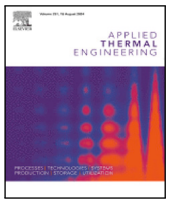
Applied Thermal Engineering 270 (2025) 126106

Contents lists available at ScienceDirect

Applied Thermal Engineering

journal homepage: www.elsevier.com/locate/ate



Research Paper

Designing reinforcement learning algorithms for building HVAC control: From experimental observation to simulation comparisons

Fangzhou Guo [a,*], Sang woo Ham [a], Donghun Kim [a], Sun Ho Kim [b], Hyeun Jun Moon [b]

[a] Building Technology & Urban Systems Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[b] Department of Architectural Engineering, Dankook University, Yongin, Republic of Korea

ARTICLE INFO

Keywords:
Reinforcement learning
Deep Q-network
Deep deterministic policy gradient
Smart building
Energy efficiency
Thermal comfort

ABSTRACT

Advanced supervisory-level control with reinforcement learning (RL) is regarded as a promising solution for HVAC systems to minimize energy consumption while maintaining thermal comfort and indoor air quality. However, most RL applications were conducted in the simulation environment rather than real-world HVAC systems. This paper developed a value-based RL controller termed Deep Q-Network (DQN) for a typical central HVAC system and evaluated its performance in a building test facility. By comparing DQN with a rule-based controller, the study not only demonstrated the cases where DQN could properly maintain indoor comfort but also discussed possible reasons why DQN failed in some other situations. Recognizing the limitations of value-based RL algorithms from the experimental tests, a simulation study was conducted to compare DQN with an alternative RL approach, an actor–critic algorithm termed Deep Deterministic Policy Gradient (DDPG). In scenarios with a relatively large action space, DDPG outperformed DQN by requiring fewer computational resources and achieving better thermal comfort, lower energy consumption, and more stable control actions. The findings suggest that the ability of DDPG to handle continuous control variables more effectively allows for faster convergence in training and more precise control in practice, which enhances the overall efficiency and reliability of the HVAC system.

1. Introduction

According to the survey by the U.S. Energy Information Administration, the total energy consumption of U.S. commercial buildings was 6,787 trillion British thermal units (TBtu) in 2018. Notably, heating, ventilation, and air conditioning (HVAC) systems account for 52% of all major fuels [1]. With the large share of energy usage, significant research efforts have been directed toward developing new technologies aimed at reducing the energy consumption of HVAC systems in buildings. These efforts include enhancements to the efficiency of the building HVAC systems and the implementation of advanced control algorithms.

Advanced control strategies, such as model predictive control (MPC) and reinforcement learning (RL), have gained increasingly more attention because those control applications can provide desired benefits (e.g., energy bill reduction) for existing buildings without major hardware and system retrofits once the controllers are properly trained and applied. MPC uses predictive models and real-time data to optimize the operation of building HVAC systems, improve energy efficiency, reduce costs, and enhance occupant comfort. It continuously adjusts the control actions based on dynamic forecasts and optimization algorithms, ensuring optimal performance. While the MPC shows promising scalability potentials for the relatively simple HVAC and building structure [2,3], designing MPC algorithms for large buildings with complex HVAC systems requires professional expertise to build and calibrate the building and HVAC models. In many cases, these complex HVAC systems exhibit non-linearity in their system dynamics, necessitating a non-linear optimizer to solve the optimization problem. This introduces inherent challenges in robustness due to the formulation structure, initialization, solvers, solver settings, and tuning process [4]. Besides, since every building and its HVAC systems are unique, a well-designed MPC algorithm for one building is difficult to generalize to another building. Those are important reasons why MPC has not been broadly applied in the industry [5].

In recent years, with the advancements in big data and powerful computing capabilities, machine learning (ML) has been integrated into almost every phase of the building life cycle, including but not limited to building energy prediction, predictive maintenance, and demand response. Especially, reinforcement learning (RL), a subset of ML designed for control problems, is emerging as an alternative

* Corresponding author.
E-mail address: fzguo@lbl.gov (F. Guo).

https://doi.org/10.1016/j.applthermaleng.2025.126106
Received 13 October 2024; Received in revised form 10 February 2025; Accepted 27 February 2025
Available online 18 March 2025

Nomenclature

Physical Parameters

$T_{pri,chws}$	Supply chilled water temperature in the primary loop [°C]
$T_{sec,chws}$	Supply chilled water temperature in the secondary loop [°C]
$T_{sec,chwr}$	Return chilled water temperature in the secondary loop [°C]
T_{ma}	Mixed air temperature [°C]
T_{oa}	Outdoor air temperature [°C]
T_{ra}	Return air temperature [°C]
T_{sa}	Supply air temperature [°C]
T_{za}	Zone air (indoor) temperature [°C]
D_{oa}	Outdoor air damper opening (0~1)
D_{ra}	Return air damper opening (0~1)
D_{ea}	Exhaust air damper opening (0~1)
$\dot{V}_{pri,chw}$	Volume flow rate of the primary chilled water loop [m^3/s]
$\dot{V}_{sec,chw}$	Volume flow rate of the secondary chilled water loop [m^3/s]
$\dot{V}_{sa}$	Volume flow rate of supply air [m^3/s]
$\dot{V}_{ra}$	Volume flow rate of return air [m^3/s]
f_{sa}	Supply fan speed (0~1)
$T_{sp,ch}$	Chilled water temperature setpoint [°C]
ϕ_{oa}	Outdoor air relative humidity
ϕ_{za}	Zone air (indoor) relative humidity
P_{ch}	Chiller power [W]
$P_{pri,chwp}$	Primary water loop pump power [W]
$P_{sec,chwp}$	Secondary water loop pump power [W]
P_{sa}	Supply fan power [W]
$u_{sec,cv}$	Cooling coil valve opening (0~1)
$\dot{Q}_{plug}$	Plug load [W]
$\dot{Q}_{light}$	Lighting load [W]
$\dot{Q}_{occ}$	Occupant sensible load [W]

Abbreviations

HVAC	Heating, ventilation, and air conditioning
RL	Reinforcement learning
DQN	Deep Q-network
DDPG	Deep deterministic policy gradient
AHU	Air handling unit

Symbols in the RL algorithms

s	State
a	Action
r	Reward
γ	Discount factor
$Q(s,a)$	Action-value function
π	Policy
θ^Q	Parameters of the Q-network
θ^π	Parameters of the actor (policy) network
$L(\theta^Q)$	Loss function for the Q-network
α	Learning rate

approach to control HVAC systems and improve building performance. Compared to MPC, RL can continuously learn by itself and improve its policies based on feedback from the environment, and it is less dependent on an accurate prediction model of the building and its HVAC systems. Also, RL is suitable for high-dimensional control problems with nonlinear dynamics and multiple control objectives, which are common in building energy systems.

This paper proposes an RL-based controller for a central HVAC system with multiple control variables and evaluates its performance in a building test facility. Moreover, different types of RL controllers are compared using simulation, providing useful suggestions for practical application. The remainder of the paper is organized as follows: Section 2 first introduces key concepts and typical categories of RL algorithms and then provides a literature review of RL applications in building energy systems. After that, the main contributions of this paper are summarized from the identified research gap. In Section 3, the performance of a value-based RL controller is demonstrated in an experimental test. Then, Section 4 compares the pros and cons of value-based and actor–critic RL algorithms on the control of central HVAC systems by simulation. Finally, Section 5 concludes the paper.

2. Related work on RL

2.1. Overview of RL algorithms

Reinforcement learning (RL) is a sub-field of machine learning that focuses on how agents should take actions in an environment to maximize the cumulative reward [6,7]. Unlike supervised learning where the model learns from a dataset of labeled examples, RL involves learning through interaction, using trial and error to discover optimal behaviors. The agent receives feedback through rewards or penalties to guide the learning process. Key components of an RL agent include states, actions, and rewards. The objective of the agent is to develop a policy, a strategy that defines the best action to take in each state to maximize its long-term rewards. Applications of RL span various domains, including robotics, gaming, finance, and healthcare, where decision-making is crucial under uncertainty.

Reinforcement learning algorithms can be broadly categorized into model-based and model-free methods. Model-based algorithms involve building a model of the environment, which is then used to simulate and plan actions. Those algorithms can be sample-efficient as they leverage the model to make more informed decisions. Model-free methods learn the optimal strategy directly from the rewards, which can be further classified into value-based, policy-based, and actor–critic methods. Value-based algorithms focus on estimating the action-value function (also called the Q-function), which represents the expected cumulative reward of state–action pairs:

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{k=0}^{T}\gamma^k r_{t+k} | S_t = s, A_t = a\right] \tag{1}$$

where T is the final time step and γ is the discount factor. The Bellman optimality equation shows that the optimal Q-function can be written as follows:

$$Q_\pi^*(s_t,a_t) = r(s_t,a_t) + \gamma \max_{a_{t+1}} Q_\pi^*(s_{t+1},a_{t+1}) \tag{2}$$

A prominent example of the value-based function is Q-learning, which learns the value of every action choice in given states to derive an optimal policy. Deep Q-Network (DQN) is a typical Q-learning algorithm. It represents the Q-function with an artificial neural network. During the training process, parameters (θ^Q) in the Q-function can be updated by:

$$L(\theta^Q) = L\left(r(s_t,a_t) + \gamma \max_{a'} Q_\pi(s_{t+1},a'), Q_\pi(s_t,a_t)\right) \tag{3}$$

$$\theta^Q \leftarrow \theta^Q - \alpha \nabla_\theta L(\theta^Q) \tag{4}$$

Table 1
Classification of RL applications to building HVAC systems.

