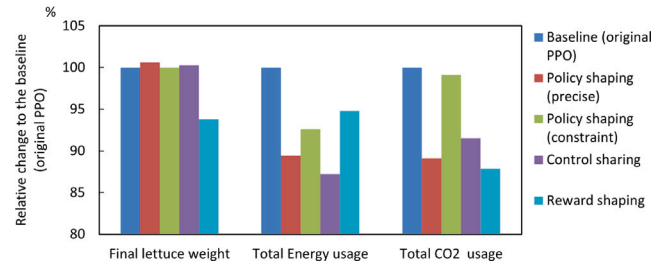


Fig. 6. Performance of interactive RL algorithms relative to baseline (original PPO).

weight while reducing energy and CO₂ usage. Policy shaping with precise advice and control sharing each achieves about 10% reductions in energy and CO₂ usage. Policy shaping with constraint advice only reduces its energy usage by less than 10%, while its CO₂ usage is only slightly reduced compared with the baseline. In contrast, reward shaping lowers both energy and CO₂ usage but also reduces the harvested dry weight, resulting in a 9.4% decrease in profit relative to the baseline.

Fig. 7 shows the average setpoints and indoor climate variables for each hour of the day. Also, only β with the highest performance improvement is selected in this figure. All interactive RL algorithms, except for reward shaping, show a similar pattern. They maintain lower temperature setpoints and consequently lower indoor temperatures than the baseline. In contrast, reward shaping shows a different pattern. It maintains low temperature setpoints and CO₂ setpoints at night. This is because it receives positive feedback for doing so, without extra cost. During the day, it also maintains low temperature setpoints and CO₂ setpoints, due to the conflict between the reward and the feedback.

Notably, indoor temperatures and CO₂ setpoints in Fig. 7 are generally higher than the corresponding setpoint values, and high indoor CO₂ and indoor values are observed at night. This is due to several factors: First, the greenhouse environment has no actuator that can actively reduce temperature or CO₂ level. Second, in the later stage of the growing cycle, CO₂ produced by the plant's respiration also increases. Third, due to the high violation constraints on CO₂ and temperature at night, and relative humidity remains within constraints, the agent is not motivated to use ventilation to lower these variables.

The comparisons of interactive RL algorithms suggest that approaches that influence the agent's action selection (policy shaping and reward shaping) can improve the RL agent's performance, even

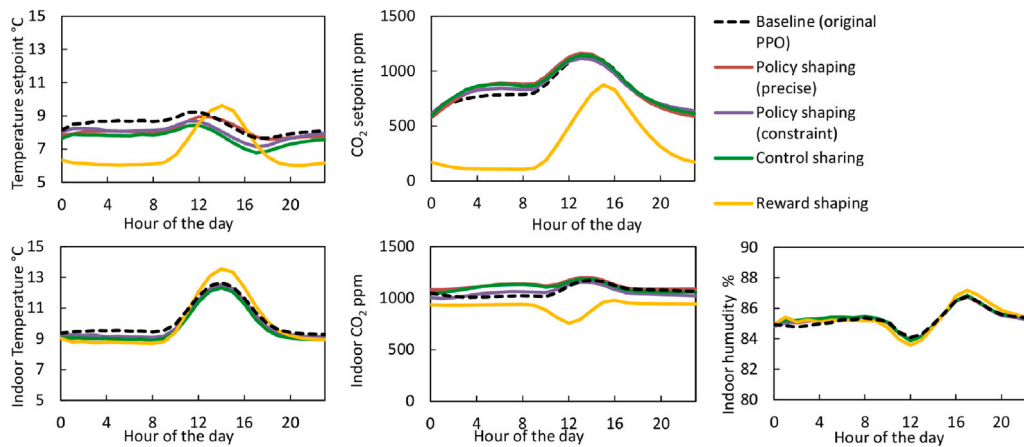


Fig. 7. Average setpoints and indoor climate variables over a day.

with imperfect inputs. Additionally, it is worth noting that under certain β values, using imperfect inputs can lead to better performance than using input provided by RL agent. This is probably because PPO uses GAE to improve its policy. As shown in (2), GAE is a type of advantage function that calculates how much better or worse taking an action is compared to $v_{\pi}(s)$. Both policy shaping and control sharing incorporate inputs by influencing the action selection of the RL agent. Since using feedback as the reward function reduces cumulative rollout rewards, $v_{\pi}(s)$ can be underestimated when incorporating imperfect inputs, leading to higher GAE estimates, which can accelerate learning. Also, an underestimated $v_{\pi}(s)$ improves exploration (Kobayashi, 2025), which can help the RL agent explore a better policy.

