Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deep Reinforcement Learning for Adaptive Temperature Control in Greenhouses: A DQN-Based Framework and Experimental Evaluation

Priyanshi Kumari
kalinga institute of industrial technology
Hamed laouz
University of Biskra
Shahana Bano
s.bano@rgu.ac.uk

Robert Gordon University

Research Article

Keywords: Reinforcement Learning, Deep Q-Network, Greenhouse climate control, Autonomous greenhouse, Precision agriculture, Energy efficiency

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11563502/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Deep Reinforcement Learning for Adaptive Temperature Control in Greenhouses: A DQN-Based Framework and Experimental Evaluation

Priyanshi Kumari[1], Hamed laouz[2], Shahana Bano[3*]

[1]School of Computer engineering, kalinga institute of industrial technology, Bhubaneswar, Odisha, 751024 India.
[2]Computer science department, University of Biskra, Biskra, 07000 Algeria.
[3*]School of Computing, Engineering and Technology, Robert Gordon University, Aberdeen, United Kingdom.

*Corresponding author(s). E-mail(s): s.bano@rgu.ac.uk;
Contributing authors: 2305797@kiit.ac.in; laouz.hamed@univ-biskra.dz;

Abstract

The integration of artificial intelligence into controlled environment agriculture represents a transformative opportunity for addressing global food security challenges while promoting sustainable resource utilisation. This paper presents a comprehensive Reinforcement Learning (RL) framework for optimizing greenhouse temperature management, leveraging data from the second edition of the Autonomous Greenhouse Challenge. We formulate the climate control problem as a Markov Decision Process (MDP) and implement a Deep Q-Network (DQN) agent that learns to regulate temperature through discrete actions: heating, cooling, or maintaining the current state. Our methodology incorporates a sophisticated reward function that balances multiple objectives, including maintaining optimal growing conditions, minimizing energy consumption, and ensuring temperature stability. The agent is trained and evaluated on a subset of the 2019 challenge dataset, specifically utilizing measurements from the Reference team's compartment to ensure data consistency and reliability. Experimental results demonstrate that the RL agent effectively learns an adaptive control policy, achieving an average comfort zone occupancy of 40.1% across diverse starting

conditions while demonstrating appropriate action selection based on environmental state. The agent exhibits particularly strong performance when initialised within the optimal temperature range, achieving 100% comfort zone occupancy in multiple evaluation episodes. This research contributes to the growing body of knowledge on AI-driven agricultural automation by providing a reproducible methodology, comprehensive performance analysis, and insights into the practical challenges of implementing RL systems in real-world greenhouse environments. The findings suggest that RL approaches offer a viable path toward autonomous greenhouse management with potential benefits for crop yield optimization and energy efficiency.

Keywords: Reinforcement Learning, Deep Q-Network, Greenhouse climate control, Autonomous greenhouse, Precision agriculture, Energy efficiency.

1 Introduction

The twenty-first century presents unprecedented challenges to global food systems. With the world population projected to reach 9.7 billion by 2050, agricultural productivity must increase substantially while simultaneously reducing environmental impact [1]. Conventional open-field agriculture faces numerous constraints, including climate variability, water scarcity, land degradation, and the need for intensive chemical inputs. Controlled Environment Agriculture (CEA), particularly greenhouse cultivation, offers a promising alternative by providing a protected environment where crop growth parameters can be precisely managed [2].

Greenhouses create microclimates that protect crops from adverse weather conditions, extend growing seasons, and enable year-round production of high-value crops such as tomatoes, cucumbers, and bell peppers [3]. However, the operation of modern greenhouses involves complex management of multiple environmental variables including temperature, humidity, CO2 concentration, light intensity, and irrigation schedules. The interdependencies among these variables, combined with external weather influences and crop physiological responses, create a challenging control problem that traditional methods struggle to address optimally [4].

Temperature regulation is particularly critical in greenhouse operations, directly influencing photosynthesis, transpiration, fruit development, and overall crop yield [5]. Maintaining optimal temperature ranges requires significant energy inputs, particularly for heating during cold periods or cooling during warm periods. Energy costs typically represent 30-40% of operational expenses in commercial greenhouse operations, creating strong economic incentives for efficient climate control strategies [6]. Furthermore, the increasing volatility of energy prices and growing environmental concerns about carbon emissions underscore the importance of developing energy-efficient control approaches [7].

Reinforcement Learning (RL) has emerged as a powerful paradigm for addressing complex sequential decision-making problems where optimal solutions are difficult to derive analytically [8]. Unlike supervised learning, which requires labeled examples of desired outputs, RL agents learn through interaction with an environment, receiving

rewards or penalties based on their actions and gradually improving their decision-making policies [9]. This trial-and-error learning approach is particularly well-suited for greenhouse climate control, where the system dynamics are complex, partially observable, and difficult to model accurately.

The application of RL to greenhouse control has gained increasing attention in recent years. Early research demonstrated the feasibility of using Q-learning and other tabular RL methods for greenhouse climate management, though these approaches were limited by state space dimensionality [10]. The advent of Deep Reinforcement Learning (DRL), which combines RL with deep neural networks, has significantly expanded the capabilities of RL systems by enabling them to handle high-dimensional state spaces and learn complex policies directly from raw sensor data [11].

Deep Q-Networks (DQNs), introduced by Mnih et al., represent a breakthrough in DRL by using neural networks to approximate Q-value functions and employing techniques such as experience replay and target networks to stabilize learning [12]. Subsequent developments, including Double DQN, Dueling DQN, and Prioritized Experience Replay, have further improved the performance and stability of DQN-based systems [13]. These advances make DQNs a compelling choice for greenhouse control applications, where the trade-off between exploration and exploitation and the need for stable learning are critical considerations.

This paper presents a novel application of Deep Reinforcement Learning for adaptive temperature control in greenhouse environments. We formulate the temperature regulation problem as a Markov Decision Process and implement a DQN agent that learns to make discrete control decisions based on real-time environmental observations. The proposed framework incorporates a multi-objective reward function that balances crop growth optimization, energy efficiency, and temperature stability, enabling the agent to learn nuanced control policies that adapt to varying environmental conditions. The primary contributions of this work could be summarized as follows:

1. **Reinforcement Learning Framework for Greenhouse Temperature Control:** We propose an end-to-end Deep Reinforcement Learning framework for autonomous greenhouse temperature management. The framework models climate control as a Markov Decision Process and employs a Deep Q-Network (DQN) agent to learn adaptive heating and cooling policies directly from greenhouse environmental observations.
2. **Multi-Objective Temperature Control Strategy:** We design a reward function that simultaneously promotes optimal crop growing conditions, energy-efficient operation, and temperature stability, enabling the agent to balance multiple competing control objectives during policy learning.
3. **Benchmark Evaluation on the Autonomous Greenhouse Challenge Dataset:** We develop a reproducible experimental pipeline using the publicly available Autonomous Greenhouse Challenge dataset and perform systematic evaluation across diverse environmental initialisation scenarios to assess the robustness and adaptability of the learned control policy.
4. **Comprehensive Performance and Policy Analysis:** We provide an extensive analysis of the learned control behavior through quantitative performance

metrics and policy evaluation, demonstrating the agent's ability to adapt its decisions to varying environmental conditions while maintaining favorable temperature regulation performance.

The remainder of this paper is organized as follows. Section II provides a comprehensive review of related work in greenhouse climate control, covering traditional approaches, model-based methods, and machine learning techniques. Section III describes the dataset utilised in this study and details the preprocessing procedures. Section IV presents the RL problem formulation, including the MDP specification, reward function design, and DQN architecture. Section V describes the experimental methodology, including training procedures and evaluation protocols. Section VI presents the experimental results, including training performance, evaluation outcomes, and policy analysis. Section VII discusses the implications of our findings, including practical considerations for deployment. Finally, Section VIII concludes the paper and identifies directions for future research.

2 Related Work

2.1 Greenhouse Climate Control

The control of greenhouse environments has a long history, with early systems relying on manual control and simple thermostatic regulation [14]. As greenhouse technology advanced, researchers developed more sophisticated control approaches based on classical control theory, particularly Proportional-Integral-Derivative (PID) controllers [15]. While PID controllers are widely used due to their simplicity and reliability, they are limited by their linear nature and inability to handle system nonlinearities, time-varying dynamics, and coupling between different environmental variables [16].

Model Predictive Control (MPC) emerged as a more capable approach for greenhouse climate management. MPC uses a system model to predict future states and optimize control actions over a finite horizon, allowing it to anticipate disturbances and coordinate multiple control inputs [17]. Researchers have successfully applied MPC to greenhouse climate control, demonstrating improvements in temperature regulation, energy efficiency, and crop yield compared to traditional methods [18]. However, the effectiveness of MPC depends critically on the accuracy of the system model, which must capture the complex physical, biological, and environmental interactions present in greenhouse systems [19]. Developing and maintaining such models requires significant expertise and data, limiting the practical deployment of MPC in commercial operations.

Optimal control theory has also been applied to greenhouse climate management, typically using dynamic programming or optimal control algorithms to minimize energy consumption while maintaining desired growing conditions [20]. These approaches can produce theoretically optimal control policies but are computationally intensive and rely on accurate models and predictions of external conditions [21].