Reference	Approach	Class of algorithm	Type of action	Actions
Ahn and Park (2020) [9]	Simulation	Value-based	Discrete	Outdoor air damper opening, chilled water temperature setpoint, cooling water temperature setpoint
Biemann et al. (2021) [10]	Simulation	Actor–critic	Continuous	Zone air temperature setpoint, fan mass flow rate
Deng et al. (2022) [11]	Simulation	Value-based	Discrete	Zone air heating/cooling setpoint
Dey et al. (2023) [12]	Simulation	Policy-based	Discrete	Zone air temperature setpoint
Du et al. (2021) [13]	Simulation	Actor–critic	Continuous	Zone air temperature setpoint
Fan et al. (2025) [14]	Simulation	Value-based	Discrete	AHU valve opening in each zone
Fu et al. (2023) [15]	Experiment	Value-based	Discrete	Zone air temperature setpoint
Gao et al. (2020) [16]	Simulation	Actor–critic	Continuous	Zone air temperature and humidity setpoint
Gao et al. (2024) [17]	Experiment	Actor–critic	Continuous	Supply air temperature, supply air flow signal
Gao and Wang, Wang et al. (2023) [18,19]	Simulation	Value-based, Actor–critic	Discrete, Continuous	Heat pump modulating signal
Guo et al. (2025) [20]	Simulation	Actor–critic	Hybrid	Outdoor air damper opening, chilled water temperature setpoint, fan speed, air purifier on/off
Liu et al. (2024) [21]	Simulation	Actor–critic	Continuous	Zone air heating/cooling setpoint
Qin et al. (2023) [22]	Experiment	Value-based	Discrete	Zone air temperature setpoint
Shin et al. (2024) [23]	Simulation	Value-based	Discrete	Heat pump on/off, zone air heating/cooling setpoint
Solinas et al. (2024) [24]	Simulation	Actor–critic	Continuous	Supply air temperature, supply air mass flow rate
Touzani et al. (2021) [25]	Experiment	Actor–critic	Continuous	Zone air heating/cooling setpoint, battery charge/discharge setpoint
Xia et al. (2023) [26]	Simulation	Value-based	Discrete	Occupant recommendation, zone air setpoint change, temperature and lighting relaxation
Wang et al. (2024) [27]	Simulation	Value-based	Discrete	Logical action that determines the zone air temperature setpoint range
Yang et al. (2021) [28]	Simulation	Value-based	Discrete	Zone air volume flow rate, power of the cooling/heating system
Yoon and Moon (2019) [29]	Simulation	Value-based	Discrete	Zone air temperature setpoint, supply air flow rate, humidifier on/off
Zhuang et al. (2023) [30]	Experiment	Actor–critic	Continuous	Damper position, supply air flow, temperature setpoint, fan speed

where $L(y, \hat{y})$ is a loss function which measures the difference between y and $\hat{y}$, and α is the learning rate. Finally, the optimal control action a_t^* at a state s_t is calculated by:

$$a_t^*(s_t) = \underset{a_t}{\operatorname{argmax}}\, Q(s_t, a_t) \tag{5}$$

Policy-based reinforcement learning algorithms, on the other hand, directly optimize a policy that maps states (s_t) to actions (a_t), without requiring the estimation of Q-functions. Those algorithms parameterize the policy function and adjust its parameters to maximize the expected reward. A classical policy-based algorithm is REINFORCE [6], which updates the parameters (θ^π) based on the policy gradient:

$$\theta^\pi \leftarrow \theta^\pi + \alpha \mathbb{E}_\pi \left[\sum_{k=0}^{T} \gamma^k r_{t+k} \nabla \ln \pi(a_t | s_t, \theta) \right] \tag{6}$$

Because REINFORCE suffers from unstable policy update issues, the Proximal Policy Optimization (PPO) algorithm [8] is later proposed to improve the training stability and robustness of REINFORCE. Compared with model-based methods, policy-based methods are particularly useful in high-dimensional or continuous action spaces. They can also naturally handle stochastic policies, providing better exploration and robustness in uncertain environments. By directly focusing on policy optimization, these algorithms can often converge faster and with greater stability in complex tasks.

The actor–critic algorithms are hybrids of the value-based and policy-based algorithms. Those algorithms learn two separate models simultaneously: an actor and a critic. The actor is responsible for selecting actions based on a parameterized policy, while the critic evaluates these actions by estimating the Q-function. This dual approach allows the actor to improve its policy based on feedback from the critic, which provides more stable and lower-variance gradient estimates than purely policy-based methods. Actor–critic methods can efficiently handle continuous action spaces and are known for their ability to converge more rapidly and stably in complex environments. Some typical examples are Deep Deterministic Policy Gradient (DDPG) [31], Advantage Actor–Critic (A2C), and Asynchronous Advantage Actor–Critic (A3C) [32].

2.2. Applications in building energy systems

In the field of building controls, RL has been applied to various subjects, including HVAC systems, batteries, appliances, domestic hot water, thermal energy storage, combined heat and power, windows, and lighting. From a survey conducted in 2020, most research tested their RL methods in the simulation environment, and only about 11% of research applied RL algorithms in real buildings [33]. Value-based RL methods are the most widely used approaches for building controls, which account for over 75% of research studies, followed by actor–critic methods which account for about 15%. But the survey also showed that policy-based and actor–critic methods have become increasingly more popular in recent years. For RL applications to the control of HVAC systems, a recent study in 2022 showed that the majority of RL methods output binary (e.g., the RL agent may select between ON/OFF or increase/decrease actions) or discrete actions (e.g., the RL agent may select an action from a few possible choices), while only a small fraction of RL methods make continuous (e.g., the RL agent needs to determine the values from a few continuous variables) or hybrid actions [34]. Table 1 categorizes the related research papers.

2.3. Research gaps and contribution

From the literature review in Table 1, a clear research gap is that most RL methods for building controls are still in the research stage. They are rarely implemented and tested for real HVAC equipment in real buildings. Also, although some researchers compared several algorithms within the category of value-based or actor–critic methods, almost no research has made a detailed comparison between the pros and cons of value-based and actor–critic methods. However, depending on the control object and the state and action spaces, one type of

Table 2
Basic settings of the office room used for the experiment.

Parameter	Value
Room area	60 [m^2]
Full load of the equipment (plug load)	865 [W]
Full load of occupants	720 [W]
Full load of lights	465 [W]

Fig. 1. Layout of the FLEXLAB test cell. It is configured as an office room in this experimental demonstration.

method could be superior to the other. Based on these research gaps, this study contributes to the RL applications as follows:

- Field demonstration is conducted in a real building test facility to illustrate the performance of the value-based RL method (i.e., DQN).
- The limitation of the value-based RL method is identified from the field demonstration results.
- Performance of the value-based and actor–critic RL methods are compared in a simulation model for an office building to control both temperature and humidity with in-depth analysis and suggestions.

3. Experimental study with DQN

This section demonstrates the performance of the DQN controller in a building test facility. From the literature review (Table 1), the DQN controller was widely applied for the building application, presumably due to its structure to encode the control variables into its decision variable. Section 3.1 details the experimental setup, including the layout of the test facility, configuration of the HVAC system, and daily schedules of occupants and equipment. Section 3.2 introduces the design of the DQN controller, including the state and action variables and the reward function. Section 3.3 shows the baseline control, which is used for comparison. Finally, Sections 3.4 and 3.5 show the test results and discuss the success and limitations of the DQN controller.

3.1. Experimental setup

The performance of the RL algorithm is assessed in a test cell located in FLEXLAB [35], an experimental test facility for evaluating building technologies at Lawrence Berkeley National Lab. As shown in Fig. 1, the test cell is configured as a real office room with six cubicles, and each cubicle has a desk, chair, equipment, and mannequin. The equipment and mannequins are both programmed as heat sources to simulate the heat generation of computers and occupants. Their schedules along with the lighting schedule are shown in Fig. 2. Key experimental settings, including the room area and full heat loads, are listed in Table 2.

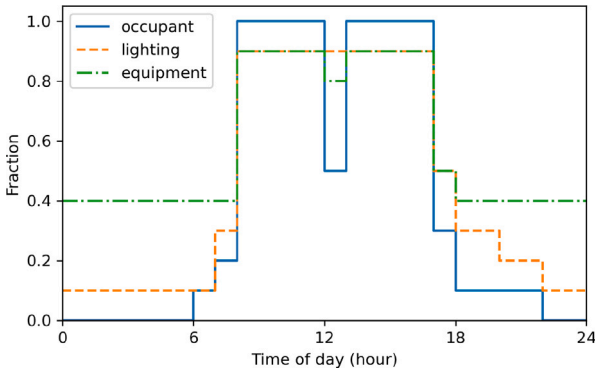


Fig. 2. Daily schedules of the equipment, occupant, and lighting in the office room. The schedules are acquired from the U.S. Department of Energy reference building EnergyPlus models [36].

Fig. 3. Outside view of the chiller and chilled water system of the FLEXLAB test cell.

Fig. 3 is an outside view of the chilled water system of the test cell, and its configuration together with the air handling unit (AHU) is shown in Fig. 4. The system is comprised of a primary and a secondary chilled water loop, and a single-duct variable air volume AHU. In the water loop, the air-cooled chiller controls the chilled water temperature to match a setpoint (T_{csp}), which can be specified by a higher-level controller. In the air loop, the supply air temperature setpoint ($T_{sp,sa}$) is adjusted by the return air temperature (T_{ra}), and the cooling coil valve ($u_{sec,cv}$) is regulated to match the supply air temperature (T_{sa}) to the setpoint. The supply air fan (f_{sa}) and the outdoor air damper (D_{oa}) will both be controlled by the higher-level controller. The exhaust, return, and outdoor air dampers are interlocked. The zone air cooling setpoint (T_{csp}) is fixed during the occupied time and has a setback during the unoccupied time.

Additionally, the experiment injected CO_2 into the cell during the occupied time to mimic the CO_2 generated by the occupants. Considering the amount of CO_2 generated by each occupant is around 5.2×10^{-6} m^3/s [37] and there are 6 occupants, 112 litter of CO_2 was injected into the room per hour from 8 am to 12 pm and from 1 pm to 5 pm.

The experiment was conducted from August to October 2023 in Berkeley, California, U.S.A. The baseline controller ran for 22 days and the RL controller ran for 9 days. The next two subsections will detail the design of the RL and baseline controllers.

3.2. Design of a simple DQN controller

According to Table 1, value-based algorithms are prevalent among RL applications, especially when the control action spaces are discrete and relatively simple. This study selected the Deep Q-Network

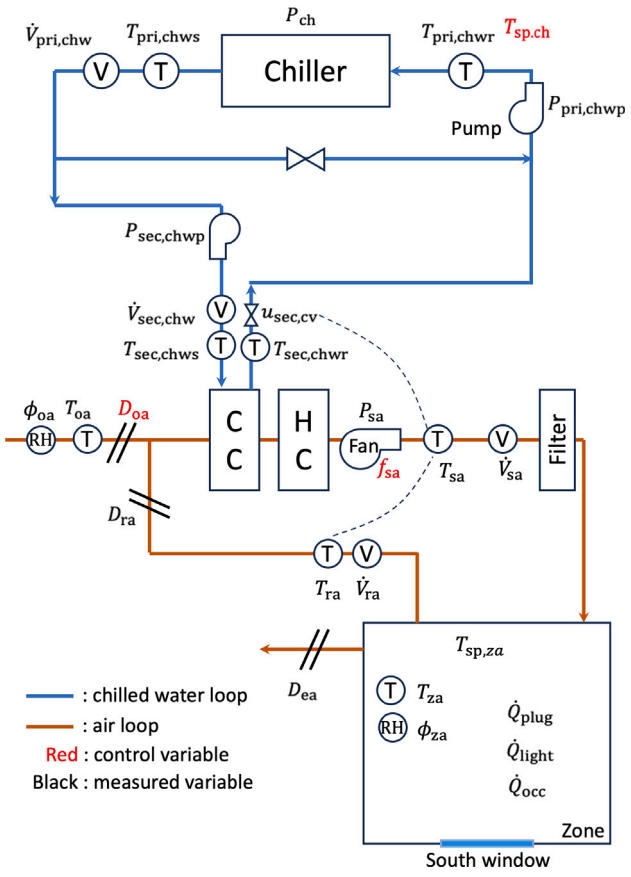


Fig. 4. Configuration of the HVAC system in the FLEXLAB test cell. The HVAC system has an air-cooled chiller, primary and secondary chilled water loop, and a variable air volume AHU.