In contrast, reward shaping follows the feedback's underlying policy and fails to improve the performance of the RL agent. Reward shaping faces the conflict between the reward and the feedback. Manipulating reward functions with imperfect feedback may misguide the learning process (Knox and Stone, 2012, 2010). Therefore, making it more sensitive to imperfect inputs than other algorithms.

5.4. Impact of input weight factor

The input weight factor β significantly influences the performance of interactive RL algorithms. This section focuses on examining its impact in the context of policy shaping with precise action advice. Control sharing, policy shaping with constraint advice, and policy shaping with an RL agent provide advice are not presented here, as they influence the RL agent's action selection, and the effect of β on them exhibits similar trends.

Fig. 8 illustrates the evolution of rollout and test rewards during training. The rollout reward refers to the cumulative reward obtained during the RL agent's training and reflects the quality of actions taken by the RL agent during the training process. As expected, a higher β accelerates the increase in rollout rewards during the early stages of training. However, this does not correspond to early improvement of test rewards; instead, it results in lower test performance in the early stages of training. Moreover, an excessively high initial β (e.g., $\beta = 0.5$) leads to slightly reduced rollout rewards in the later stages of training, likely due to the way PPO leverages the advantage function to update its policy. In contrast, using a moderately low initial β (β in the range of 0.05 to 0.2) yields improvements in both rollout and test rewards.

This result shows that proper β is crucial for policy shaping with precise advice. Only using a proper range of β can improve performance. A high value of β slows down learning and degrades performance, while a small value of β accelerates learning and improves performance. This is probably also due to PPO using GAE as well.

Fig. 9 shows how β influences the performance. In the early stage of training, the RL agent struggles to find a policy better than the input. As β increases, the RL agent takes good actions (input) more frequently, and the critic overestimates $v_{\pi}(s)$. As shown in Fig. 8A, increasing β leads to higher rollout rewards in the early stages of training. This suggests that the RL agent selects actions with high rewards more frequently, and $v_{\pi}(s)$ can be overestimated. As $v_{\pi}(s)$ is overestimated, GAE estimates decrease, and the policy learns more slowly. Therefore, a too high β accelerates the increase of rollout rewards but does not translate to high test rewards. On the other hand, when β is low, although $v_{\pi}(s)$ is slightly overestimated and the policy learns slightly slower, good actions (input) still have a bigger chance of being selected. Therefore, the policy may learn slightly slower to slightly faster (depending on β).

In the later stage of training, the RL agent has already learned a policy better than the input. In this case, incorporating inputs leads to an underestimation of $v_{\pi}(s)$, which makes the policy learns faster. With a low β , the agent can still explore and exploit the environment based on its own policy to find a better policy. However, with a high β , the agent has a smaller chance of selecting actions based on its exploration. Therefore, good actions might be missed, and the policy learns more slowly.

5.5. Impact of input availability

Figs. 10 and 11 show the test loss of F function (π_{grower}) during the training process. In training, each update interval consists of 2048 steps. When 1024 inputs are provided, this means that only half of the steps can receive inputs. The test data is generated by allowing both a well-trained and a poorly trained RL agent to interact with the test environment. Simulated inputs corresponding to these interactions are then collected to form the test dataset. Including both well-trained and poorly trained agents ensures a more balanced and representative test set. In both figures, the time steps are limited to 400,000 steps instead of 500,000 steps. As estimated inputs are not utilized during the final 100,000 steps, the F function (π_{grower}) is no longer updated during this period.