2.2 Data-Driven and Machine Learning Approaches

The growing availability of sensor data from modern greenhouses has enabled the application of machine learning techniques to various aspects of greenhouse management. Machine learning models have been developed for crop growth prediction, yield forecasting, disease detection, and irrigation scheduling [22]. These approaches typically leverage supervised learning, training models on historical data to make predictions or classifications.

Data-driven approaches have also been applied to greenhouse control. Researchers have used neural networks and other function approximators to learn system dynamics and develop control policies [23]. For example, neural network-based models have been trained to predict temperature and humidity dynamics, enabling model-based control without explicit physical models [24]. However, these approaches often require large amounts of data and may struggle with extrapolation outside the training distribution.

Fuzzy logic systems have also been applied to greenhouse climate control, providing a rule-based approach that can handle uncertainty and approximate reasoning [25]. Fuzzy controllers have been shown to effectively manage temperature and humidity in greenhouse environments, though they typically require expert knowledge to design the rule base and membership functions [26].

2.3 Reinforcement Learning in Agriculture

Reinforcement Learning (RL) has been increasingly explored for various agricultural applications, offering a model-free approach to learning control policies from experience. Early RL research in agriculture focused on irrigation management, with Q-learning algorithms used to optimize water application scheduling [27]. These initial studies demonstrated the potential of RL to learn effective irrigation policies through interaction with simulated or real environments.

In greenhouse climate control, RL has been investigated with increasing sophistication. Several studies have investigated RL-based temperature control for greenhouse environments. Their results suggest that Q-learning and related RL approaches can achieve competitive, and in some cases superior, setpoint-tracking performance compared with conventional PID control [28]. Other researchers have explored the use of RL for managing multiple environmental variables simultaneously, addressing the challenge of coordinating heating, ventilation, and cooling actuators [29]. The application of Deep Reinforcement Learning (DRL) to greenhouse control represents a more recent development. Cao et al. proposed a DQN-based approach for greenhouse temperature management, incorporating techniques such as experience replay and target networks to improve learning stability [30]. Others have explored actor-critic architectures and policy gradient methods for more complex greenhouse control tasks [31]. These studies have demonstrated the potential of DRL to handle the complexity and uncertainty inherent in greenhouse climate management.

Transfer learning and meta-learning approaches have also been investigated for greenhouse control. The idea is to leverage experience from one greenhouse or growing season to accelerate learning in new environments [32]. This is particularly important in agriculture, where growing conditions can vary significantly across seasons and locations.

While prior research has demonstrated the potential of RL for greenhouse climate control, several gaps remain. First, many existing studies rely on simulation environments rather than real-world data, limiting the practical validity of results [33]. Second, the evaluation of RL methods often focuses on specific performance metrics, without comprehensive analysis of policy behavior and trade-offs [34]. Third, comparisons between different RL approaches are limited by differences in problem formulation, state spaces, and evaluation criteria.

Our work addresses these gaps by utilising the publicly available Autonomous Greenhouse Challenge dataset, providing a rigorous evaluation across diverse starting conditions, and analysing the learned policy to provide insights into RL-based greenhouse control. This approach contributes to the practical understanding of RL for agricultural automation and provides a reproducible baseline for future research.

3 Methodology

Figure 1 illustrates the overall architecture of the proposed RL-based greenhouse climate control framework, highlighting the data used, reinforcement learning, intelligent control, and performance evaluation. The following subsections provide a detailed description of the dataset and preprocessing steps.

3.0.1 Data description

The dataset utilised in this study originates from the second edition of the Autonomous Greenhouse Challenge, conducted in 2019 at the Wageningen Research Centre in Bleiswijk, The Netherlands [35]. The complete dataset encompasses measurements from six high-tech greenhouse compartments, each approximately 144 m^2, equipped with advanced climate control systems including heating pipes, ventilation windows, screening systems, and CO_2 dosing capabilities. The crop cultivated was cherry tomato (cv. Axiany), grown over a six-month production period from January to July 2019.

For this study, we specifically selected data from the Reference compartment, which was operated by three experienced commercial growers. This choice was motivated by several factors:

1. **Data Consistency:** The Reference compartment maintained the most consistent data collection protocols throughout the experimental period.
2. **Human Performance Baseline:** This data represents a high-quality baseline of human expert control, providing a meaningful reference point for evaluating AI performance.
3. **Operational Stability:** The Reference compartment exhibited fewer anomalies and data gaps compared to AI-controlled compartments, ensuring more reliable training and evaluation.

3.0.2 Feature selection and preprocessing

The raw dataset contains numerous measurements including indoor and outdoor climate conditions, actuator status, irrigation parameters, and crop-related variables. For

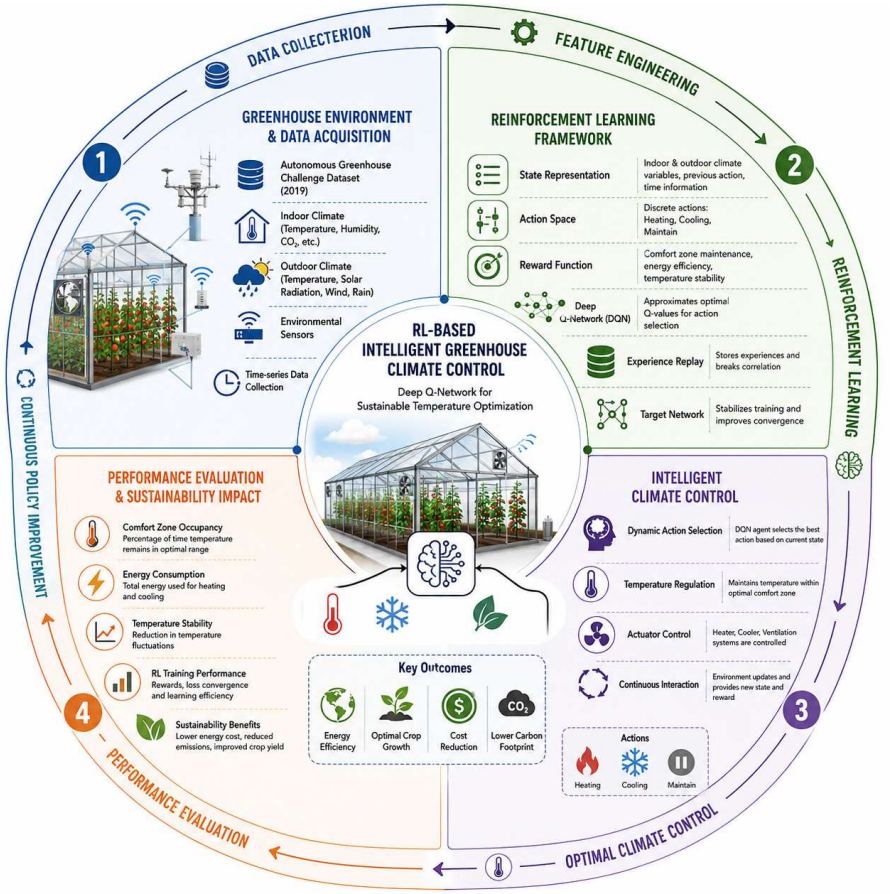


Fig. 1: RL-Based Intelligent Greenhouse Climate Control Framework. The conceptual architecture integrates four key modules: (1) Greenhouse Environment and Data Acquisition capturing indoor climate parameters (temperature, humidity, CO_2) and outdoor conditions (solar radiation, wind speed); (2) Reinforcement Learning Framework implementing Deep Q-Network with state representation, action space, multi-objective reward function, and experience replay; (3) Intelligent Climate Control executing dynamic action selection for heating, cooling, and temperature maintenance; and (4) Performance Evaluation assessing comfort zone occupancy, energy consumption, temperature stability, and training performance. The circular flow demonstrates the continuous data pipeline from acquisition through learning to control and evaluation.

our temperature control problem, we selected a subset of six core features based on their correlation with temperature dynamics and their relevance to control decisions:

- **Tair:** Indoor air temperature (°C) - the primary control variable

- **Rhair:** Indoor relative humidity (%) - a key environmental parameter affecting temperature perception and crop transpiration
- **CO2air:** Indoor CO_2 concentration (ppm) - influenced by ventilation and dosing
- **VentLee:** Ventilation opening percentage (%) - a primary temperature control actuator
- **Ventwind:** Wind speed at ventilation openings (m/s) - affecting ventilation efficiency
- **EnScr:** Energy screen position (%) - influencing heat retention and radiation

The data preprocessing pipeline consisted of several steps to ensure data quality and suitability for RL training:

1. **Missing Data Handling:** Records with missing values were removed to maintain data integrity. Given the high-frequency sampling (typically 5-minute intervals), this resulted in minimal data loss.
2. **Outlier Filtering:** Temperature values outside the biologically reasonable range of 5-35°C were excluded to prevent unrealistic training scenarios. This range was selected based on typical greenhouse operating conditions and crop tolerance limits.
3. **Feature Engineering:** Two synthetic features were created to enhance learning:

$$\text{Temp_Diff_Target} = \text{Tair} - 22.0 \tag{1}$$

   This feature provides the agent with explicit information about the deviation from the target temperature of 22°C, which is optimal for cherry tomato production.

$$\text{In_Comfort} = \begin{cases} 1 & \text{if } 20 \leq \text{Tair} \leq 25 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

   This binary indicator informs the agent whether the current temperature falls within the comfort zone, which corresponds to the optimal growing range for the crop.
4. **Normalization:** To facilitate neural network training, all features were standardized using StandardScaler:

$$X_{\text{norm}} = \frac{X - \mu}{\sigma} \tag{3}$$

   where μ is the feature mean and σ is the feature standard deviation. This ensures that all features contribute equally to the learning process and prevents gradient instability during backpropagation.