(DQN) [38], a typical value-based algorithm, as the RL method to evaluate the feasibility of the RL controller in a real building. DQN was chosen for its relatively simple structure and straightforward action space design. Different from other Q-learning algorithms, DQN represents the Q-function with artificial neural networks. Additionally, to overcome the instability problems in training, DQN uses two neural networks: a Q-network updated at each iteration and a target Q-network updated at every fixed number of iterations. The neural network has an input layer, two hidden layers each with 128 neurons, and an output layer that generates the Q-value for each discrete action. The hyperparameters are chosen based on an iterative approach. Simply, we increase the size of the network and choose the smallest one that does not show further performance improvements. The states of the DQN controller include the time of day, outdoor air temperature (T_{oa}), outdoor air humidity ratio (ω_{oa}), incident solar radiation (q_{sol}), zone air temperature (T_{za}), and zone air humidity ratio (ω_{za}). The DQN was trained by using an environment, called a hybrid simulator. The hybrid simulator is a simulation model that combines a building gray-box model and an HVAC model. The model was calibrated based on the measurement data, and the details of the model can be found in Section 4.1.

Table 3 shows the variables controlled by DQN. The zone air temperature setpoint is fixed at 22 °C when the office is occupied, which is very common in U.S. office buildings. The setpoint is not adjustable by DQN because we investigate the energy-saving potential by coordinating equipment, e.g., coordinating compressors and fans, rather than increasing the zone air setpoint. The supply air temperature setpoint will be switched to 12.8 °C when the thermostat requests for cooling. Because the test cell is arranged as a single zone, this setpoint value is not adjustable by the DQN controller. In our initial plan, the supply air fan speed, outdoor air damper opening, and chilled water temperature setpoint are controlled by the DQN algorithm. But after discretizing the three control variables and then enumerating their combinations, the action space becomes relatively large, which makes training very difficult. Therefore, we plan to investigate the performance of DQN with large action spaces using simulation (detailed in Section 4), and develop a relatively simple DQN controller for the experiment. During the occupied time, the supply air fan speed is fixed at 30%, the damper can open at 20% or 100% (full economizer mode), and the chilled water temperature can be set at 6.7 °C or 10 °C. During the unoccupied time, the fan speed is 10%, and the damper can either close or have a 20% opening. The size of the action space is 4.

The reward function has two components: total energy consumption (r_E) and temperature violation (r_T). They are expressed as follows:

$$r_t = \begin{cases} r_E + \alpha_1 r_T, & \text{from 6AM to 10PM} \\ r_E, & \text{from 10PM to 6AM} \end{cases} \tag{7}$$

where

$$r_E = -(E_{chiller} + E_{pump} + E_{fan}) \tag{8}$$

$$r_T = -([T_{za} - T_u]_+ + [T_l - T_{za}]_+) \tag{9}$$

The r_E term adds up the energy consumption of the chiller, fan, and pumps in a time step (unit: kWh). The r_T term penalizes the RL controller if the zone air temperature rises above T_u or falls below T_l, where T_u and T_l are determined based on the zone air temperature setpoint:

$$T_u = T_{csp} + 0.5°\text{C} \tag{10}$$

$$T_l = T_{csp} - 1°\text{C} \tag{11}$$

No penalty is imposed if the zone air temperature is within the boundary. The α_1 in Eq. (7) determines the importance of r_T compared to r_E. Since occupant comfort is usually prioritized over energy savings, we impose a penalty on r_T that is 10 times larger than r_E when the zone air temperature goes beyond the boundary. For this HVAC system, the magnitude of r_E is usually between 0 and 0.25 kWh, so α_1 is set to 2.5.

The DQN algorithm is first pre-trained using a calibrated simulation model (discussed in Section 4), and then applied to control the HVAC system in the test cell. Control commands are sent every 15 min.

3.3. Design of the baseline controller

For comparison, a rule-based controller is adopted as the baseline (Fig. 5). During the occupied time (from 6 AM to 10 PM), the zone air temperature setpoint is fixed at 22 °C, the supply air temperature setpoint is 12.8 °C, supply air fan operates at 30% of its full speed, the outdoor air damper maintains at its minimum adjustable position of 20% opening, and the chilled water temperature setpoint is set at 6.7 °C. During the unoccupied time (from 10 PM to 6 AM), the zone air temperature setpoint is reset to 29 °C, the supply air fan operates at only 10% of its full speed, and the outdoor air damper is fully closed. As shown in Fig. 5, according to the cooling requirement (zone air temperature vs. cooling setpoint), the chilled water temperature and supply air temperature swing to provide cooling.

3.4. Test results: successful control of RL

To guarantee convergence, the RL controller was trained on selected 4 days of data showing a wide temperature range based on the local weather data, and the training was repeated several times. While the training result is not always consistent, we found it reached convergence after 400–500 episodes as shown in Fig. 6.

Table 3
The choices of control variables for the simple DQN controller used in the experiment.

	Occupied time (6AM–10PM)	Unoccupied time (10PM–6AM)
Zone air temperature setpoint	22 °C	29 °C
Supply air temperature setpoint	12.8 °C (55 °F)	12.8 °C (55 °F)
Supply air fan speed	30%	10%
Outdoor air damper opening	20%, 100%	0, 20%
Chilled water temperature setpoint	6.7 °C (44 °F), 10 °C	6.7 °C (44 °F), 10 °C

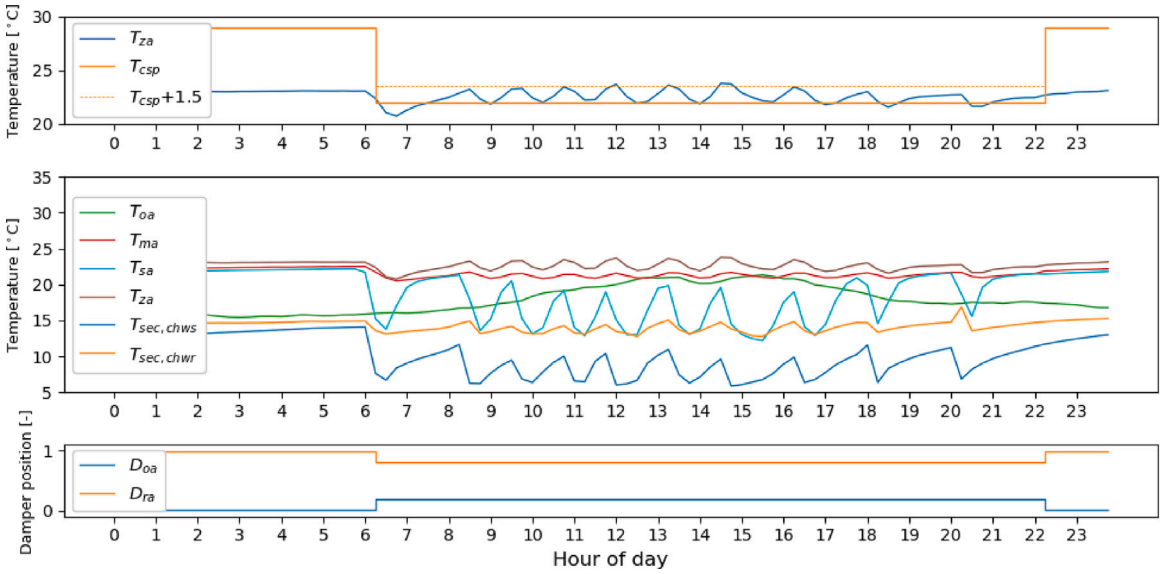


Fig. 5. Typical operational characteristics of baseline control on one day. The figure shows the zone temperature (T_{za}), zone cooling setpoint (T_{csp}), outdoor temperature, air temperature in the AHU, chilled water temperature, and damper opening. The occupied time is between 6 AM and 10 PM. During this time, the cooling setpoint is reduced to 22 °C and the outdoor damper maintains at 20% opening. The chiller and supply fan start operating to cool the room.

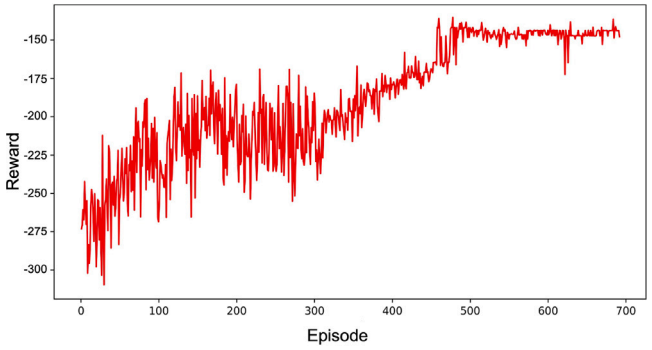


Fig. 6. Reward profile for DQN training. The DQN reaches convergence after training for about 500 episodes.

For the comparison, we visualized the operational characteristics of the RL controller in Fig. 7. This day was chosen based on weather similar to that in Fig. 5. On this day, the outdoor air temperature was mild, and the RL tried to provide free cooling (economizer operation) by opening the outdoor air damper. When the operation started at 6 AM, the cooling operation was initiated, and the chilled water was provided. At the same time, the RL controller tried to adjust the damper position to provide enough cooling load. Once the cooling operation is stabilized, the OA damper is fully opened during the daytime.

Fig. 8 shows the overall experimental results between the energy consumption of the HVAC system under the control of RL and baseline algorithms. We split the experimental data into two pieces (Mild and Hot days) as the HVAC system showed different operational characteristics. When the outdoor climate is mild (Mild days) with a daily average temperature below 24 °C, the HVAC system consumes less electricity under the RL controller at the same outdoor temperature. In this situation, the RL agent can enable the economizer mode to cool the room passively most of the day and increase the chilled water temperature setpoint to 10 °C. As the RL agent requests a higher chilled water temperature, the COP of the chiller will increase, leading to more energy savings. In order to compare their energy consumption under the same outdoor conditions, linear regression is performed using data from the Mild days, and the energy performance of each controller is evaluated on the days when the other controller is applied. The regression shows that, after assessing the energy performance of the two controllers in all Mild days (21 days with baseline control and 7 days with RL control), 16.8% energy saving can be achieved by using the RL controller.