Fig. 10 depicts the test loss of the F function (control sharing) during training. A reduction in the number of inputs leads to a significant increase in the loss of the F function. Additionally, the selective strategy does not reduce the loss compared to the random strategy, except when 256 inputs are used per update interval. Fig. 11 presents the test loss of π_{grower} (policy shaping) during training. At the initial time step, the loss is 2.23 and then decreases rapidly. Across all three cases, the selective strategy either does not increase the loss (for 1024 and 512 inputs) or only slightly increases it (for 256 inputs) during the early stages of training. Moreover, the selective strategy consistently

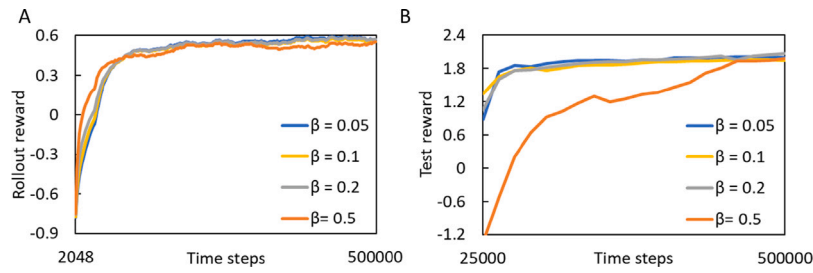


Fig. 8. Impact of the β for policy shaping with precise advice. A: rollout reward, B: test reward.

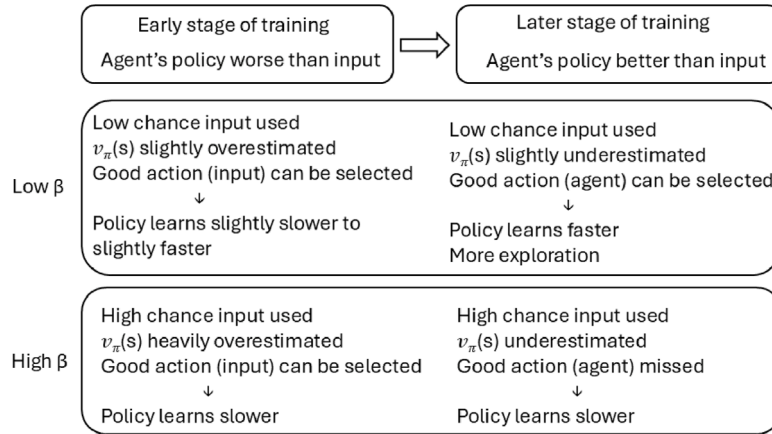


Fig. 9. Impact of the β for policy shaping with precise advice.

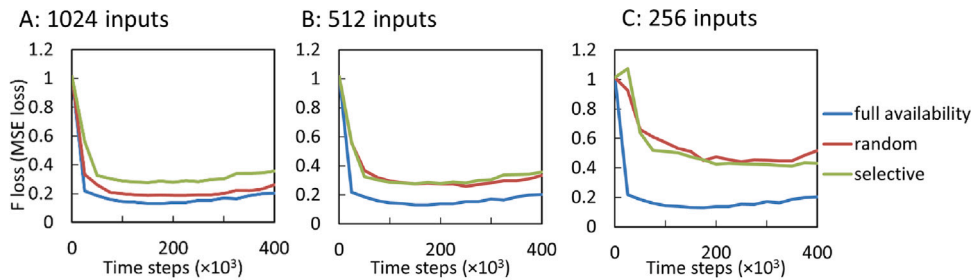


Fig. 10. Changes in the average loss of F function (control sharing).

