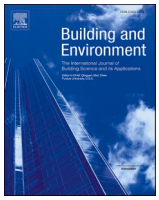
Building and Environment 244 (2023) 110816

Contents lists available at ScienceDirect

Building and Environment

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



MF^2: Model-free reinforcement learning for modeling-free building HVAC control with data-driven environment construction in a residential building

Man Wang [a,b], Borong Lin [a,b,*]

[a] Department of Building Science, Tsinghua University, Beijing, 100084, China
[b] Key Laboratory of Eco Planning & Green Building, Ministry of Education, Tsinghua University, China

ARTICLE INFO

Keywords:
HVAC
Energy efficiency
Reinforcement learning
Environment
DQN
DDPG

ABSTRACT

Reinforcement Learning (RL) has advanced energy-efficient control of building Heating, Ventilation and Air Conditioning (HVAC) systems. Constructing a suitable RL environment for buildings is a crucial challenge. Compared to widely-used simulation-based environments, data-driven approaches offer higher training efficiency but face convergence difficulties due to influential factors, limiting their current application.

To explore data-driven construction of RL environments for building HVAC systems, this study proposes two strategies for controlling room temperature setpoints in a residential building. XGBoost and Long Short-Term Memory Network (LSTM) are trained for energy consumption and room temperature prediction. One strategy predicts parameters for on-off states, while the other for power-on states. The XGBoost models are integrated into an OpenAI Gym environment. The first strategy achieves 0.8634 R^2 and 0.2423 Root Mean Squared Error (RMSE) for energy consumption prediction. The R^2 of room air temperature models are approximately 0.99 and the RMSE are lower than 0.31. The second strategy achieves 0.9181 R^2 and 0.1042 RMSE for energy consumption prediction and similar performance for room temperature prediction. Deep Q-learning (DQN) and Deep Deterministic Policy Gradient (DDPG) algorithms are separately trained using these environments. Results show that the first strategy fails to induce the correct training of RL models, while the second strategy successfully induces a useable DDPG model for controlling building HVAC systems but fails to induce a useable DQN model. We analyze the reasons behind these observations. Compared to the original room temperature setpoint method, the DDPG-based HVAC control logic achieves a 10.06% energy-saving effect while ensuring comfort.

1. Introduction

HVAC systems have inherent problems [1]. Firstly, building air conditioning systems are nonlinear. Their equipment performance curves and thermodynamic properties of working mediums all exhibit nonlinear characteristics. Additionally, building HVAC systems are dynamic. On one hand, indoor and outdoor disturbances such as weather and personnel are constantly changing, and on the other hand, temperature, frequency and other settings of the air conditioning system may change momentarily. These characteristics above impede energy efficiency control of building HVAC systems.

However, the operation of building HVAC systems is highly dependent on manual labor, which cannot capture the dynamic changes in weather and personnel. And it is also a waste of data and sensors for building automation systems (BAS) [1]. Manual operation can no longer meet the requirements for energy-saving operation of buildings, and it is very urgent to explore more efficient ways to control building HVAC systems.

The control strategies for building HVAC systems can be roughly divided into three stages. The first stage is evolutionary algorithm control [2–7], during which bio-inspired evolutionary algorithms, with genetic algorithms as the representative, are mainly used to search for the optimal solution of proportional, integral and derivative (PID) control and improve the performance of PID control. In Ref. [8], researchers trained a model based on the genetic algorithm to tune PID controllers in HVAC systems automatically. The results indicated that the proposed model was useful for this task. In Ref. [9], Lu et al. established mathematical models associated with cooling loads and energy consumption for heat exchangers and energy-consuming devices. Then they solved the formula of mix-integer nonlinear constraint optimization of system energy by modified genetic algorithms. In Ref. [10], researchers used a multi-objective genetic algorithm to permit the

* Corresponding author. Department of Building Science, Tsinghua University, Beijing, 100084, China.
E-mail address: linbr@tsinghua.edu.cn (B. Lin).

https://doi.org/10.1016/j.buildenv.2023.110816
Received 19 May 2023; Received in revised form 18 August 2023; Accepted 6 September 2023
Available online 7 September 2023

optimal operation of the building's HVAC systems. And they validated the proposed optimization process on an existing VAV system in two summers. The main issue with this approach is the low efficiency of the algorithm used, making it unsuitable for controlling complex air conditioning systems with a large number of parameters.