However, as shown by the three upper-right data points (Hot days) in Fig. 8, when the outdoor temperature is high, the cooling load will be very high and the HVAC system needs to work hard to avoid the penalty of comfort violation. In this case, the economizer mode has to be turned off and the chilled water temperature has to be reduced to provide enough cooling load. As a result, the RL controller barely has any advantage compared to the baseline. This behavior is actually viewed as a failure of the control, and we present a more detailed analysis in the following Section 3.5.

Fig. 9 compares the indoor CO_2 concentration at occupied time when the daily averaged outdoor temperature is below 24 °C, and Fig. 10 shows the daily variation. When the economizer mode is enabled in RL control, the CO_2 concentration in the room will be diluted by the outdoor air, which can be considered an additional benefit.

3.5. Test results: limitations of RL

As discussed in the previous section, the RL controller may not give clear energy benefits over the Baseline controller for hot days as the

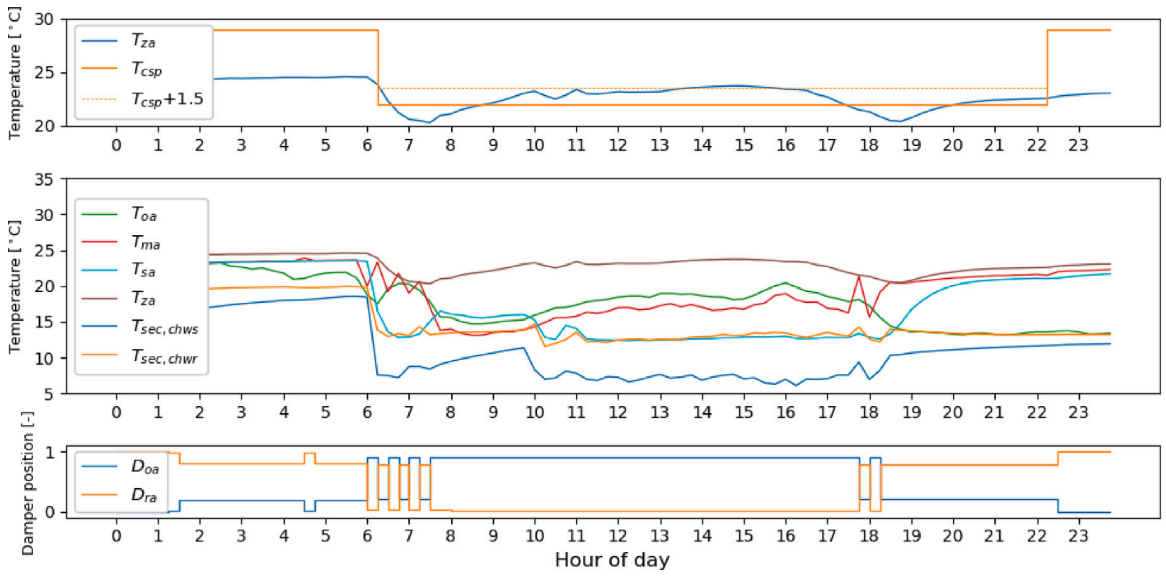


Fig. 7. Typical operational characteristics of RL control on a Mild day. During the occupied time between 6 AM and 10 PM, the room temperature is maintained around the cooling setpoint, and outdoor damper (D_{oa}) is fully opened to help passively cool the room..

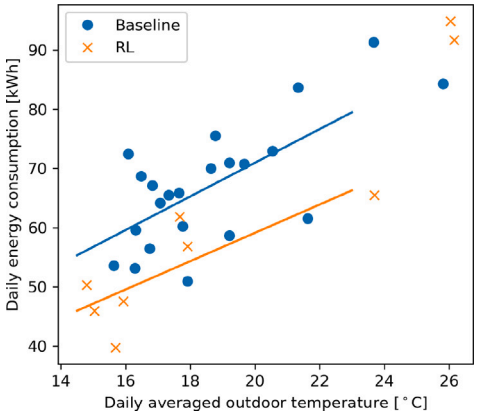


Fig. 8. Comparison of energy consumption between the RL and the baseline controller. The blue and orange curves are respectively linear fits of the baseline and RL data on the Mild days (daily average temperature below 24 °C).

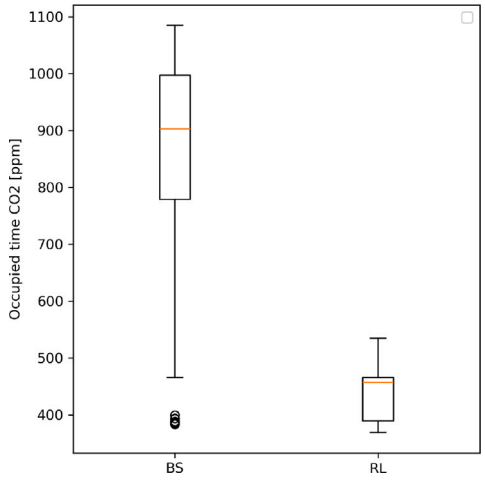


Fig. 9. The box plot compares the indoor CO_2 concentration between RL and baseline on the Mild days (BS: baseline control, RL: reinforcement control).

economizer operation is not available. However, since we have another control variable, a chiller water setpoint, we can expect some minor savings from adjusting the chilled water setpoint, especially, during the time when the cooling load is not full-capacity.

Fig. 11 shows the operational characteristics of the RL control on a hot day. In the early morning (6AM–10AM), the RL controller tried to utilize free cooling by adjusting the damper operation. In addition, it tried to use a higher chilled water setpoint to reduce chiller energy consumption. However, when the outdoor air temperature increased in the late afternoon, the high chilled water temperature was not enough to meet the cooling load even with the minimum outdoor air position near noon. In this case, the RL controller needed to lower the chilled water setpoint, but it did not reduce it, and our safety algorithm reduced it after one hour. The safety algorithm is designed to prevent the worst-case scenario. For example, when the temperature violation is severe, the operation force to the minimum outdoor air position and low chilled water setpoint.

Due to the black-box nature of the RL controller, we cannot clearly explain why this occurred. However, we have several hypotheses regarding the unintended outputs:

1. The training data may not cover a sufficiently wide range of hot air conditions, as the weather in FLEXLAB is typically hottest between September and October. Although we included days with similar (or slightly cooler) weather conditions in the training data, it may not have been sufficient for the RL controller to fully explore the action space.
2. Under hot conditions, the supply air temperature fails to meet the setpoint, and the situation worsens if the room temperature also falls short of the setpoint. Specifically, once the zone setpoint is violated, the mixed air temperature rises, increasing the cooling demand to meet the supply air setpoint. In this scenario, the higher chilled water temperature is insufficient to reach the supply air setpoint. This is likely a rare or unseen condition in the hybrid simulator used for RL training.
3. The quality of the training may not be truly optimal. For this experiment, we designed the RL controller with very limited action spaces to simplify the problem and ensure optimal operations with faster training. However, even with these constraints, the controller's convergence was unstable during training, and its performance was sometimes sub-optimal. Also, because of the

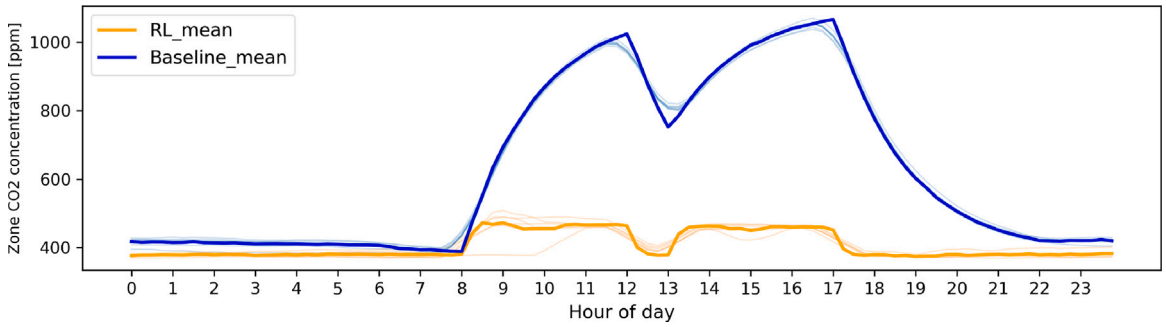


Fig. 10. The light blue and light orange curves are the daily variations of the CO_2 concentration under the RL and baseline control on the Mild days. The dark blue and dark orange line are the average of all days.

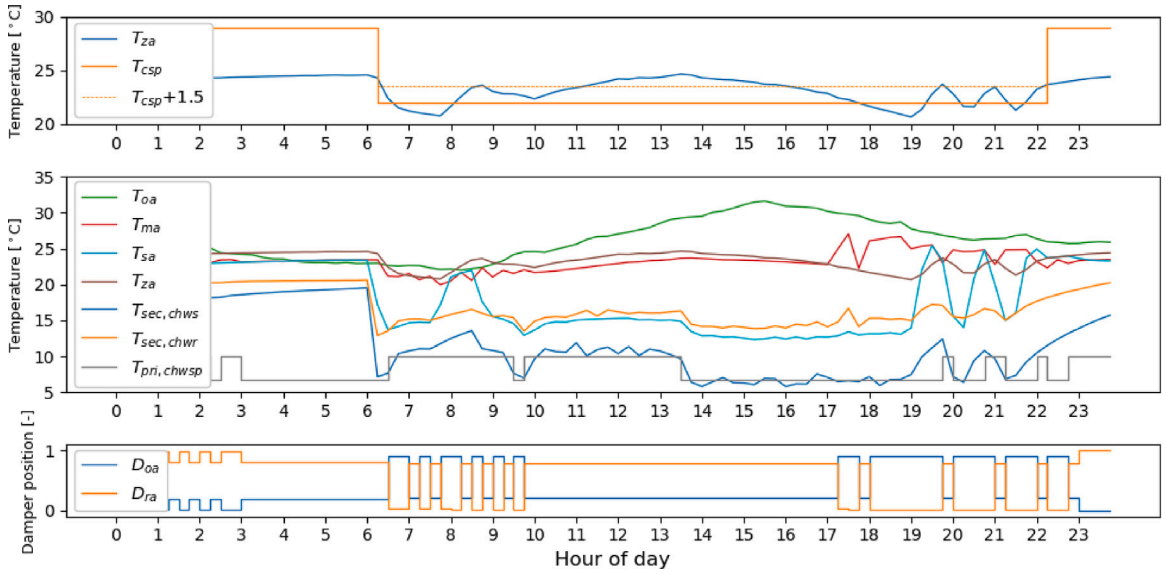


Fig. 11. Typical operational characteristics of RL control on a Hot day. The room temperature is not properly controlled around the cooling setpoint because RL failed to reduced the chilled water temperature setpoint to meet the cooling load.

small action spaces, it may be possible that the RL controller does not capture the continuous operational characteristics between the actions and the outcomes. While part of this could be attributed to the mismatch between the hybrid simulator and the real building, it also stems from convergence issues within the RL algorithm itself.