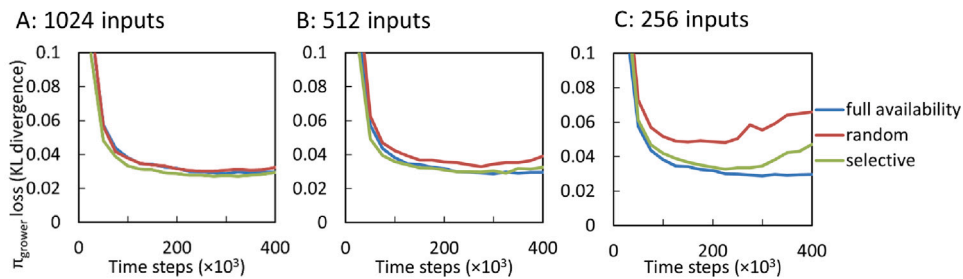


Fig. 11. Changes in the average loss of π_{grower} (policy shaping).

results in a lower loss of π_{grower} compared to the random strategy across all input availabilities. It is also noteworthy that the loss increases during the later stages of training in the cases with 512 and 256 inputs per update interval, regardless of the strategy employed.

These results indicate that limited input availability leads to different effects on the two types of inputs. For feedback under control sharing, neither the selective nor the random strategy is effective in maintaining a low loss of the F function. In contrast, for action advice,

the selective strategy consistently outperforms the random strategy in reducing the loss of π_{grower} . This distinction suggests that action advice is less sensitive to limited input availability than feedback. The effectiveness of the selective strategy in the context of action advice further implies that selecting inputs based on estimated discrepancy yields more informative updates, thereby enhancing the learning efficiency of π_{grower} . Additionally, the increase in loss observed during the later stages of training in Fig. 11C suggests that π_{grower} may be overfitting, which could hinder its generalization to unseen data. Future work may address this issue by incorporating regularization techniques or applying early stopping to improve generalization performance.

6. Conclusion

Interactive RL incorporates human (grower) input with the RL agent's learning to improve the agent's overall performance. However, it has not yet been applied to greenhouse climate control and faces challenges of imperfect inputs. This work investigated the possibility and performance of applying interactive RL with imperfect inputs in greenhouse climate control. The key contributions and findings are as follows: (1) Three interactive RL algorithms, policy shaping, control sharing, and reward shaping, are proposed and evaluated in a simulated greenhouse environment with simulated imperfect inputs. Simulation results indicate that interactive RL, even with imperfect grower inputs, can enhance agent performance, provided the approach to incorporate inputs is appropriately selected. As a consequence of using GAE in training PPO agent, approaches influencing action selection (policy shaping and control sharing) are more robust to the incorporation of imperfect input, when the level of incorporation is properly defined. (2) This work analyzes the trade-off between key characteristics of imperfect input, including availability, cognitive bias, latency, and knowledge level. The drawbacks of strategies such as pre-extracting grower knowledge and direct grower interaction are also discussed. (3) A neural network-based method to address limited input availability is proposed. This method uses a neural network to estimate inputs and convert the input-providing process to one similar to pool-based active learning. Test results indicate that action advice is more robust to limited availability than feedback, particularly when using the discrepancy-based selection strategy.

Despite the contributions, this work has certain limitations that can be addressed in future studies. Only PPO is selected as the baseline RL algorithm. As discussed, GAE used in PPO might be the reason why policy shaping and control sharing are robust to imperfect inputs and sensitive to the input weight factor β . However, this may not be true for other RL algorithms. The performance of RL algorithms without using GAE may not be improved, which is not explored in this work. Also, in the DQN-TAMER framework (Knox and Stone, 2012), a too-high β leads to worse performance, but a higher level of input incorporation should be tolerated. Another limitation is that

Future work can also explore interactive RL in continuous action spaces.

Additionally, future studies can focus on understanding grower's control strategy and their decision-making process through data-driven approaches, like inverse RL. This not only helps provide better inputs and improve RL agent's reward design, but also enables growers to gain a clearer understanding of their intentions and strategies, thereby achieving more effective control of greenhouse climate.