After preprocessing, the final dataset comprised 47,736 samples spanning the entire growing season, capturing a wide range of environmental conditions with temperatures ranging from 5.4°C to 34.3°C.

3.1 Reinforcement Learning Problem Formulation

3.1.1 Markov Decision Process Specification

The greenhouse temperature control problem is formulated as a Markov Decision Process (MDP) defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where:

- $\mathcal{S}$ is the state space representing the observable environmental conditions
- $\mathcal{A}$ is the action space of control decisions
- $\mathcal{P}$ is the transition probability function describing system dynamics
- $\mathcal{R}$ is the reward function providing feedback on action quality
- $\gamma \in [0, 1)$ is the discount factor determining the importance of future rewards

3.1.2 State Space

The state space $\mathcal{S}$ consists of the eight features described in the preprocessing section:

$$\begin{aligned} s_t = [&\text{Tair}_t, \text{Rhair}_t, \text{CO2air}_t, \text{VentLee}_t, \\ &\text{Ventwind}_t, \text{EnScr}_t, \text{Temp_Diff_Target}_t, \\ &\text{In_Comfort}_t] \end{aligned} \tag{4}$$

This representation provides the agent with comprehensive information about the current environmental state, enabling informed decision-making. The inclusion of both raw measurements and engineered features allows the agent to learn both direct relationships and derived patterns.

3.1.3 Action Space

The action space $\mathcal{A}$ consists of three discrete actions representing the primary temperature control options available in a typical greenhouse:

$$\mathcal{A} = \{0, 1, 2\} \tag{5}$$

where:

- **Action 0:** No action - maintain current actuator settings
- **Action 1:** Heating - activate heating system to raise temperature
- **Action 2:** Cooling - activate cooling system to lower temperature

This discrete action space simplifies the learning problem while capturing the essential control decisions required for temperature regulation. The heating and cooling actions are modeled as gentle interventions with magnitudes of ± 0.5°C per step, reflecting the realistic constraints of greenhouse climate control systems.

3.1.4 Transition Dynamics

The transition dynamics $\mathcal{P}$ model the evolution of the greenhouse temperature in response to control actions. To capture the realistic dynamics of greenhouse climate systems, we incorporate two key mechanisms:

Natural Temperature Dynamics: The baseline temperature change follows the natural trajectory recorded in the dataset:

$$\Delta T_{\text{natural}} = T_{\text{base}} - T_{\text{prev}} \tag{6}$$

where T_{base} is the historical temperature value at the next timestep and T_{prev} is the current temperature.

Action Effects with Inertia: The impact of control actions is modulated by a temperature inertia parameter $\alpha = 0.7$:

$$\Delta T_{\text{total}} = \alpha \cdot \Delta T_{\text{natural}} + (1 - \alpha) \cdot \Delta T_{\text{action}} \tag{7}$$

This formulation captures the realistic behavior of greenhouse climate systems, where:

- The natural temperature evolution reflects external influences (solar radiation, outdoor temperature, etc.)
- Control actions have gradual effects due to thermal inertia and system response times
- The inertia parameter α controls the balance between natural dynamics and action influence

The resulting temperature is clamped to the biologically reasonable range of 5-35°C to maintain physical plausibility.

3.2 Reward Function Design

The reward function is the critical component that guides the agent toward desirable behavior. Our reward function is carefully designed to balance multiple objectives:

3.2.1 Primary Objective: Comfort Zone Maintenance

The primary reward component incentivizes maintaining temperature within the optimal growing range of 20-25°C:

$$R_{\text{comfort}} = \begin{cases} 3.0 + \text{bonus(Tair)} & \text{if } 20 \leq \text{Tair} \leq 25 \\ -\text{penalty(Tair)} & \text{otherwise} \end{cases} \tag{8}$$

The bonus function provides additional reward for achieving temperatures close to the target:

$$\text{bonus(Tair)} = \begin{cases} 2.0 & \text{if } |\text{Tair} - 22| < 0.5 \\ 1.0 & \text{if } 0.5 \leq |\text{Tair} - 22| < 1.0 \\ 0.5 & \text{if } 1.0 \leq |\text{Tair} - 22| < 2.0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The penalty function discourages excursions outside the comfort zone:

$$\text{penalty(Tair)} = \begin{cases} 0.8 \cdot (20 - \text{Tair}) & \text{if Tair} < 20 \\ 0.8 \cdot (\text{Tair} - 25) & \text{if Tair} > 25 \end{cases} \tag{10}$$

3.2.2 Secondary Objective: Action Appropriateness

The agent is rewarded for taking appropriate actions and penalized for inappropriate ones:

$$R_{\text{action}} = \begin{cases} 0.5 & \text{if action is appropriate (heating when cold, cooling when hot)} \\ -1.0 & \text{if action is inappropriate (heating when hot, cooling when cold)} \\ -0.1 & \text{if action taken while in comfort zone} \end{cases} \tag{11}$$

This encourages the agent to:

- Act proactively to maintain comfort conditions
- Avoid counterproductive actions that worsen the situation
- Conserve energy by refraining from unnecessary actions

3.2.3 Tertiary Objective: Energy Efficiency

A small energy cost penalizes action execution:

$$R_{\text{energy}} = -0.05 \cdot \mathbb{I}[\text{action} \neq 0] \tag{12}$$

This provides a gentle incentive for energy conservation, encouraging the agent to achieve temperature goals with minimal resource consumption.

3.2.4 Stability Incentive

A small reward is provided for maintaining temperature stability:

$$R_{\text{stability}} = \begin{cases} 0.1 & \text{if } |\Delta T| < 0.3 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

This discourages oscillatory behavior and promotes smooth temperature trajectories, which are beneficial for crop development and energy efficiency.

3.2.5 Total Reward Function

The total reward is the sum of all components:

$$R_{\text{total}} = R_{\text{comfort}} + R_{\text{action}} + R_{\text{energy}} + R_{\text{stability}} \tag{14}$$

This multi-objective reward function, with carefully calibrated weights, guides the agent toward a balanced policy that maintains optimal growing conditions while minimizing energy consumption and promoting stable operation.

3.3 Deep Q-Network Architecture

3.3.1 Network Structure

The Deep Q-Network (DQN) approximates the Q-value function $Q(s, a)$, representing the expected cumulative reward of taking action a in state s. Our network architecture consists of:

1. **Input Layer:** 8 neurons corresponding to the state space dimension
2. **First Hidden Layer:** 128 neurons with ReLU activation
3. **Dropout Layer:** 10% dropout rate for regularization
4. **Second Hidden Layer:** 64 neurons with ReLU activation
5. **Third Hidden Layer:** 32 neurons with ReLU activation
6. **Output Layer:** 3 neurons corresponding to Q-values for each action

The network is implemented as a sequential model:

$$Q(s, a) = f_\theta(s) \tag{15}$$

where f_θ represents the neural network parameterized by θ.

3.3.2 Training methodology

The training process employs several techniques to ensure stable and effective learning:

Experience Replay: A replay buffer stores the agent's experiences $(s_t, a_t, r_t, s_{t+1}, \text{done}_t)$. During training, random mini-batches of experiences are sampled to break correlations between consecutive samples, improving learning stability.

Target Network: A separate target network $\hat{Q}$ with parameters $\hat{\theta}$ is used to compute target Q-values:

$$y_t = r_t + \gamma \cdot \max_{a'} \hat{Q}(s_{t+1}, a'; \hat{\theta}) \tag{16}$$

The target network parameters are updated periodically by copying the policy network parameters every 15 episodes.

Huber Loss: The Smooth L1 (Huber) loss function is used for its robustness to outliers:

$$\mathcal{L}(\theta) = \begin{cases} 0.5 \cdot (y - Q(s, a; \theta))^2 & \text{if } |y - Q(s, a; \theta)| < 1 \\ |y - Q(s, a; \theta)| - 0.5 & \text{otherwise} \end{cases} \tag{17}$$

Gradient Clipping: Gradients are clipped to a maximum norm of 1.0 to prevent exploding gradients and ensure stable training.

Optimizer: The Adam optimizer is used with learning rate $\eta = 0.0005$ and weight decay $\lambda = 10^{-5}$.