4. Simulation: comparing DQN with DDPG

In the last section, we evaluated the experimental performance of a DQN controller and showed that compared with a ruled-based baseline controller, DQN can achieve energy savings by appropriately adjusting the damper position and the chilled water temperature setpoint. However, we also observed the unexpected behavior of the DQN controller, and we identified this may be attributed to the training performances. In terms of the training, the drawback of the DQN is clear. In each step, DQN is required to evaluate the Q-function for all possible actions to select the best one. Hence, it is only computationally feasible for a small action space with discrete values. However, an HVAC system usually has multiple continuous control variables, so those variables have to be discretized, which may cause the HVAC system to reach sub-optimal operating conditions.

This section will evaluate the performance of the DQN controller when its action space is relatively large. However, to overcome this disadvantage in the application to HVAC systems, we also propose an actor–critic algorithm termed Deep Deterministic Policy Gradient (DDPG). DDPG learns the Q-function (termed "critic") and the policy function (termed "actor") simultaneously, both represented by neural networks. The critic network is trained by minimizing the mean squared error between its Q-value predictions and the target Q-values, while the actor network is trained by maximizing the expected Q-value estimated by the critic network, using gradients derived from the critic network. The actor maps a state directly to the best action at this state, so the actor's output can be continuous variables. Thus, for the control problem in this paper, the action space of DDPG is only three, corresponding to the three control variables (fan, damper, and chilled water) in the HVAC system. This significant reduction of the action space can accelerate the training process a lot, and as shown in the following discussions, several issues in the design of DQN can be effectively avoided with continuous actions.

In this section, the performance of DQN and DDPG in every possible aspect is compared using simulation. Section 4.1 first introduces the hybrid simulator which is used to train and test the RL algorithms. Then, Section 4.2 details the design of DQN and DDPG algorithms to control the HVAC system. Finally, Section 4.3 compares the simulation results, including the computational time, reward, thermal comfort, control actions, and energy consumption. Meanwhile, some lessons learned and a few suggestions are also provided.

4.1. Development of the hybrid simulation model

The hybrid simulator comprises an HVAC system model, a building envelope model, and an indoor moisture transfer model. The HVAC

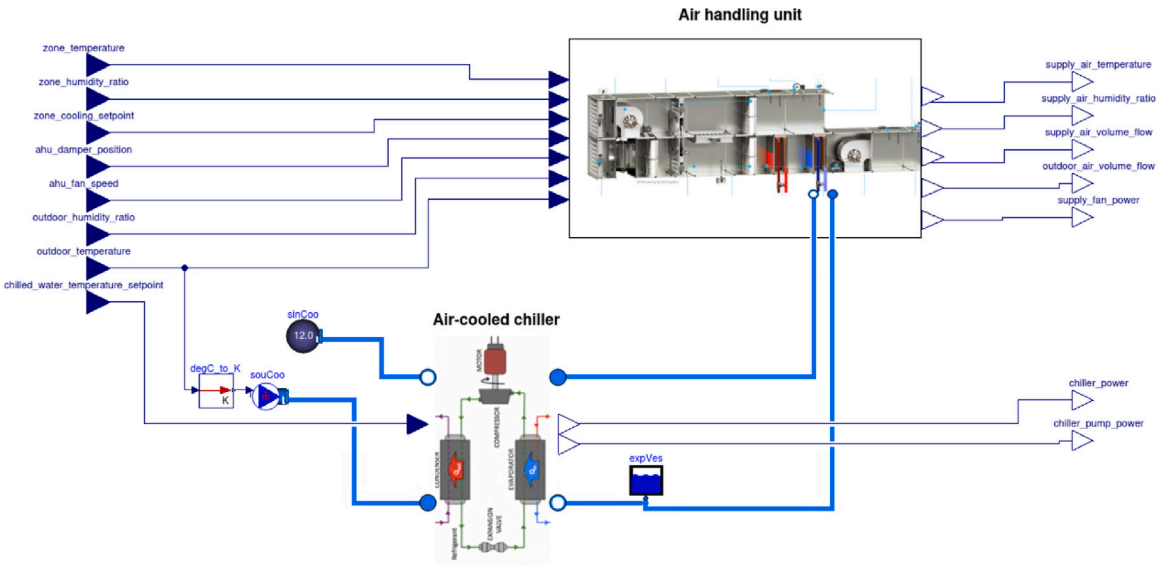


Fig. 12. Network scheme of the central HVAC system represented in Modelica.

system model is developed by the Modelica simulation tool using the Buildings library [39]. Shown in Fig. 12, it includes a chiller model, an AHU model, and the connected chilled water loop. After calibration, the model is exported as a Functional Mock-up Unit (FMU) [40] and then imported into Python by the PyFMI package [41].

A Resistance–Capacitance (2R2C) model is developed for the building envelope. The heat transfer equations are expressed as follows:

$$C_{za}\frac{\mathrm{d}T_{za}}{\mathrm{d}t} = \frac{T_w - T_{za}}{R_{zw}} + \frac{T_{oa} - T_{za}}{R_{zo}} + \dot{q}_{sol}\beta A_{win} + \dot{Q}_{hvac} + \dot{Q}_{gain} \tag{12}$$

$$C_w\frac{\mathrm{d}T_w}{\mathrm{d}t} = \frac{T_{za} - T_w}{R_{zw}} + \dot{q}_{sol}(1-\beta)A_{win} \tag{13}$$

where Eq. (12) represents the dynamics of the zone air temperature and Eq. (13) represents the dynamics of the building thermal mass. $\dot{q}_{sol}$ is the incident solar radiation that can be calculated from Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI), and Diffuse Horizontal Irradiance (DHI), $\dot{Q}_{hvac}$ is the sensible heating capacity provided by the AHU, and $\dot{Q}_{gain}$ is the combination of plug load, sensible occupant load, and lighting load. In total, 6 parameters in this gray-box model need to be determined through training, namely: C_{za}, C_w, R_{zw}, R_{zo}, β, and A_{win}.

The indoor moisture transfer is described by an ordinary differential equation (ODE):

$$\rho_a V\frac{\mathrm{d}\omega_{za}}{\mathrm{d}t} = \rho_a \dot{V}_{sa}(\omega_{sa} - \omega_{za}) + \dot{S}_\omega \tag{14}$$

where ω refers to the zone air or outdoor humidity ratio [kg H_2O/kg dry air], $\dot{S}_\omega$ refers to the moisture generation by occupants, which is set to 2.43×10^{-5} kg/s/person [42].

The HVAC system model and the RC model are both calibrated using the experimental data obtained during July and August, 2023. In the HVAC system model, the mean absolute percentage error (MAPE) of chiller, chilled water, pump, fan, and supply air are all lower than 10%. The root mean square error (RMSE) of the RC model is 0.54 °C.

4.2. Development of RL controllers

4.2.1. Design of a modified DQN controller

To further investigate the performance of DQN, in this simulation, we modified the DQN controller by increasing the number of its control actions. Shown in Table 4, the fan speed, damper opening, and chilled water temperature are discretized. During the occupied time, the fan speed can change from 10% to 40% (because the fan is oversized, the highest value is not 100%), the damper from 20% opening (minimum adjustable position) to 100% opening (economizer mode), and the chilled water temperature from 5 °C to 15 °C. Considering the combinations of the three control variables, the total action space for DQN is 48.

The ϵ-greedy approach is adopted for exploration in training. The initial exploration rate (ϵ) is 1.0, and as training proceeds, ϵ gradually decreases to a minimum value of 0.3. Note that because the action space is large, a higher exploration rate is necessary for the neural network model to learn the Q-value of each discrete action. Regarding other hyperparameters, the learning rate is set to 1×10^{-5}, the discount factor (γ) is 0.98, the size of the replay buffer is 5×10^5, and the mini-batch size is 128.

4.2.2. Design of the DDPG controller

The DDPG algorithm trains four neural networks, including a critic that maps the state and actions to a Q-value, an actor that maps states to the optimal actions, as well as a target critic and a target actor whose weights are slowly updated by the weights of the critic and the actor. Similar to DQN, the actor in DDPG also has an input layer, two hidden layers each with 128 neurons, and an output layer, but the output layer directly generates the optimal action of the three continuous control variables.

The states of the DDPG controller are the same as the DQN controller discussed in Section 3.2. The range of the control variables is shown in Table 5. The zone air and supply air temperature setpoints are still fixed. During the daytime, the fan speed can be controlled between 10% and 100%, the damper opening between 20% and 100%, and the chilled water temperature setpoint between 5 °C and 15 °C. The wide range of the three control variables allows the DDPG controller to make better decisions.

To balance exploration and exploitation, DDPG uses an Ornstein–Uhlenbeck (OU) process to generate noises in the output actions. The reversion rate is 0.15 and the standard deviation is 0.2 in the OU process. As to the hyperparameters for training, the learning rate is set to 2.5×10^{-5} for the actor and 5×10^{-5} for the critic, and the rate to update the target networks is 0.005. The discount factor, the size of the replay buffer, and the mini-batch size are the same as DQN.

4.2.3. Design of the reward function

Unlike the experiment, the indoor moisture transfer model is added to the simulation. Hence, the reward function is edited to include the

Table 4
The choices of control variables for the modified DQN controller in the simulation.

	Occupied time (6AM–10PM)	Unoccupied time (10PM–6AM)
Zone air temperature setpoint	22.2 °C (72 °F)	24.4 °C (76 °F)
Supply air temperature setpoint	12.8 °C (55 °F)	12.8 °C (55 °F)
Supply air fan speed	10%, 20%, 30%, 40%	10%, 20%, 30%, 40%
Outdoor air damper opening	20%, 47%, 73%, 100%	0, 10%, 20%, 30%
Chilled water temperature setpoint	5 °C, 10 °C, 15 °C	5 °C, 10 °C, 15 °C

Table 5
The ranges of control variables for the DDPG controller in the simulation.