CRedit authorship contribution statement

Maxiu Xiao: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jianglin Lan:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization. **Jingxin Yu:** Writing – review & editing, Validation, Methodology, Conceptualization. **Weihong Ma:** Writing – review &

editing, Validation, Methodology, Formal analysis, Conceptualization. **Qiuju Xie:** Writing – review & editing, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Congcong Sun:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

The lettuce greenhouse model is described by the following equations:

$$\begin{aligned}\frac{dx_1(t)}{dt} &= p_{1,1}\phi_{\text{phot},c}(t) - p_{1,2}x_1(t)2^{x_3(t)/10-5/2}, \\ \frac{dx_2(t)}{dt} &= \frac{1}{p_{2,1}}(-\phi_{\text{phot},c}(t) + p_{2,2}x_1(t)2^{x_3(t)/10-5/2} + u_1(t)10^{-6} - \phi_{\text{vent},c}(t)), \\ \frac{dx_3(t)}{dt} &= \frac{1}{p_{3,1}}(u_3(t) - (p_{3,2}u_2(t)10^{-3} + p_{3,3})(x_3(t) - d_3(t)) + p_{3,4}d_1(t)), \\ \frac{dx_4(t)}{dt} &= \frac{1}{p_{4,1}}(\phi_{\text{transp},h}(t) - \phi_{\text{vent},h}(t)),\end{aligned}$$

with

$$\begin{aligned}\phi_{\text{phot},c}(t) &= (1 - \exp(-p_{1,3}x_1(t)))[p_{1,4}d_1(t)(-p_{1,5}x_3(t)^2 + p_{1,6}x_3(t) - p_{1,7}) \\ &\quad \times (x_2(t) - p_{1,8})]/\varphi(t), \\ \varphi(t) &= p_{1,4}d_1(t) + (-p_{1,5}x_3(t)^2 + p_{1,6}x_3(t) - p_{1,7})(x_2(t) - p_{1,8}), \\ \phi_{\text{vent},c}(t) &= (u_2(t)10^{-3} + p_{2,3})(x_2(t) - d_2(t)), \\ \phi_{\text{vent},h}(t) &= (u_2(t)10^{-3} + p_{2,3})(x_4(t) - d_4(t)), \\ \phi_{\text{transp},h}(t) &= p_{4,2}(1 - \exp(-p_{1,3}x_1(t))) \\ &\quad \times \left(\frac{p_{4,3}}{p_{4,4}(x_3(t) + p_{4,5})} \exp\left(\frac{p_{4,6}x_3(t)}{x_3(t) + p_{4,7}}\right) - x_4(t) \right),\end{aligned}$$

$$u_1(t) = p_{5,1}(u_{CO_2}(t) - y_2(t)) + p_{5,2} \int_{t-1 \text{ hour}}^t (u_{CO_2}(\tau) - y_2(\tau)) d\tau,$$

$$u_3(t) = p_{5,3}(u_{\text{temp}}(t) - y_3(t)) + p_{5,4} \int_{t-1 \text{ hour}}^t (u_{\text{temp}}(\tau) - y_3(\tau)) d\tau,$$

where the meaning of variables is shown in Table 10. Furthermore, $\phi_{\text{phot},c}(t)$, $\phi_{\text{vent},c}(t)$, $\phi_{\text{transp},h}(t)$ and $\phi_{\text{vent},h}(t)$ are the gross canopy photosynthesis rate, mass exchange of CO_2 through the vents, canopy transpiration and mass exchange of H_2O through the vents, respectively. The measurement equation is defined as

$$y_1(t) = 10^3 \cdot x_1(t) \quad \text{g m}^{-2},$$

$$y_2(t) = \frac{10^6 \cdot p_{2,4}(x_3(t) + p_{2,5})}{p_{2,6}p_{2,7}} \cdot x_2(t), \quad \text{ppm},$$

$$y_3(t) = x_3(t), \quad ^\circ\text{C},$$

$$y_4(t) = \frac{10^2 \cdot p_{2,4}(x_3(t) + p_{2,5})}{11 \cdot \exp\left(\frac{p_{4,8}x_3(t)}{x_3(t) + p_{4,9}}\right)} \cdot x_4(t), \quad \%,$$

The model parameters $p_{i,j}$ are defined and presented in Table 11.

Data availability

Data will be made available on request.

Table 10
Meaning of the lettuce greenhouse model parameters.