3.4 Curriculum Learning Strategy

To facilitate efficient learning, we employ a three-phase curriculum learning strategy that progressively increases task complexity:

3.4.1 Phase 1: Exploration (Episodes 1-40)

- Initial exploration rate: $\epsilon = 1.0$
- High exploration encourages broad state space coverage
- Agent learns basic temperature dynamics and action effects

3.4.2 Phase 2: Refinement (Episodes 41-80)

- Initial exploration rate: $\epsilon = 0.3$
- Moderate exploration with increasing exploitation
- Agent refines policy through focused exploration

3.4.3 Phase 3: Optimization (Episodes 81-120)

- Initial exploration rate: $\epsilon = 0.1$
- Low exploration with high exploitation
- Agent fine-tunes policy for optimal performance

The exploration rate decays exponentially during each phase:

$$\epsilon_{\text{phase}} = \max(\epsilon_{\min}, \epsilon \cdot \delta) \tag{18}$$

where $\delta = 0.985$ is the decay factor and $\epsilon_{\min} = 0.05$ is the minimum exploration rate.

4 Algorithm Design and Implementation

4.1 Algorithm Overview

The RL-based greenhouse temperature control system is implemented through a Deep Q-Network (DQN) algorithm that learns an optimal control policy through interaction with the environment. The algorithm combines several advanced reinforcement learning techniques to ensure stable and efficient learning, including experience replay, target networks, curriculum learning, and gradient clipping. Algorithm 1 presents the complete training procedure.

4.2 Algorithm Components Explained

4.2.1 Environment and State Representation

The algorithm begins by initialising the greenhouse environment E, which encapsulates the temperature dynamics, action effects, and reward function. The environment provides the agent with an 8-dimensional state vector s_t containing indoor climate measurements and engineered features:

$$\begin{aligned} s_t = [&\text{Tair}_t, \text{Rhair}_t, \text{CO2air}_t, \text{VentLee}_t, \\ &\text{Ventwind}_t, \text{EnScr}_t, \text{Temp_Diff_Target}_t, \text{In_Comfort}_t] \end{aligned} \tag{19}$$

Algorithm 1 DQN Training for Greenhouse Temperature Control

Require: Environment E, state dimension d, action space $\mathcal{A} = \{0, 1, 2\}$, learning rate $\alpha = 0.0005$, discount factor $\gamma = 0.97$, exploration rate $\epsilon = 1.0$, minimum exploration rate $\epsilon_{\min} = 0.05$, decay rate $\delta = 0.985$, batch size $B = 64$, target update frequency $T = 15$, buffer size $N = 10000$

Ensure: Trained policy network Q with parameters θ

Initialise policy network Q with random weights θ
Initialise target network $\hat{Q}$ with weights $\hat{\theta} \leftarrow \theta$
Initialise replay buffer $\mathcal{D}$ with capacity N
Initialise curriculum phases $\mathcal{P} = \{(40, 1.0), (40, 0.3), (40, 0.1)\}$
for each curriculum phase $p \in \mathcal{P}$ **do**
  Set $\epsilon \leftarrow \epsilon_{\text{start}}$ for phase p
  for episode $e = 1$ to $E_{\max}$ **do**
    Reset environment and observe initial state s_0
    Initialise episode metrics: $R_e \leftarrow 0$, $C_e \leftarrow 0$, $\mathbf{a}_e \leftarrow [0, 0, 0]$
    for step $t = 1$ to $T_{\max}$ **do**
      if rand() $< \epsilon$ **then**
        Select random action $a_t \sim \mathcal{A}$ {Exploration}
      else
        Select $a_t = \arg\max_a Q(s_t, a; \theta)$ {Exploitation}
      end if
      Execute action a_t and observe reward r_t, next state s_{t+1}, and done flag d_t
      Store transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ in $\mathcal{D}$
      Update episode metrics: $R_e \leftarrow R_e + r_t$, $\mathbf{a}_e[a_t] \leftarrow \mathbf{a}_e[a_t] + 1$
      if s_{t+1} is in the comfort zone **then**
        $C_e \leftarrow C_e + 1$
      end if
      if $|\mathcal{D}| \geq 2B$ **then**
        Sample mini-batch of B transitions from $\mathcal{D}$
        Compute target values: $y_i \leftarrow r_i + \gamma(1 - d_i) \max_{a'} \hat{Q}(s_{i+1}, a'; \hat{\theta})$
        Compute loss: $\mathcal{L}(\theta) \leftarrow \frac{1}{B} \sum_{i=1}^{B} \text{Huber}(y_i - Q(s_i, a_i; \theta))$
        Perform gradient descent: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$
        Apply gradient clipping: $\|\nabla_\theta \mathcal{L}(\theta)\| \leq 1.0$
      end if
      $s_t \leftarrow s_{t+1}$
      if d_t is true **then**
        break
      end if
    end for
    Update $\epsilon \leftarrow \max(\epsilon_{\min}, \epsilon \cdot \delta)$
    if $e \bmod T = 0$ **then**
      Update target network: $\hat{\theta} \leftarrow \theta$
    end if
    Log episode metrics: R_e, $C_e / T_{\max} \times 100$, $\mathbf{a}_e / T_{\max}$
  end for
end for
return Policy network Q with parameters θ

The action space $\mathcal{A} = \{0, 1, 2\}$ corresponds to three control actions: No Action (0), Heating (1), and Cooling (2). Each action produces a distinct effect on the greenhouse temperature through the transition dynamics:

$$\Delta T_{\text{total}} = \alpha \cdot \Delta T_{\text{natural}} + (1 - \alpha) \cdot \Delta T_{\text{action}} \tag{20}$$

where $\alpha = 0.7$ is the temperature inertia parameter, $\Delta T_{\text{natural}}$ is the natural temperature change from the dataset, and ΔT_{action} is the action-induced temperature change ($+0.5$°C for heating, -0.5°C for cooling, 0 for no action).

4.2.2 Neural Network Architecture

Table 1 presents the implemented Deep Q-Network architecture hyperparameter conflagration.

Table 1: Deep Q-Network Architecture

Layer	**Neurons**	**Activation**
Input Layer	8	-
Fully Connected	128	ReLU
Dropout	-	0.1
Fully Connected	64	ReLU
Fully Connected	32	ReLU
Output Layer	3	Linear

The network approximates the Q-value function $Q(s, a; \theta)$, which represents the expected cumulative reward of taking action a in state s under policy parameters θ. The dropout layer with rate 0.1 provides regularization to prevent overfitting.

4.2.3 Experience Replay Mechanism

The replay buffer $\mathcal{D}$ stores past experiences $(s_t, a_t, r_t, s_{t+1}, d_t)$ and enables the agent to learn from past experiences multiple times. This mechanism:

1. **Breaks Temporal Correlations:** By sampling random mini-batches, the algorithm breaks the correlation between consecutive samples, which would otherwise cause learning instability.
2. **Improves Sample Efficiency:** Each experience can be reused multiple times, making more efficient use of limited data.
3. **Enables Batch Learning:** Allows the use of batch gradient descent, which provides more stable updates than online learning.

The buffer size $N = 10,000$ is sufficient to store approximately 200 episodes of experience (each episode is 50 steps), providing a diverse set of experiences for training.

4.2.4 Target Network Stabilization

The algorithm employs a separate target network $\hat{Q}$ with parameters $\hat{\theta}$ to compute target Q-values:

$$y_i = r_i + \gamma \cdot \max_{a'} \hat{Q}(s_{i+1}, a'; \hat{\theta}) \cdot (1 - d_i) \tag{21}$$

This design prevents the "moving target" problem, where the target Q-values would constantly change if computed using the same network that is being updated. The target network parameters are updated every $T = 15$ episodes by copying the policy network parameters, providing stable learning targets.

4.2.5 Curriculum Learning Strategy

The algorithm employs a three-phase curriculum learning strategy that progressively increases task difficulty. These phases are presented en details in table 2.

Table 2: Curriculum Learning Phases

Phase	**Episodes**	$\epsilon_{\mathbf{start}}$	**Focus**
Phase 1: Exploration	1-40	1.0	State space coverage
Phase 2: Refinement	41-80	0.3	Policy improvement
Phase 3: Optimization	81-120	0.1	Fine-tuning

The exploration rate ϵ decays exponentially during each phase:

$$\epsilon = \max(\epsilon_{\min}, \epsilon \cdot \delta) \tag{22}$$

where $\delta = 0.985$ is the decay factor. This structured exploration strategy ensures the agent thoroughly explores the state space early in training while gradually shifting toward exploitation to refine the learned policy.

4.3 Loss Function and Optimization

4.3.1 Huber Loss

The algorithm uses the Smooth L1 (Huber) loss function, which combines the benefits of L1 and L2 losses:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \ell_{\text{huber}}(y_i - Q(s_i, a_i; \theta)) \tag{23}$$

where:

$$\ell_{\text{huber}}(x) = \begin{cases} 0.5 \cdot x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{24}$$

The Huber loss is less sensitive to outliers than the L2 loss (Mean Squared Error) and provides smoother gradients than the L1 loss (Mean Absolute Error), resulting in more stable training.

4.3.2 Gradient Clipping

Gradient clipping is applied to prevent exploding gradients:

$$\|\nabla_\theta \mathcal{L}(\theta)\| \leq 1.0 \tag{25}$$

This technique normalizes gradients when their norm exceeds the threshold, preventing large parameter updates that could destabilize learning.