	Occupied time (6AM–10PM)	Unoccupied time (10PM–6AM)
Zone air temperature setpoint	22.2 °C (72 °F)	24.4 °C (76 °F)
Supply air temperature setpoint	12.8 °C (55 °F)	12.8 °C (55 °F)
Supply air fan speed	10%~100%	10%~40%
Outdoor air damper opening	20~100%	0~30%
Chilled water temperature setpoint	5 °C~15 °C	5 °C~15 °C

penalty of violation of zone air relative humidity (r_ϕ):

$$r_t = r_E + \alpha_1 r_T + \alpha_2 r_\phi \tag{15}$$

where r_E and r_T has been shown in Eqs. (8) and (9), and

$$r_\phi = -[\phi_{za} - \phi_u]_+ \tag{16}$$

The r_ϕ term penalizes the controller when the zone air relative humidity is higher than ϕ_u, which equals 60%. Since the indoor environment is unlikely to have very low relative humidity in the hot summer, a lower bound of the relative humidity is not specified. To determine an appropriate value for α_2, the performance of humidity control by DQN and DDPG is evaluated with α_2 ranging from 2.0 to 10.0 using simulation. The difference in the control performance is small, and we finally set α_2 at 8.0, which is an optimal value for this building.

4.3. Comparison of simulation results

DQN and DDPG are applied to control the HVAC system by interacting with the hybrid simulation model. The weather data in July 2023 in Berkeley, California, U.S.A. are used for training the controllers, and the subsequent data in August 2023 are used for testing. This subsection compares the testing results from several aspects.

4.3.1. Computational time and reward

First of all, the computational time of the two algorithms is compared. The algorithms are trained multiple times on a workstation with CPU and GPU cores, where DQN is trained for 500 episodes and DDPG is trained for 300 episodes. Based on our calculation, on average, DQN takes about 125 s to finish one episode and DDPG takes about 100 s. This means the training of DDPG is faster when the number of episodes is the same. This is because DDPG directly outputs continuous actions from the actor network, avoiding the computationally expensive process of evaluating and selecting among a lot of discrete actions.

Fig. 13 compares the change of reward as the RL model is trained by an increasing number of episodes. In the first 300 episodes, the reward of DQN fluctuates severely as its model output is unstable. Although it discovers a good control strategy after 300 episodes, the strategy is still not optimal. DQN is hard to train when the action space is relatively large and the optimal strategy is not guaranteed to find. In comparison, DDPG takes fewer than 40 episodes to discover the optimal strategy, which is fast and robust. Because of its continuous action space handling and deterministic policy optimization, DDPG allows for more efficient exploration and exploitation. Also, its actor–critic architecture reduces the computational burden of evaluating every possible discrete action in the Q-function, leading to quicker convergence in high-dimensional environments. As a result, DDPG has huge advantages over DQN in computational efficiency when the action space is continuous and has multiple variables.

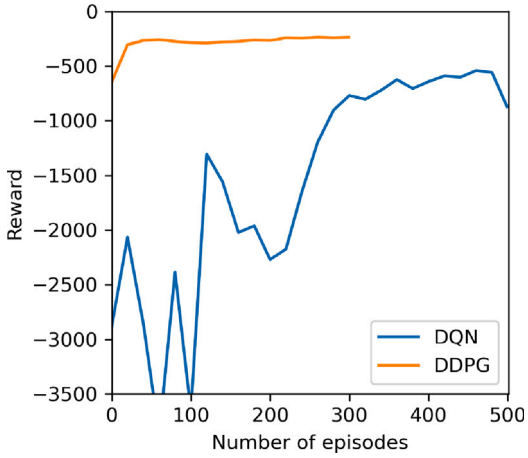


Fig. 13. A comparison of reward on the test set when DQN and DDPG are trained by different numbers of episodes.

4.3.2. Thermal comfort, control actions, and energy consumption

Next, the performance of the two algorithms is compared. The dynamics of the zone and the HVAC system in August (testing phase) are illustrated in Fig. 14 (DQN) and 15 (DDPG), where the shaded area indicates the unoccupied time. From the first plot of each figure, the temperature control of DQN is not as good as DDPG, as the zone air temperature sometimes even reaches around 25 °C. This is because DQN may fully open the outdoor air damper by mistake when the outdoor temperature is high (e.g., on August 7th and 30th). Also, from the second plot in Fig. 14, the humidity control of DQN is not satisfying as well, since the indoor relative humidity often rises above 60%. However, the DDPG controller can guarantee indoor thermal comfort almost all the time. The last three plots in each figure display the control commands of chilled water temperature, damper opening, and fan speed. DQN can only take several limited values for each control variable, which is usually not the optimal decision in most operating conditions. In comparison, DDPG can output continuous values in a wide range, allowing the algorithm to search for the optimal decision within a specified boundary. From the significant change of control actions each day, the optimal decision heavily depends on the operating condition.

Fig. 16 shows the zone air temperature and relative humidity in occupied time with DQN and DDPG controllers. In both cases, the zone air temperature setpoint is fixed at 22.2 °C. DDPG is capable of always maintaining the zone air temperature between 21 °C and 23 °C in August, but when the DQN controller is used, the temperature sometimes goes beyond the boundary, causing thermal discomfort for occupants. Besides, in DQN control, the indoor relative humidity rises

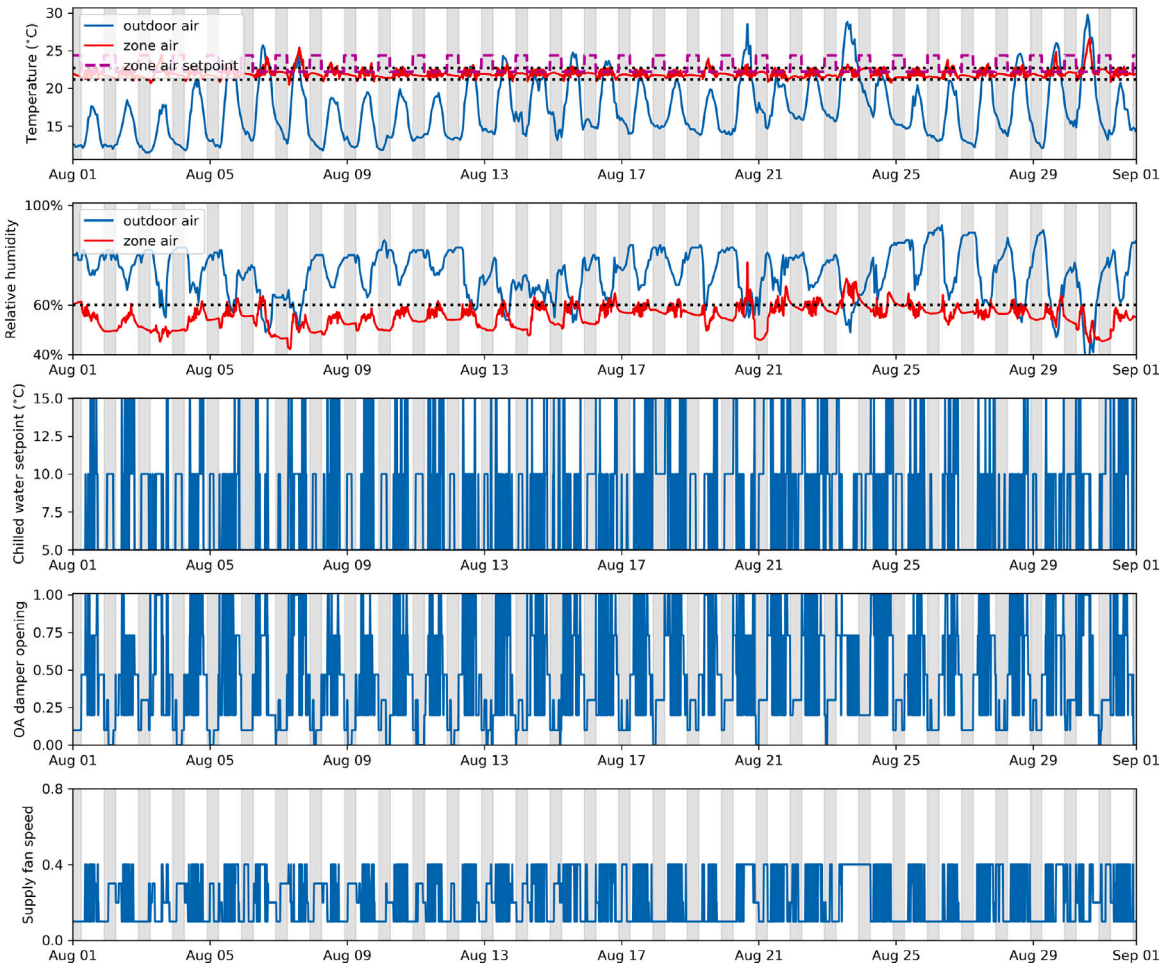


Fig. 14. Performance of the DQN controller on the test set. The temperature and humidity control are not satisfying as constraints are sometimes violated. DQN can only generate several discrete actions which may not be optimal.

above 60% in 17.4% of the time in August, but this number is 6.5% for the DDPG controller. This further proves that DDPG's humidity control is better than DQN's.

Fig. 17 compares the control actions of the two algorithms. The actions optimized by DDPG are usually not equal to the values in the action space of DQN. This result explains the limitation of DQN: its actions could be sub-optimal since the optimal action may stay somewhere between the discrete values in the action space. For example, even though the optimal damper opening is 30%, DQN can only send out a command of 20% or 47%, because the 30% opening is not in its action space. However, further discretizing the control variables is also not feasible, because training would be tough given a large action space. Additionally, due to the difficulty in training, DQN sometimes cannot find a good strategy. For instance, DQN often maintains the chilled water temperature at 5 °C, but the low setpoint is usually unnecessary since DDPG satisfies the thermal comfort with a much higher chilled water temperature. Fig. 18 compares the energy consumption. Because the chilled water temperature in DQN control is very low, the chiller has a lower efficiency and consumes more energy. As a result, the energy consumption of DQN is 74% higher than DDPG.

4.3.3. *In-depth analysis and lessons learned*

Based on the above results, the property of DQN that is only capable of generating discrete actions can cause a few issues in the control of HVAC systems. The following are some comments related to DQN:

1. Only limited choices are possible for each control variable. For example, in our study, the chilled water temperature only has three choices, and the fan speed and damper opening both only have four choices, but their combinations still add up to 48 actions, which is considered a relatively large space. Other researchers may also discretize the zone air temperature setpoint (see Table 1). DQN with a large action space is very difficult to train, because the Q-function must estimate the Q-values of all actions, which leads to an exponential increase in the number of state–action pairs and significantly increases the complexity and computational demands of the learning process.
2. Because choices are limited, the optimal value of a control variable usually cannot be reached. For example, the action space of the outdoor air damper opening is discretized to 20%, 47%, 73%, and 100%. If the optimal damper opening is 60%, then DQN can only output 47% or 73%, which is either too low or too high for the damper. In practice, DQN may even switch between different actions, causing unnecessary hunting in control (as shown in the fourth subplot of Fig. 14), which can affect the performance, stability, and safety of the HVAC system.
3. The discrete action space has to be determined before training the DQN, which could be difficult if the software engineer is unfamiliar with the HVAC system. Consider a case where the supply fan in the AHU is oversized. If the engineer does not have this information in advance, s/he may discretize the fan speed into 10%, 40%, 70%, and 100%. In this case, although the optimal fan speed could be between 10% and 30%, DQN will always output a value of 40% to avoid comfort-related penalties, wasting a lot of energy on the fan side.