$x_1(t)$	dry weight (kg/m ²)	$d_1(t)$	radiation (W/m ²)
$x_2(t)$	indoor CO ₂ (kg/m ³)	$d_2(t)$	outdoor CO ₂ (kg/m ³)
$x_3(t)$	indoor temperature (°C)	$d_3(t)$	outdoor temperature (°C)
$x_4(t)$	indoor humidity (%)	$d_4(t)$	outdoor humidity (kg/m ³)
$u_1(t)$	CO ₂ injection (mg/m ² /s)	$y_1(t)$	dry-weight (kg/m ²)
$u_2(t)$	ventilation (mm/s)	$y_2(t)$	indoor CO ₂ (ppm)
$u_3(t)$	heating (W/m ²)	$y_3(t)$	indoor temperature (°C)
$u_{CO_2}(t)$	CO ₂ setpoint (ppm)	$y_4(t)$	indoor humidity (%)
$u_{temp}(t)$	temperature setpoint (°C)		

Table 11
Model parameter values (Van Henten, 1994).

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$p_{1,1}$	0.544	$p_{2,1}$	4.1	$p_{3,1}$	3·10 ⁴	$p_{4,1}$	4.1	$p_{5,1}$	0.05
$p_{1,2}$	2.65 ·10 ⁻⁷	$p_{2,2}$	4.87 ·10 ⁻⁷	$p_{3,2}$	1290	$p_{4,2}$	0.0036	$p_{5,2}$	3 ·10 ⁻⁶
$p_{1,3}$	53	$p_{2,3}$	7.5 ·10 ⁻⁶	$p_{3,3}$	6.1	$p_{4,3}$	9348	$p_{5,3}$	55
$p_{1,4}$	3.55 ·10 ⁻⁹	$p_{2,4}$	8.31	$p_{3,4}$	0.2	$p_{4,4}$	8314	$p_{5,4}$	2.5 ·10 ⁻²
$p_{1,5}$	5.11 ·10 ⁻⁶	$p_{2,5}$	273.15			$p_{4,5}$	273.15		
$p_{1,6}$	2.3 ·10 ⁻⁴	$p_{2,6}$	101 325			$p_{4,6}$	17.4		
$p_{1,7}$	6.29 ·10 ⁻⁴	$p_{2,7}$	0.044			$p_{4,7}$	239		
$p_{1,8}$	5.2 ·10 ⁻⁵					$p_{4,8}$	17.269		
						$p_{4,9}$	238.3		