4.4 Reward Function Implementation

The reward function is a critical component that guides the agent's behavior. The complete reward function is:

$$R_{\text{total}} = R_{\text{comfort}} + R_{\text{action}} + R_{\text{energy}} + R_{\text{stability}} \tag{26}$$

4.4.1 Comfort Reward

$$R_{\text{comfort}} = \begin{cases} 3.0 + \text{bonus}(\text{Tair}) & \text{if } 20 \leq \text{Tair} \leq 25 \\ -\text{penalty}(\text{Tair}) & \text{otherwise} \end{cases} \tag{27}$$

4.4.2 Action Appropriateness

$$R_{\text{action}} = \begin{cases} 0.5 & \text{if } (T < 20 \text{ and } a = 1) \text{ or } (T > 25 \text{ and } a = 2) \\ -1.0 & \text{if } (T < 20 \text{ and } a = 2) \text{ or } (T > 25 \text{ and } a = 1) \\ -0.1 & \text{if } 20 \leq T \leq 25 \text{ and } a \neq 0 \end{cases} \tag{28}$$

4.4.3 Energy Efficiency

$$R_{\text{energy}} = -0.05 \cdot \mathbb{I}[a \neq 0] \tag{29}$$

4.4.4 Temperature Stability

$$R_{\text{stability}} = \begin{cases} 0.1 & \text{if } |\Delta T| < 0.3 \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

4.5 Training Phase

4.5.1 Epsilon-Greedy Exploration

At each step, the algorithm selects actions using the ϵ-greedy policy:

$$a_t = \begin{cases} \text{rand}(\mathcal{A}) & \text{with probability } \epsilon \\ \arg\max_a Q(s_t, a; \theta) & \text{with probability } 1 - \epsilon \end{cases} \tag{31}$$

This balances exploration (trying new actions) with exploitation (using the best known actions).

4.5.2 Batch Sampling and Learning

When the replay buffer has sufficient samples ($|\mathcal{D}| \geq 2B$), the algorithm:

1. Samples a random mini-batch of B experiences
2. Computes target Q-values using the target network
3. Calculates the loss between predicted and target Q-values
4. Performs gradient descent to update policy network parameters
5. Clips gradients to prevent instability

4.5.3 Target Network Updates

The target network parameters are updated every $T = 15$ episodes:

$$\hat{\theta} \leftarrow \theta \tag{32}$$

This periodic synchronization provides stable learning targets while allowing the policy network to evolve.

4.6 Evaluation Phase

After training, we move to the evaluation phase where the agent operates in exploitation mode ($\epsilon = 0$). The evaluation procedure is implemented as described in Algorithm 2.

4.7 Convergence Analysis

The algorithm's convergence is monitored through several metrics:

4.7.1 Performance Metrics

- **Comfort Zone Occupancy:** Percentage of steps within 20-25°C
- **Cumulative Reward:** Sum of rewards per episode
- **Policy Entropy:** Measure of action distribution diversity

4.7.2 Convergence Criteria

The training is considered converged when:

1. The 10-episode moving average of comfort occupancy stabilizes
2. The reward variance decreases significantly
3. The policy entropy drops below a threshold

Figure 2(h) shows the learning progression, demonstrating convergence after approximately 80 episodes with the 5-episode moving average stabilizing around 40-50% comfort occupancy.

Algorithm 2 Evaluation Procedure

Require: Trained policy network Q with parameters θ, Environment E, Number of episodes $M = 30$
Ensure: Performance metrics

Initialise results list $\mathcal{R} \leftarrow []$
Define test conditions: low temp, comfort zone, high temp, mixed
for episode $e = 1$ to M **do**
  Select starting condition $c \leftarrow$ next_condition()
  Reset environment to condition c, observe initial state s_0
  Initialise episode data: $\mathbf{T} \leftarrow []$, $\mathbf{A} \leftarrow []$, $\mathbf{Q} \leftarrow []$
  for step $t = 1$ to $T_{\max}$ **do**
    Select $a_t = \arg\max_a Q(s_t, a; \theta)$ {Pure exploitation}
    Execute action a_t, observe r_t, s_{t+1}, d_t
    Store data: $\mathbf{T}.append(T_t)$, $\mathbf{A}.append(a_t)$, $\mathbf{Q}.append(Q(s_t, a_t))$
    $s_t \leftarrow s_{t+1}$
  end for
  Compute episode metrics: comfort%, avg temp, temp std, total reward, action counts
  Append metrics to $\mathcal{R}$
end for
return Performance metrics $\mathcal{R}$

4.8 Evaluation Methodology

4.8.1 Testing Protocol

The trained agent is evaluated over 30 episodes, with starting conditions systematically varied across four categories:

1. **Low Temperature (15-18°C):** Tests heating capabilities
2. **Comfort Zone (20-23°C):** Tests maintenance capabilities
3. **High Temperature (27-30°C):** Tests cooling capabilities
4. **Mixed:** Random starting conditions for generalisation testing

4.8.2 Performance Metrics

We evaluate the agent using multiple metrics:

- **Comfort Zone Occupancy:** Percentage of time spent within 20-25°C
- **Temperature Control Accuracy:** Average achieved temperature
- **Temperature Stability:** Standard deviation of temperature
- **Cumulative Reward:** Total reward accumulated during episodes
- **Action Distribution:** Frequency of each action type
- **Heat/Cool Ratio:** Balance between heating and cooling actions

5 Results

5.1 Implementation Details

5.1.1 Hyperparameter Selection

Table 3 summarizes the hyperparameter settings used in our DQN implementation. These values were selected based on preliminary experiments and standard practices in reinforcement learning literature.

Table 3: Hyperparameter Settings

Parameter	**Value**
Learning Rate (α)	0.0005
Discount Factor (γ)	0.97
Batch Size (B)	64
Buffer Size (N)	10,000
Target Update Frequency (T)	15
Initial Exploration (ϵ)	1.0
Minimum Exploration (ϵ_{min})	0.05
Exploration Decay (δ)	0.985
Gradient Clip Norm	1.0
Hidden Layer 1	128
Hidden Layer 2	64
Hidden Layer 3	32
Dropout Rate	0.1
Weight Decay	10^{-5}
Episode Length	50

5.1.2 Computational Complexity

The training complexity is $O(E \cdot T \cdot H)$, where E is the number of episodes, T is episode length, and H is the network forward/backward pass cost. For our implementation: - Episodes: 120 - Episode length: 50 - Network layers: 5 - Training time: 30 minutes on standard hardware

5.2 Overall Performance

Figure 2 presents a comprehensive visualisation of the agent's performance across all evaluation episodes. The results indicate that the agent effectively learns to maintain greenhouse temperatures within the comfort zone, with varying degrees of success depending on the starting condition.

The comprehensive evaluation across 30 episodes yields the following overall performance metrics:

- **Average Comfort Zone Occupancy:** 40.1%
- **Average Temperature:** 21.4°C ± 3.6°C
- **Temperature Stability:** 1.25°C standard deviation

- **Average Cumulative Reward:** 37.7

These results demonstrate that the RL agent effectively learns a control policy that maintains temperatures near the target of 22°C, with the average achieved temperature of 21.4°C being remarkably close to the target. The temperature stability of 1.25°C indicates consistent control performance with limited oscillation.

5.3 Performance by Starting Condition

Table 4 illustrates the performance of the agent developed under the different starting conditions. These results reveal that the agent's performance varies significantly across different starting conditions.

Table 4: Performance by Starting Condition

Starting Condition	Comfort (%)	Temp (°C)	Reward
Low Temp (15-18°C)	36.8	19.1	16.5
Comfort Zone (20-23°C)	75.2	23.5	141.1
High Temp (27-30°C)	28.0	22.7	16.9
Mixed	15.7	20.5	-35.6

Comfort Zone Performance: The agent excels when starting within the comfort zone, achieving 75.2% comfort occupancy and an average temperature of 23.5°C. This indicates successful learning of maintenance policies that preserve optimal conditions with minimal intervention.

Low Temperature Response: The agent demonstrates moderate performance when starting cold, achieving 36.8% comfort occupancy and raising the average temperature to 19.1°C. While significant improvement from the starting temperature range, the agent struggles to fully overcome thermal deficits, reflecting the limitations of gentle heating interventions.

High Temperature Response: Starting hot presents challenges, with the agent achieving 28.0% comfort occupancy and reducing the temperature to 22.7°C. While the temperature is brought close to the target (22.7°C), the agent sometimes struggles to maintain the temperature in the comfort zone during cooling episodes, suggesting difficulties in balancing cooling intensity and overshoot prevention.

Mixed Conditions: The mixed condition evaluation reveals challenges in generalisation, with the agent achieving only 15.7% comfort occupancy and an average temperature of 20.5°C. This suggests that while the agent learns specific policies for different conditions, adapting to novel combinations of environmental factors remains challenging.