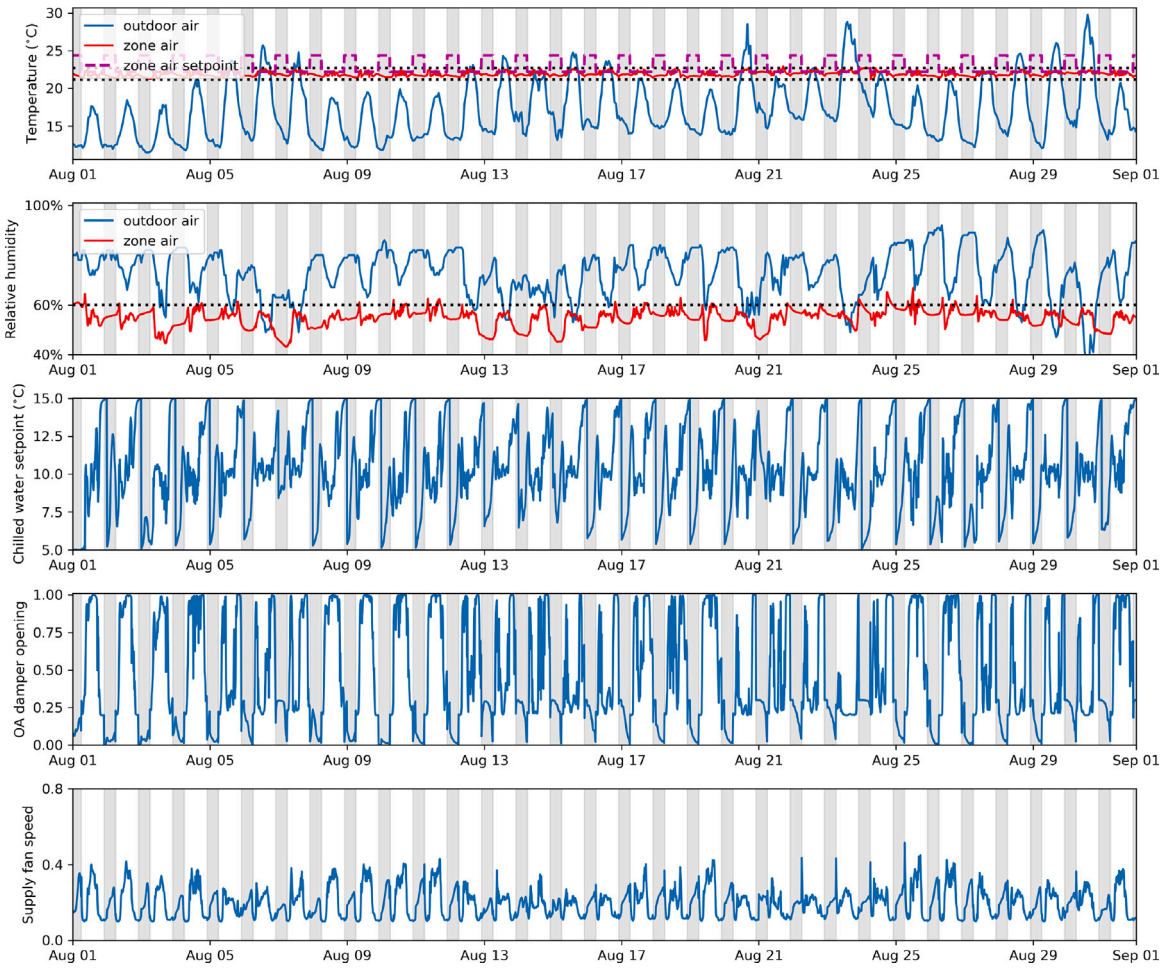


Fig. 15. Performance of the DDPG controller on the test set. The temperature and humidity control are much better than the DQN controller. DDPG can generate continuous actions based on the current operating condition.

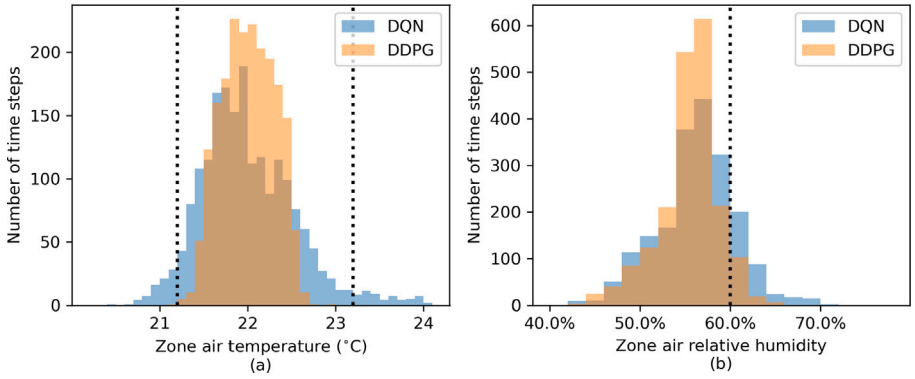


Fig. 16. Comparison of the zone air temperature and relative humidity between DQN and DDPG controllers (occupied time in the testing phase). On average, DDPG has better temperature and humidity control than DQN.

However, the structure of DDPG allows it to avoid the above three issues effectively. Here are a few points:

1. Because the actions of DDPG are continuous, the size of its action space is equal to the number of control variables. In this simulation, the number of actions is only 3, compared to DQN which has 48 actions. The low dimension of action space significantly reduces the computational time and difficulty in training. For example, our simulation shows that DDPG usually finds an optimal strategy after training for about 40 episodes. This efficient and rapid convergence makes DDPG particularly advantageous in the deployment of advanced controllers in HVAC systems.
2. Because DDPG uses policy gradient to search for the optimal action in an continuous action space, a wide boundary can be specified for each control variable which fully covers the equipment's adjusting range. For example, the dampers and fans can be adjusted from their minimum adjustable position/speed to a full opening/speed, and the chilled water temperature setpoint can be specified between 5 °C and 15 °C. This operational range is known during the design phase of the HVAC system, so

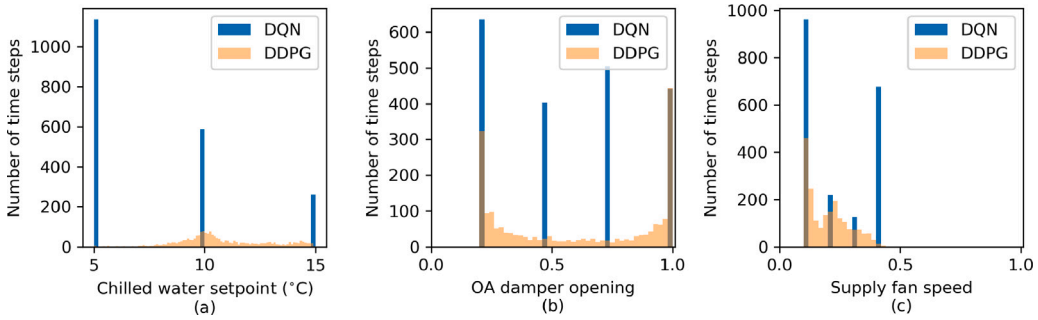


Fig. 17. Comparison of the control actions between DQN and DDPG controllers (occupied time in the testing phase). Note that DQN can only output discrete actions while the actions of DDPG are continuous.

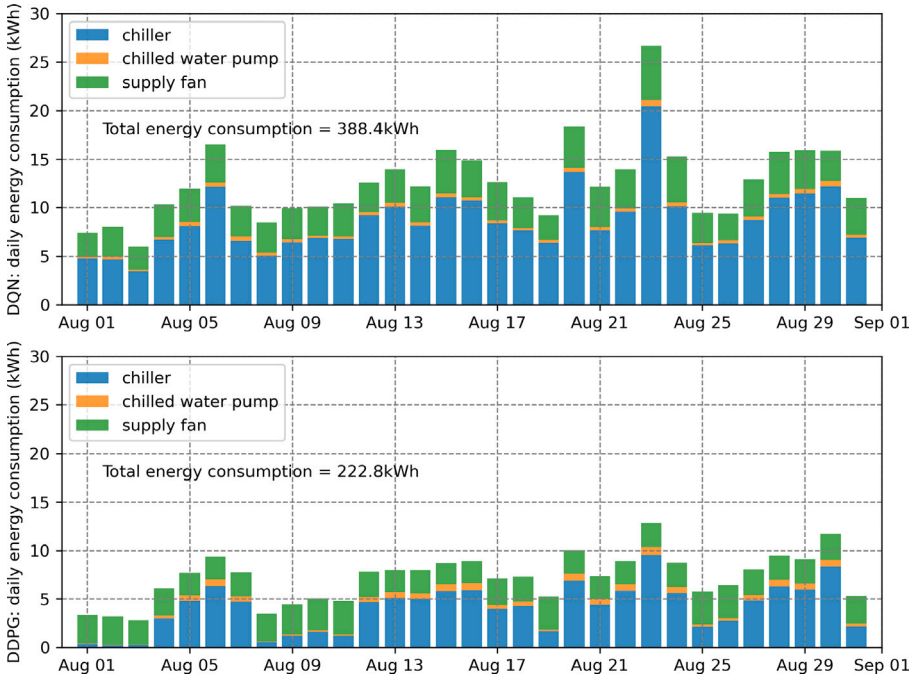


Fig. 18. Comparison of the total energy consumption between DQN and DDPG controllers (testing phase). The DQN controller consumes 74% more energy than the DDPG controller.

the software engineer can determine the setpoint range of each control variable without too much effort.

3. Because the operational range of control variables are determined by HVAC engineers during the design phase, simply adopting them as variable boundaries in DDPG helps prevent safety issues. In this simulation, the minimum outdoor air damper opening is 20%, so the damper cannot be fully closed during the occupied time. Also, the minimum fan speed is 10%, so the RL controller cannot mistakenly switch off the fan when the chiller is running. Note that, ensuring DDPG adhering to these safety constraints will increase its acceptance among HVAC technicians, who prioritize maintaining system integrity and preventing potential malfunctions.

Compared to DQN, DDPG is more suitable for control problems with a high-dimensional action space. Especially, the control of HVAC systems belongs to this category considering its complexity and large degrees of freedom. The designed RL controller usually needs to control multiple devices on the actuator level such as dampers, fans, valves, chillers, and pumps, and also determine setpoints on the supervisory level, such as the chilled water setpoint, supply air setpoint, and zone air setpoint. DDPG could be more competent than DQN given the large numbers of control variables.