References

- Ajagekar, A., Mattson, N.S., You, F., 2023. Energy-efficient ai-based control of semi-closed greenhouses leveraging robust optimization in deep reinforcement learning. *Adv. Appl. Energy* 9, 100119.
- Arzate Cruz, C., Igarashi, T., 2020. A survey on interactive reinforcement learning: Design principles and open challenges. In: *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. pp. 1195–1209.
- Bignold, A., Cruz, F., Dazeley, R., Vamplew, P., Foale, C., 2021. An evaluation methodology for interactive reinforcement learning with simulated users. *Biomimetics* 6 (1), 13.
- Cacciarelli, D., Kulahci, M., 2024. Active learning for data streams: a survey. *Mach. Learn.* 113 (1), 185–239.
- Cederborg, T., Grover, I., Isbell, Jr., C.L., Thomaz, A.L., 2015. Policy shaping with human teachers. In: *IJCAI*. pp. 3366–3372.
- Christiaensen, L., Rutledge, Z., Taylor, J.E., 2020. The future of work in agriculture: Some reflections. (9193), World Bank Policy Research Working Paper.
- Cruz, F., Magg, S., Nagai, Y., Wermter, S., 2018. Improving interactive reinforcement learning: What makes a good teacher? *Connect. Sci.* 30 (3), 306–325.
- Dulac-Arnold, G., Levine, N., Mankowitz, D.J., Li, J., Paduraru, C., Gowal, S., Hester, T., 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Mach. Learn.* 110 (9), 2419–2468.
- Goddek, S., Körner, O., Keesman, K.J., Tester, M.A., Lefers, R., Fleskens, L., Joyce, A., van Os, E., Gross, A., Leemans, R., 2023. How greenhouse horticulture in arid regions can contribute to climate-resilient and sustainable food security. *Glob. Food Secur.* 38, 100701.
- Griffith, S., Subramanian, K., Scholz, J., Isbell, C.L., Thomaz, A.L., 2013. Policy shaping: Integrating human feedback with reinforcement learning. *Adv. Neural Inf. Process. Syst.* 26.
- Higinio, R., Jesus, V.-A., 2007. A fourth-order runge-kutta method based on BDF-type Chebyshev approximations. *J. Comput. Appl. Math.* 204 (1), 124–136.
- Huang, S., Dossa, R.F.J., Raffin, A., Kanervisto, A., Wang, W., 2022. The 37 implementation details of proximal policy optimization. In: *ICLR Blog Track*. URL <https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/>.
- Kessler Faulkner, T., Gutierrez, R.A., Short, E.S., Hoffman, G., Thomaz, A.L., 2019. Active attention-modified policy shaping: socially interactive agents track. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. pp. 728–736.
- Kessler Faulkner, T.A., Schaertl Short, E., Thomaz, A.L., 2020. Interactive reinforcement learning with inaccurate feedback. In: *2020 IEEE International Conference on Robotics and Automation. ICRA*, pp. 7498–7504. <http://dx.doi.org/10.1109/ICRA40945.2020.9197219>.
- Kim, J., You, F., 2025. Energy-efficient greenhouse climate control using Gaussian process-based stochastic model predictive control. *Appl. Energy* 391, 125841.
- KNMI, 2025. KNMI - Hourly weather data in the Netherlands. 12.02.2025. URL <https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens>.
- Knox, W.B., Stone, P., 2008. Tamer: Training an agent manually via evaluative reinforcement. In: *2008 7th IEEE International Conference on Development and Learning. IEEE*, pp. 292–297.
- Knox, W.B., Stone, P., 2010. Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In: *AAMAS*, vol. 10, pp. 5–12.
- Knox, W.B., Stone, P., 2012. Reinforcement learning from simultaneous human and MDP reward. In: *AAMAS*, vol. 1004, Valencia, pp. 475–482.
- Kobayashi, T., 2025. Intentionally-underestimated value function at terminal state for temporal-difference learning with mis-designed reward. *Results Control. Optim.* 18, 100530.
- Lan, X., Tans, P., Thoning, K., 2025. Trends in globally-averaged CO2 determined from NOAA Global Monitoring Laboratory measurements. 2025. 14-Mar. <http://dx.doi.org/10.15138/9N0H-ZH07>.
- Lin, J., Ma, Z., Gomez, R., Nakamura, K., He, B., Li, G., 2020. A review on interactive reinforcement learning from human social feedback. *IEEE Access* 8, 120757–120765. <http://dx.doi.org/10.1109/ACCESS.2020.3006254>.
- Liu, S., Wu, C., Li, Y., Zhang, L., 2023. Boosting feedback efficiency of interactive reinforcement learning by adaptive learning from scores. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE*, pp. 7561–7567.
- Luo, J., Xu, C., Wu, J., Levine, S., 2025. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Sci. Robot.* 10 (105), eads5033.
- Mahmood, F., Govindan, R., Bermak, A., Yang, D., Al-Ansari, T., 2023. Data-driven robust model predictive control for greenhouse temperature control and energy utilisation assessment. *Appl. Energy* 343, 121190. <http://dx.doi.org/10.1016/j.apenergy.2023.121190>, URL <https://www.sciencedirect.com/science/article/pii/S03062619230005548>.
- Mallick, S., Airaldi, F., Dabiri, A., Sun, C., De Schutter, B., 2025. Reinforcement learning-based model predictive control for greenhouse climate control. *Smart Agric. Technol.* 10, 100751. <http://dx.doi.org/10.1016/j.atech.2024.100751>, URL <https://www.sciencedirect.com/science/article/pii/S2772375524003551>.
- Mansour, M., Sathyanarayanan, K.K., Sauerteig, P., Streif, S., 2025. Adaptive robust greenhouse climate control: Combining deep reinforcement learning and economic optimization. *Smart Agric. Technol.* 12, 101327. <http://dx.doi.org/10.1016/j.atech.2025.101327>, URL <https://www.sciencedirect.com/science/article/pii/S2772375525005581>.
- Meng, Y., Liu, C., Zhao, J., Huang, J., Jing, G., 2025. Stackelberg game-based anti-disturbance control for unmanned surface vessels via integrative reinforcement learning. *Intell. Robot.* 5 (1), 88–104.
- Morcego, B., Yin, W., Boersma, S., Van Henten, E., Puig, V., Sun, C., 2023. Reinforcement learning versus model predictive control on greenhouse climate control. *Comput. Electron. Agric.* 215, 108372.
- Msaad, S., Harraway, M., McAllister, R.D., 2025. RL-guided MPC for autonomous greenhouse control. *arXiv preprint arXiv:2506.13278*.
- Paris, B., Vadorou, F., Balafoutis, A.T., Vaiopoulos, K., Kyriakarakos, G., Manolakos, D., Papadakis, G., 2022. Energy use in greenhouses in the EU: A review recommending energy efficiency measures and renewable energy sources adoption. *Appl. Sci.* 12 (10), 5150.
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N., 2021. Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.* 22 (268), 1–8.
- Robles Algarin, C., Callejas Cabarcas, J., Polo Llanos, A., 2017. Low-cost fuzzy logic control for greenhouse environments with web monitoring. *Electronics* 6 (4), <http://dx.doi.org/10.3390/electronics6040071>, URL <https://www.mdpi.com/2079-9292/6/4/71>.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