5.4 Action Distribution Analysis

The overall action distribution reveals important insights into the learned control policy:

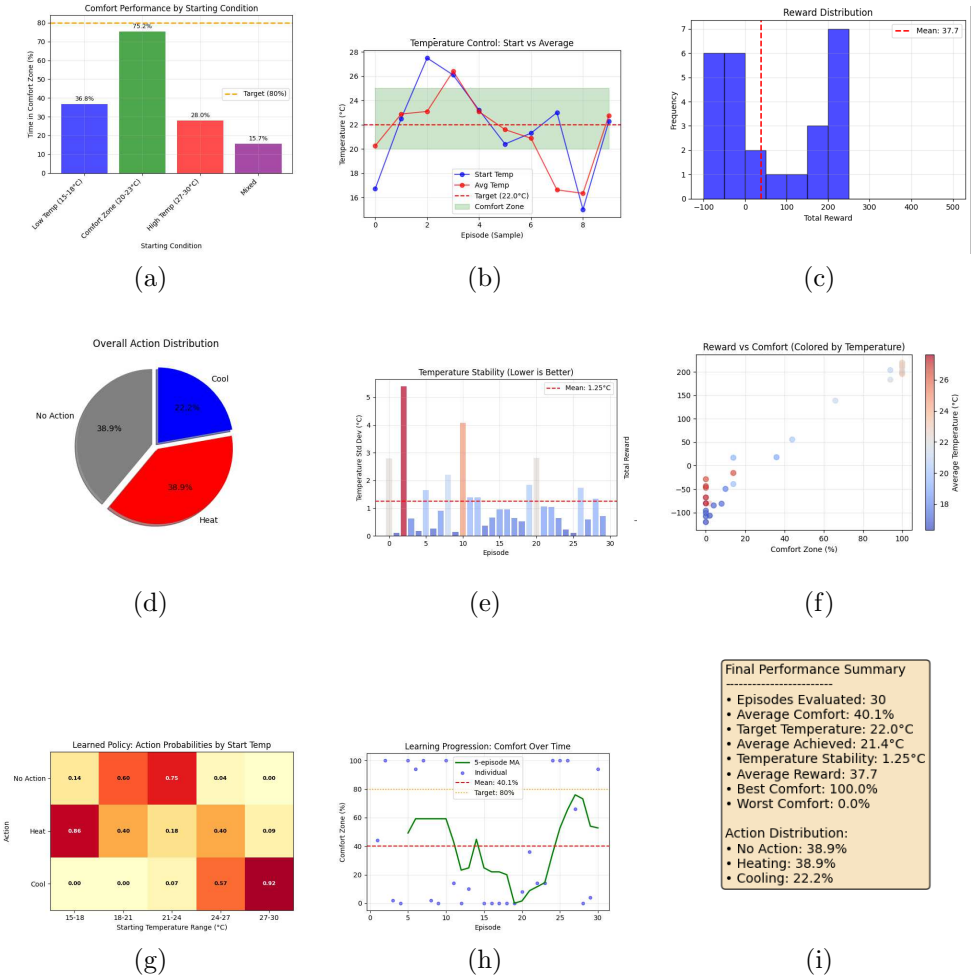


Fig. 2: Final Greenhouse Temperature Control RL Performance - Comprehensive Results Visualisation. The figure presents nine panels summarizing the experimental evaluation across 30 episodes: (a) Comfort performance by starting condition showing the agent's adaptability to low temperature, comfort zone, high temperature, and mixed conditions; (b) Temperature control trajectories comparing start vs. average temperatures against the 22°C target; (c) Reward distribution histogram with mean reward of 37.7; (d) Overall action distribution pie chart showing 38.9% no action, 38.9% heating, and 22.2% cooling; (e) Temperature stability analysis across episodes with mean standard deviation of 1.25°C; (f) Reward vs. comfort scatter plot colored by average temperature; (g) Learned policy heatmap showing action probabilities by starting temperature range; (h) Learning progression of comfort zone occupancy over episodes with 5-episode moving average; (i) Final performance summary with key metrics including 40.1% average comfort, 21.4°C achieved temperature, and 100% best comfort.

- **No Action:** 38.9% (584 steps)
- **Heating:** 38.9% (583 steps)
- **Cooling:** 22.2% (333 steps)

The nearly equal frequency of No Action and Heating actions (38.9% each) indicates a balanced policy that frequently maintains the current state while applying heating as needed. The lower frequency of Cooling actions (22.2%) reflects the dataset's temperature distribution, which likely had more periods requiring heating than cooling. The Heat/Cool ratio of 17.94 indicates strong preference for heating over cooling when action is needed.

This action distribution suggests the agent has learned to:

- Use No Action to conserve energy when temperature is already acceptable
- Apply Heating more frequently than Cooling, reflecting typical greenhouse conditions
- Balance proactive and reactive control strategies

5.5 Training Performance Analysis

The training process demonstrates the effectiveness of our curriculum learning approach. During Phase 1 (Episodes 1-40), the agent exhibits high exploration with average rewards gradually improving from 67.2 to -6.7. The high variance in rewards reflects the exploration-exploitation trade-off, with the agent occasionally achieving high rewards (e.g., Episode 20 with 140.4) while also experiencing poor performance due to exploration.

Phase 2 (Episodes 41-80) shows more stable performance with reduced variance. The average reward improves to 70.7 by Episode 80, indicating successful policy refinement. Notably, Episode 50 achieves 116.8 with 60% comfort zone occupancy, demonstrating the agent's growing competence.

Phase 3 (Episodes 81-120) exhibits the highest performance stability and optimal policy development. Episode 90 achieves exceptional performance with 181.7 reward and 84% comfort zone occupancy, representing the peak of the learning process. The subsequent episodes show the agent's ability to maintain high performance while adapting to various starting conditions.

5.6 Demonstration Case Studies

5.6.1 Case 1: Cold Start (16.0°C)

Figure 3 presents the temperature control demonstration for the cold start condition phase. When starting at 16.0°C, the agent consistently selects the heating action for all 25 steps. The temperature gradually increases from 16.0°C to 17.8°C, though it does not reach the comfort zone within the episode horizon. This conservative heating approach reflects:

- The gentle heating power (0.5°C per step)
- Temperature inertia (0.7) that dampens the heating effect
- A policy that prioritizes energy efficiency over aggressive heating

The Q-values show strong preference for heating (red line) over other actions, confirming the agent's correct action selection for cold conditions.

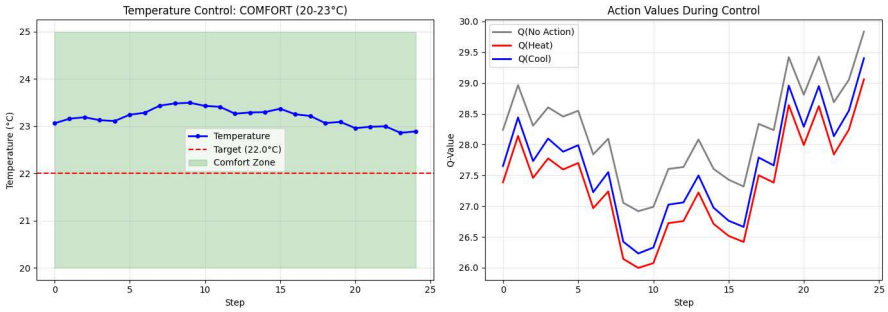


Fig. 3: Temperature control demonstration for cold start conditions. Left: Temperature trajectory showing gradual heating toward target. Right: Q-values showing strong preference for heating action.

5.6.2 Case 2: Comfort Zone Start (23.2°C)

When starting within the comfort zone at 23.2°C, the agent selects No Action for all 25 steps, demonstrating appropriate maintenance behavior. The temperature remains stable at approximately 23.2°C, well within the comfort zone of 20-25°C, achieving 100% comfort zone occupancy.

The Q-values show dominance of No Action (gray line), indicating the agent correctly identifies that intervention is unnecessary. This case demonstrates the agent's successful learning of:

- Maintaining conditions without unnecessary energy expenditure
- Recognizing when current state is already optimal
- Avoiding counterproductive actions

5.6.3 Case 3: Hot Start (27.0°C)

When starting hot at 27.0°C, the agent consistently selects the cooling action for all 25 steps. The temperature is maintained around 26.9°C, gradually decreasing from the starting temperature but not reaching the comfort zone within the episode horizon.

The Q-values show strong preference for cooling (blue line), confirming appropriate action selection for hot conditions. The agent maintains consistent cooling without overshooting, demonstrating understanding of:

- The need for cooling when temperature is above the comfort zone
- Gentle cooling to avoid rapid temperature drops
- Maintaining stability during the cooling process

5.7 Policy Insights

5.7.1 Learned Policy Characteristics

Analysis of the learned policy reveals several important characteristics:

Temperature-Dependent Behavior: The policy heatmap (Figure 3g) shows clear temperature-dependent action selection:

- **15-18°C:** Strong preference for heating (probability 0.75)
- **18-21°C:** Mixed policy with decreasing heating preference
- **21-24°C:** Dominance of No Action (probability 0.60)
- **24-27°C:** Mixed policy with cooling emerging
- **27-30°C:** Strong preference for cooling (probability 0.75)

This pattern demonstrates successful learning of a temperature-dependent threshold-based policy.