5. Conclusion

This paper designed two reinforcement learning (RL) algorithms for central HVAC system management in an office building: a value-based algorithm termed DQN and an actor–critic algorithm termed DDPG. The performance of the DQN controller was evaluated through experiments in a building test facility, and a detailed comparison between DQN and DDPG was conducted on a simulation platform with a hybrid simulator. The results demonstrated that both DQN and DDPG can provide appropriate cooling to an office building, ensuring thermal comfort and minimizing energy consumption. However, the value-based algorithm also showed some limitations. Particularly, it applies to only small discrete action spaces, which restricts its ability to find optimal actions involving multiple continuous control variables in an HVAC system. In contrast, DDPG proved to be more efficient for larger action spaces, requiring fewer computational resources, converging faster, and delivering better temperature and humidity control along with lower energy consumption. By directly handling continuous actions through an actor–critic architecture and optimizing the policy using deterministic policy gradients, DDPG can successfully address the limitations encountered by DQN in the control of HVAC systems.

Future work can focus on the experimental assessment of actor–critic RL algorithms on the control of HVAC systems. Currently, most

applications of RL on building energy systems are simulation studies, and we hope to test its long-term performance in real buildings. This can help us identify practical issues of RL applications and refine the whole RL controller for better control performance.

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.

Acknowledgments

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20212020800120).

Data availability

The authors do not have permission to share data.

References

[1] Energy Information Administration, Commercial Buildings Energy Consumption Survey, US Energy Information Administration USA, 2018.
[2] D. Kim, J.E. Braun, Development, implementation and performance of a model predictive controller for packaged air conditioners in small and medium-sized commercial building applications, Energy Build. 178 (2018) 49–60.
[3] S. woo Ham, D. Kim, T. Barham, K. Ramseyer, The first field application of a low-cost MPC for grid-interactive K-12 schools: Lessons-learned and savings assessment, Energy Build. 296 (2023) 113351.
[4] E. Zanetti, D. Kim, D. Blum, R. Scoccia, M. Aprile, Performance comparison of quadratic, nonlinear, and mixed integer nonlinear MPC formulations and solvers on an air source heat pump hydronic floor heating system, J. Build. Perform. Simul. 16 (2) (2023) 144–162, http://dx.doi.org/10.1080/19401493.2022.2120631, URL https://www.tandfonline.com/doi/full/10.1080/19401493.2022.2120631.
[5] G.D. Kontes, G.I. Giannakis, V. Sánchez, P. de Agustin-Camacho, A. Romero-Amorrortu, N. Panagiotidou, D.V. Rovas, S. Steiger, C. Mutschler, G. Gruen, Simulation-based evaluation and optimization of control strategies in buildings, Energies 11 (12) (2018) 3376.
[6] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[7] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.
[9] K.U. Ahn, C.S. Park, Application of deep Q-networks for model-free optimal control balancing between different HVAC systems, Sci. Technol. Built Environ. 26 (1) (2020) 61–74, http://dx.doi.org/10.1080/23744731.2019.1680234, Publisher: Taylor & Francis.
[10] M. Biemann, F. Scheller, X. Liu, L. Huang, Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control, Appl. Energy 298 (2021) 117164, http://dx.doi.org/10.1016/j.apenergy.2021.117164, URL https://www.sciencedirect.com/science/article/pii/S0306261921005961.
[11] X. Deng, Y. Zhang, Y. Zhang, H. Qi, Towards optimal HVAC control in non-stationary building environments combining active change detection and deep reinforcement learning, Build. Environ. 211 (2022) 108680, http://dx.doi.org/10.1016/j.buildenv.2021.108680, URL https://linkinghub.elsevier.com/retrieve/pii/S0360132321010702.
[12] S. Dey, T. Marzullo, X. Zhang, G. Henze, Reinforcement learning building control approach harnessing imitation learning, Energy AI 14 (2023) 100255.
[13] Y. Du, H. Zandi, O. Kotevska, K. Kurte, J. Munk, K. Amasyali, E. Mckee, F. Li, Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning, Appl. Energy 281 (2021) 116117, http://dx.doi.org/10.1016/j.apenergy.2020.116117, URL https://www.sciencedirect.com/science/article/pii/S030626192031535X.
[14] Y. Fan, Q. Fu, J. Chen, Y. Wang, Y. Lu, K. Liu, A deep reinforcement learning control method for multi-zone precooling in commercial buildings, Appl. Therm. Eng. 260 (2025) 124987.
[15] Q. Fu, Z. Li, Z. Ding, J. Chen, J. Luo, Y. Wang, Y. Lu, ED-DQN: An event-driven deep reinforcement learning control method for multi-zone residential buildings, Build. Environ. 242 (2023) 110546.
[16] G. Gao, J. Li, Y. Wen, DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning, IEEE Internet Things J. 7 (9) (2020) 8472–8484, http://dx.doi.org/10.1109/JIOT.2020.2992117.
[17] Y. Gao, S. Shi, S. Miyata, Y. Akashi, Successful application of predictive information in deep reinforcement learning control: A case study based on an office building HVAC system, Energy 291 (2024) 130344.
[18] C. Gao, D. Wang, Comparative study of model-based and model-free reinforcement learning control performance in HVAC systems, J. Build. Eng. 74 (2023) 106852.
[19] D. Wang, W. Zheng, Z. Wang, Y. Wang, X. Pang, W. Wang, Comparison of reinforcement learning and model predictive control for building energy system optimization, Appl. Therm. Eng. 228 (2023) 120430.
[20] F. Guo, S. woo Ham, D. Kim, H.J. Moon, Deep reinforcement learning control for co-optimizing energy consumption, thermal comfort, and indoor air quality in an office building, Appl. Energy 377 (2025) 124467.
[21] X. Liu, Y. Wu, H. Wu, Enhancing HVAC energy management through multi-zone occupant-centric approach: A multi-agent deep reinforcement learning solution, Energy Build. 303 (2024) 113770.
[22] H. Qin, Z. Yu, T. Li, X. Liu, L. Li, Energy-efficient heating control for nearly zero energy residential buildings with deep reinforcement learning, Energy 264 (2023) 126209.
[23] M. Shin, S. Kim, Y. Kim, A. Song, Y. Kim, H.Y. Kim, Development of an HVAC system control method using weather forecasting data with deep reinforcement learning algorithms, Build. Environ. 248 (2024) 111069.
[24] F.M. Solinas, A. Macii, E. Patti, L. Bottaccioli, An online reinforcement learning approach for HVAC control, Expert Syst. Appl. 238 (2024) 121749.
[25] S. Touzani, A.K. Prakash, Z. Wang, S. Agarwal, M. Pritoni, M. Kiran, R. Brown, J. Granderson, Controlling distributed energy resources via deep reinforcement learning for load flexibility and energy efficiency, Appl. Energy 304 (2021) 117733, http://dx.doi.org/10.1016/j.apenergy.2021.117733, URL https://linkinghub.elsevier.com/retrieve/pii/S0306261921010801.
[26] S. Xia, P. Wei, Y. Liu, A. Sonta, X. Jiang, RECA: A multi-task deep reinforcement learning-based recommender system for co-optimizing energy, comfort and air quality in commercial buildings, in: Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, ACM, Istanbul Turkey, 2023, pp. 99–109, http://dx.doi.org/10.1145/3600100.3623735, URL https://dl.acm.org/doi/10.1145/3600100.3623735.
[27] H. Wang, X. Chen, N. Vital, E. Duffy, A. Razi, Energy optimization for HVAC systems in multi-VAV open offices: A deep reinforcement learning approach, Appl. Energy 356 (2024) 122354.
[28] T. Yang, L. Zhao, W. Li, J. Wu, A.Y. Zomaya, Towards healthy and cost-effective indoor environment management in smart homes: A deep reinforcement learning approach, Appl. Energy 300 (2021) 117335.
[29] Y.R. Yoon, H.J. Moon, Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling, Energy Build. 203 (2019) 109420, http://dx.doi.org/10.1016/j.enbuild.2019.109420, URL https://www.sciencedirect.com/science/article/pii/S0378778819310692.
[30] D. Zhuang, V.J. Gan, Z.D. Tekler, A. Chong, S. Tian, X. Shi, Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning, Appl. Energy 338 (2023) 120936.
[31] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, 2019, arXiv:1509.02971 [cs, stat].
[32] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Harley, T.P. Lillicrap, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning.
[33] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Appl. Energy 269 (2020) 115036, http://dx.doi.org/10.1016/j.apenergy.2020.115036, URL https://linkinghub.elsevier.com/retrieve/pii/S0306261920305481.
[34] S. Sierla, H. Ihasalo, V. Vyatkin, A review of reinforcement learning applications to control of heating, ventilation and air conditioning systems, Energies 15 (10) (2022) 3526, http://dx.doi.org/10.3390/en15103526, Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, URL https://www.mdpi.com/1996-1073/15/10/3526.
[35] FLEXLAB: The World's Most Advanced Integrated Building and Grid Technologies Testbed, Lawrence Berkeley National Laboratory, 2021, URL https://flexlab.lbl.gov/.
[36] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg, M. Yazdanian, J. Huang, D. Crawley, U.S. Department of Energy Commercial Reference Building Models of the National Building Stock, Tech. Rep. NREL/TP-5500-46861, 1009264, 2011, http://dx.doi.org/10.2172/1009264, URL http://www.osti.gov/servlets/purl/1009264-pitlfN/.
[37] A.A. Standard, et al., Standard guide for using indoor carbon dioxide concentrations to evaluate indoor air quality and ventilation, in: ASTM Standard D6245-12, Standard Guide for using Indoor Carbon Dioxide Concentrations To Evaluate Indoor Air Quality and Ventilation, ASTM International West Conshohocken, PA, 2012.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, 2013, arXiv preprint arXiv:1312.5602.

[39] M. Wetter, W. Zuo, T.S. Nouidui, X. Pang, Modelica Buildings library, J. Build. Perform. Simul. 7 (4) (2014) 253–270, http://dx.doi.org/10.1080/19401493.2013.765506.

[40] T. Nouidui, M. Wetter, W. Zuo, Functional mock-up unit for co-simulation import in EnergyPlus, J. Build. Perform. Simul. 7 (3) (2014) 192–202, Publisher: Taylor & Francis.

[41] C. Andersson, J. Åkesson, C. Führer, Pyfmi: A Python Package for Simulation of Coupled Dynamic Models with the Functional Mock-Up Interface, Centre for Mathematical Sciences, Lund University Lund, Sweden, 2016.

[42] American Society of Heating Refrigerating and Air-Conditioning Engineers, 2001 Ashrae Handbook : Fundamentals (SI), ASHRAE, 2001.