- Singi, S., He, Z., Pan, A., Patel, S., Sigurdsson, G.A., Piramuthu, R., Song, S., Ciocarlie, M., 2024. Decision making for human-in-the-loop robotic agents via uncertainty-aware reinforcement learning. In: 2024 IEEE International Conference on Robotics and Automation. ICRA, pp. 7939–7945. <http://dx.doi.org/10.1109/ICRA57147.2024.10611425>.
- Towers, M., Kwiatkowski, A., Terry, J., Balis, J.U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al., 2024. Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint [arXiv:2407.17032](https://arxiv.org/abs/2407.17032).
- Van Henten, E., 1994. *Greenhouse Climate Management: an Optimal Control Approach*. Wageningen University and Research.
- Van Straten, G., van Willigenburg, G., van Henten, E., van Ooteghem, R., 2010. *Optimal Control of Greenhouse Cultivation*. CRC Press.
- Vatistas, C., Avgoustaki, D.D., Bartzanas, T., 2022. A systematic literature review on controlled-environment agriculture: How vertical farms and greenhouses can influence the sustainability and footprint of urban microclimate with local food production. *Atmosphere* 13 (8), <http://dx.doi.org/10.3390/atmos13081258>, URL <https://www.mdpi.com/2073-4433/13/8/1258>.
- Wageningen Social and Economic Research, 2025. Agro & food portal. 12.02.2025. URL <https://agrimatie.nl/SectorResultaat.aspx?subpubID=2232§orID=2240>.
- Wu, J., Huang, Z., Hu, Z., Lv, C., 2023a. Toward human-in-the-loop AI: Enhancing deep reinforcement learning via real-time human guidance for autonomous driving. *Engineering*.
- Wu, J., Huang, Z., Hu, Z., Lv, C., 2023b. Toward human-in-the-loop AI: Enhancing deep reinforcement learning via real-time human guidance for autonomous driving. *Engineering* 21, 75–91. <http://dx.doi.org/10.1016/j.eng.2022.05.017>, URL <https://www.sciencedirect.com/science/article/pii/S2095809922004878>.
- Zhang, J., Cho, K., 2016. Query-efficient imitation learning for end-to-end autonomous driving. arXiv preprint [arXiv:1605.06450](https://arxiv.org/abs/1605.06450).
- Zhang, M., Yan, T., Wang, W., Jia, X., Wang, J., Klemeš, J.J., 2022. Energy-saving design and control strategy towards modern sustainable greenhouse: A review. *Renew. Sustain. Energy Rev.* 164, 112602. <http://dx.doi.org/10.1016/j.rser.2022.112602>, URL <https://www.sciencedirect.com/science/article/pii/S1364032122004981>.