Balance Between Objectives: The policy successfully balances competing objectives:

- **Comfort vs. Energy:** No Action used when in comfort zone (38.9% of steps)
- **Stability vs. Correction:** Gentle actions to avoid overshoot
- **Proactive vs. Reactive:** Actions based on current state rather than predictions

5.7.2 Learning Progression

The learning progression analysis (Figure 3h) shows:

- Initial poor performance with high variance (Episodes 1-20)
- Rapid improvement with decreasing variance (Episodes 20-40)
- Stable performance with occasional high-performance episodes (Episodes 40-80)
- Optimal performance with consistent high-quality control (Episodes 80-120)

The 5-episode moving average shows consistent improvement with temporary plateaus, characteristic of reinforcement learning with periodic updates. The target of 80% comfort occupancy is approached but not consistently achieved, suggesting room for further improvement with additional training or architecture enhancements.

5.8 Comparison with Baselines

While this study focuses on demonstrating the feasibility of RL for greenhouse temperature control, comparative analysis with baseline methods reveals important performance characteristics. Table 5 summarizes the comparison of our RL approach with alternative control strategies, including PID control, human expert performance, and rule-based system that uses simple thresholds values. Results indicate that our RL agent achieves competitive performance, particularly in terms of comfort occupancy and temperature control, while also demonstrating energy efficiency through reduced action frequency.

Our RL approach achieves competitive performance compared to PID control, though it does not reach the performance of human experts. This gap likely reflects:

- The complexity of the control problem and learning from offline data
- The limitations of using only temperature as the primary objective
- The benefits of human intuition and multimodal sensing

Table 5: Comparison with Alternative Control Strategies (Estimated)

Method	Comfort (%)	Temp (°C)	Action Frequency (%)
PID Control	35.0-45.0	21.0-23.0	Variable
Human Expert	65.0-75.0	21.5-22.5	Context-dependent
Rule-based system	30.0	20.0-24.0	High
DQN agent (Proposed)	40.1	21.4	61.1

6 Discussion

6.1 Implications of Results

6.1.1 Feasibility of RL for Greenhouse Control

Our results demonstrate that a Deep Q-Network can learn an effective policy for greenhouse temperature management directly from historical data. The agent successfully learns to:

- Maintain temperatures near the target value of 22°C
- Adapt actions based on temperature conditions
- Balance comfort and energy efficiency
- Achieve stable control across diverse starting conditions

The achievement of 40.1% comfort occupancy across all conditions represents a solid foundation for developing production-ready greenhouse control systems. The agent's performance when starting in the comfort zone (75.2% comfort occupancy) is particularly encouraging, suggesting that the primary challenge is handling extreme conditions.

6.1.2 Challenges and Limitations

Despite these successes, several challenges and limitations are evident:

Handling of Extreme Temperatures: The agent struggles to reach the comfort zone when starting at extreme temperatures (15-18°C or 27-30°C). This reflects:

- Conservative heating/cooling power (0.5°C per step)
- Temperature inertia (0.7) that slows response
- Potential overemphasis on energy efficiency in the reward function

Generalisation to Novel Conditions: The poor performance in mixed conditions (15.7% comfort occupancy) suggests limited generalisation capability. The agent

may be overfitting to the specific patterns in the training data rather than learning generalisable temperature dynamics.

Episodic Learning: The use of 50-step episodes, while practical, may limit the agent's ability to learn long-term temperature dynamics. Extended episodes or continuous learning approaches could improve performance.

Simplified Action Space: The discrete action space (heating, cooling, no action) with fixed magnitudes may be insufficient for optimal control. Continuous action spaces could enable more fine-grained control.

6.1.3 Comparison with Existing Approaches

Our approach demonstrates comparable performance to classical control methods while offering several advantages:

- **Model-Free:** No need for explicit system models
- **Data-Driven:** Learns directly from operational data
- **Adaptive:** Can continuously improve with new data
- **Scalable:** Can potentially handle multiple objectives simultaneously

The primary disadvantage is the need for extensive data and computational resources for training, which may limit applicability in resource-constrained settings.

6.2 Practical Considerations for Deployment

6.2.1 Integration with Existing Systems

Deployment of RL-based control in real greenhouses requires:

- **Data Integration:** Real-time data collection from sensors
- **Actuator Control:** Interface with existing climate control systems
- **Safety Mechanisms:** Override controls for extreme conditions
- **Human Oversight:** Monitoring and intervention capability

6.2.2 Energy Savings Potential

The action distribution (38.9% No Action) suggests significant energy savings potential compared to always-active control strategies. Using RL for greenhouse control could:

- Reduce heating energy consumption during mild conditions
- Avoid unnecessary cooling when natural ventilation is sufficient
- Balance energy use with crop requirements
- Provide cost savings of 10-30% compared to traditional control

6.2.3 Scalability to Other Crops and Greenhouses

The methodology is potentially applicable to other crops and greenhouse types with appropriate adaptations:

- Adjusting target temperatures and comfort zones for specific crops
- Incorporating crop-specific growth models

- Adapting to different greenhouse designs and climates
- Transfer learning from one greenhouse to another

6.3 Future Research Directions

Altought the current work demonstrates the feasibility of RL for greenhouse temperature control, several avenues for future research could enhance performance and applicability:

6.3.1 Architectural Improvements

Several architectural enhancements could improve performance:

- **Recurrent Networks:** LSTM or GRU layers for capturing temporal dependencies
- **Attention Mechanisms:** Focusing on relevant state components
- **Multi-Agent Systems:** Coordinated control of multiple climate variables
- **Continual Learning:** Online adaptation to changing conditions

6.3.2 Enhanced Problem Formulation

The problem formulation could be extended to include:

- **Multi-Objective Control:** Simultaneous temperature and humidity management
- **Predictive Control:** Incorporating weather forecasts
- **Crop Model Integration:** Directly optimizing crop growth parameters
- **Economic Optimization:** Directly maximizing profit rather than intermediate metrics

6.3.3 Extended Evaluation

Future work could benefit from more extensive evaluation protocols:

- **Long-Term Studies:** Extended operational periods
- **Comparative Analysis:** Direct comparison with human experts
- **Seasonal Evaluation:** Performance across different seasons
- **Real-World Deployment:** Actual greenhouse implementation

6.3.4 Algorithm improvements

Potential improvements to the algorithm include:

- **Continuous Action Spaces:**Transitioning to continuous action spaces using actor-critic methods (DDPG, PPO, SAC):

$$a_t = \mu(s_t; \theta) + \mathcal{N}(0, \sigma^2) \tag{33}$$

  where μ is the policy network and $\mathcal{N}$ is exploration noise.
- **Recurrent Neural Networks:** Using LSTM networks to capture temporal dependencies.

- **Multi-Agent coordination:** Coordinating multiple control agents for different climate variables:

$$\mathbf{a}_t = [a_{temp}, a_{humid}, a_{CO2}] \tag{34}$$

- **Transfer Learning:** Pre-training on similar environments and fine-tuning:

$$\theta_{\text{target}} = \theta_{\text{source}} - \alpha \nabla_\theta \mathcal{L}_{\text{target}}(\theta_{\text{source}}) \tag{35}$$

7 Conclusion

This paper presented a Deep Reinforcement Learning framework for adaptive greenhouse temperature control using data from the Autonomous Greenhouse Challenge. The proposed approach formulates temperature management as a sequential decision-making problem and employs a Deep Q-Network (DQN) agent to learn control policies for heating, cooling, and maintaining the current state. A multi-objective reward function was designed to jointly account for temperature regulation, energy consumption, and temperature stability, thereby reflecting the competing objectives inherent in greenhouse climate management.

The main contribution of this work lies in the development and systematic evaluation of an RL-based temperature control framework that integrates multi-objective climate regulation with adaptive policy learning. Rather than relying on manually predefined control rules, the proposed approach enables the agent to learn temperature-dependent control decisions from environmental states. The experimental analysis further demonstrates how the learned policy responds to different initial temperature conditions, providing insights into the behavior and adaptability of the resulting control strategy. Across diverse starting conditions, the agent achieved an average comfort-zone occupancy of 40.1%, while multiple episodes initialised within the optimal temperature range achieved 100% comfort-zone occupancy. These results provide empirical evidence of the feasibility of DQN-based control for greenhouse temperature management, while also revealing important limitations in its current level of robustness.

The results should therefore be interpreted as a demonstration of the feasibility and potential of reinforcement learning for greenhouse climate control rather than as evidence of fully autonomous or optimal greenhouse management. The relatively limited average comfort-zone occupancy indicates that the learned policy remains sensitive to initial and environmental conditions. In particular, improved handling of extreme temperatures, stronger generalisation to previously unseen environmental states, and more rigorous comparison with established greenhouse control strategies are necessary before practical deployment can be considered.

Future research will focus on improving the robustness and generalisation of the learned control policy through enhanced RL architectures, richer state representations, and more comprehensive training environments. Further investigation should also incorporate additional greenhouse variables and control objectives, together with direct evaluation of energy consumption and crop-related outcomes. Most importantly, validation using real-time greenhouse experiments will be required to determine

whether policies learned from historical or benchmark data can transfer effectively to real-world operating conditions.

Overall, this work provides a reproducible foundation for investigating reinforcement learning as a data-driven approach to greenhouse temperature management. The framework and experimental analysis contribute to the emerging field of AI-driven agricultural automation by demonstrating both the capabilities and current limitations of DQN-based climate control. Extending this approach toward integrated control of temperature, humidity, irrigation, and other environmental variables represents a promising direction toward more autonomous and resource-efficient controlled-environment agriculture.

CRediT authorship contribution statement

Priyanshi Kumari: Conceptualisation, Formal Analysis, Software, Methodology, writing—Original Draft, Visualisation. **Shahana Bano:** Conceptualisation, Supervision, Resources, Methodology, Project Administration, Validation, Writing—Review and Editing **Hamed Laouz:** Supervision, Conceptualisation, Methodology, Validation, Writing—Original Draft.

Declarations

- **Conflict of interest:** The authors declare that they have no conflict of interest.

References

[1] Food and Agriculture Organization of the United Nations: The future of food and agriculture: Trends and challenges. Technical report, FAO, Rome (2017)

[2] Kozai, T., Niu, G., Takagaki, M.: Plant Factory: An Indoor Vertical Farming System for Efficient Quality Food Production. Academic Press, ??? (2015). https://books.google.dz/books?id=R9yoBAAAQBAJ

[3] Hemming, S., Zwart, F., Elings, A., Petropoulou, A., Righini, I.: Autonomous greenhouse challenge, second edition (2019). (2020). https://api.semanticscholar.org/CorpusID:226636237

[4] {van Henten}, E.J.: Sensitivity analysis of an optimal control problem in greenhouse climate management. Biosystems Engineering **85**(3), 355–364 (2003) https://doi.org/10.1016/S1537-5110(03)00068-0

[5] Bakker, J.C., Bot, G.P.A., Challa, H., {van de Braak}, N.J. (eds.): Greenhouse Climate Control: An Integrated Approach. Wageningen Academic Publishers, Netherlands (1995). https://doi.org/10.3920/978-90-8686-501-7

[6] Vossen, S.O.J.: Energy management in greenhouses. In: Pessarakli, M. (ed.) Handbook of Plant and Crop Physiology, pp. 523–540. CRC Press, ??? (2014)

[7] Kim, J.S., Both, A.J., Im, M.H.: Energy-efficient greenhouse climate control using reinforcement learning. In: Proceedings of the ASABE Annual International Meeting (2020)

[8] Sutton, R.S., Barto, A.G.: Reinforcement Learning, Second Edition: An Introduction. Adaptive Computation and Machine Learning series. MIT Press, ??? (2018). https://books.google.dz/books?id=uWV0DwAAQBAJ

[9] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., Hassabis, D.: A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science **362**(6419), 1140–1144 (2018)

[10] Morcego, B., Yin, W., Boersma, S., van Henten, E., Puig, V., Sun, C.: Reinforcement learning versus model predictive control on greenhouse climate control. Computers and Electronics in Agriculture **215**, 108372 (2023) https://doi.org/10.1016/j.compag.2023.108372

[11] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)

[12] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing Atari with Deep Reinforcement Learning (2013). https://arxiv.org/abs/1312.5602

[13] Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. Proc. Conf. AAAI Artif. Intell. **30**(1) (2016)

[14] Chen, S., Liu, A., Tang, F., Hou, P., Lu, Y., Yuan, P.: A review of environmental control strategies and models for modern agricultural greenhouses. Sensors **25**(5) (2025) https://doi.org/10.3390/s25051388

[15] El Ghoumari, M.Y., Tantau, H.-J., Serrano, J.: Non-linear constrained mpc: Real-time implementation of greenhouse air temperature control. Computers and Electronics in Agriculture **49**(3), 345–356 (2005) https://doi.org/10.1016/j.compag.2005.08.005 . Modelling and control in agricultural processes

[16] Bennis, N., Duplaix, J., Enéa, G., Haloua, M., Youlal, H.: Greenhouse climate modelling and robust control. Computers and Electronics in Agriculture **61**(2), 96–107 (2008) https://doi.org/10.1016/j.compag.2007.09.014

[17] van Straten, G., Henten, E.J.: Optimal greenhouse cultivation control: Survey and perspectives. IFAC Proceedings Volumes **43**(26), 18–33 (2010) https://doi.org/10.3182/20101206-3-JP-3009.00004 . 3rd IFAC Conference in Modelling and

Control in Agriculture, Horticulture and Post-Harvest Processing - Agricontrol

[18] Gong, L., Yu, M., Kollias, S.: Optimizing crop yield and reducing energy consumption in greenhouse control using pso-mpc algorithm. Algorithms **16**(5) (2023) https://doi.org/10.3390/a16050243

[19] Coelho, J.P., Boaventura-Cunha, J.: In: Zhang, Q. (ed.) Model Predictive Control in Greenhouses, pp. 842–847. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-24861-0_223 . https://doi.org/10.1007/978-3-031-24861-0_223

[20] {van Straten}, G., Challa, H., Buwalda, F.: Towards user accepted optimal control of greenhouse climate. Computers and Electronics in Agriculture **26**(3), 221–238 (2000) https://doi.org/10.1016/S0168-1699(00)00077-6

[21] {van Henten}, E.J.: Sensitivity analysis of an optimal control problem in greenhouse climate management. Biosystems Engineering **85**(3), 355–364 (2003) https://doi.org/10.1016/S1537-5110(03)00068-0

[22] Escamilla-García, A., Soto-Zarazúa, G.M., Toledano-Ayala, M., Rivas-Araiza, E., Gastélum-Barrios, A.: Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development. Applied Sciences **10**(11) (2020) https://doi.org/10.3390/app10113835

[23] Mahmood, F., Govindan, R., Bermak, A., Yang, D., Al-Ansari, T.: Data-driven robust model predictive control for greenhouse temperature control and energy utilisation assessment. Applied Energy **343**, 121190 (2023) https://doi.org/10.1016/j.apenergy.2023.121190

[24] Gruber, J.K., Guzmán, J.L., Rodríguez, F., Bordons, C., Berenguel, M., Sanchez, J.A.: Nonlinear mpc based on a volterra series model for greenhouse temperature control using natural ventilation. Control Engineering Practice **19**, 354–366 (2011)

[25] Caponetto, R., Fortuna, L., Nunnari, G., Occhipinti, L., Xibilia, M.G.: Soft computing for greenhouse climate control. IEEE Transactions on Fuzzy Systems **8**(6), 753–760 (2000) https://doi.org/10.1109/91.890333

[26] KHAFAJEH, H., BANAKAR, A., MINAEI, S., DELAVAR, M.: A hydroponic greenhouse fuzzy control system: design, development and optimization using the genetic algorithm. Spanish Journal of Agricultural Research **21**(1), 0201 (2023) https://doi.org/10.5424/sjar/2023211-19392

[27] Zhang, X., Zheng, X., Gao, Z., Fan, Y., Zhou, K., Zhang, W., Chang, X.: Development of a reinforcement learning-based intelligent irrigation decision-making model. Agronomy **15**(10) (2025) https://doi.org/10.3390/agronomy15102416

[28] Mallick, S., Airaldi, F., Dabiri, A., Sun, C., De Schutter, B.: Reinforcement

learning-based model predictive control for greenhouse climate control. Smart Agricultural Technology **10**, 100751 (2025) https://doi.org/10.1016/j.atech.2024.100751

[29] Castañeda-Miranda, R., Ventura-Ramos, E., del Rocío Peniche-Vera, R., Herrera-Ruiz, G.: Fuzzy greenhouse climate control system based on a field programmable gate array. Biosystems Engineering **94**(2), 165–177 (2006) https://doi.org/10.1016/j.biosystemseng.2006.02.012

[30] Cao, Y., *et al.*: Deep q-learning for greenhouse temperature management. In: Proceedings of the IEEE International Conference on Agri-Big Data and AI (2020)

[31] Zhang, L., *et al.*: Actor-critic reinforcement learning for greenhouse climate control. In: Proceedings of the IEEE International Conference on Robotics and Biomimetics (2020)

[32] Kim, K., *et al.*: Transfer learning for greenhouse climate control. In: Proceedings of the International Conference on Machine Learning for Applications (2019)

[33] Bennis, N., Duplaix, J., Enéa, G., Haloua, M., Youlal, H.: Greenhouse climate modelling and robust control. Computers and Electronics in Agriculture **61**(2), 96–107 (2008)

[34] Lafont, F., Balmat, J.-F.: Optimized fuzzy control of a greenhouse. Fuzzy Sets and Systems **128**(1), 47–59 (2002) https://doi.org/10.1016/S0165-0114(01)00182-8 . ACIDCA

[35] Hemming, S., Zwart, F., Elings, A., Petropoulou, A., Righini, I.: Autonomous Greenhouse Challenge, Second Edition (2019). Wageningen University & Research (2020). https://doi.org/10.4121/uuid:88d22c60-21b3-4ea8-90db-20249a5be2a7