The second stage is Model Predictive Control (MPC) [11–15], where this approach involves establishing a combined thermodynamic model for the building and HVAC equipment, then determining the optimal control parameters of the HVAC system through computational methods. In Ref. [16], Yudong et al. developed a deep learning (DL) based MPC model to realize real-time control of building thermal environment. They achieved 4%–7% energy saving on average through DL-based MPC compared with adaptive and conventional PID control. PengFei et al. [17]proposed a practically effective and computationally efficient MPC algorithm to optimize building energy usage while maintaining thermal comfort in a multi-zone medium-sized commercial building, which achieved 17.5% energy saving. Bing et al. [18] illustrated a methodology to control the HVAC system of the building based on the prediction of occupant behavior patterns and local weather conditions. Then a Nonlinear Model Predictive Control (NMPC) was designed and reduced 17.8% energy consumption in the experiment. The main problem with this method is that the thermodynamic model, which is optimized for the solution, heavily relies on the experience of researchers. The model may be over-simplified, leading to a poor match between the model and the actual scenario.

The third stage is Machine Learning (ML) based control [19,20]. This method mainly determines control parameters by learning the hidden expert experience in the data through ML algorithms. InRef [21], researchers established a rule-based HVAC control system using a multi-layer perceptron network. The system outperformed other alternatives when the deep and medium bounds are utilized. In Ref. [22], transfer learning is utilized to overcome the challenge that deep RL used for HVAC system control took too much time training. Zhe et al. [23] applied Long Short-Term Memory Networks (LSTM) o predict miscellaneous electric loads, lighting loads, occupant counts and internal heat gains in two office buildings. Prediction errors of internal heat gains are reduced from 12% to 8% in Building A and 26%–16% in Building B. Currently, the main challenges are defficult feature engineering, inappropriate model application, and insufficient integration with expert knowledge.

In recent years, RL [24–34] has received significant attention in ML control algorithms. This is because, compared to deep learning, RL has lower dependence on historical data and theoretically enables model-free control. In Ref. [35], researchers meant to establish a multi-agent deep RL method for the building Cooling Water System Control (MA-CWSC) to optimize the load distribution, cooling tower fan frequency, and cooling water pump frequency of different types of chillers. The MA-CWSC method achieved an 11.1% improvement compared with rule-based control. Xi et al. [36] developed Transfer Learning and Deep Reinforcement Learning (TL-DRL) integrated framework. By fine-tuning the last few layers of the target Deep Q-learning (DQN) in the target building, they improved the training efficiency by about 13.28%. Yue et al. [37]presented a Deep Reinforcement Learning (DRL) based multivariate occupant-centric control framework. By considering personalized thermal comfort and occupant presence, they achieved 14% cooling energy saving with 11% thermal acceptable improvement.

In air conditioning control based on RL, an important issue is how to construct the environment for training the RL agent. Currently, the most common approach is to build the environment using building simulation software such as EnergyPlus and Modelica. In Ref. [38], Takao et al. proposed an RL testbed for power consumption optimization based on Energyplus. They develop a data center simulation model as an RL environment and achieved 22% improvement compared to a model-based control algorithm built in the EnergyPlus. In Ref. [39], researchers estimated the control algorithm based on DQN on the Energyplus-BCVTB testbed. The proposed agent could reduce CO_2 below 800 ppm all the time with superior PMV and 4–5% energy saving. In Ref. [40], an RL environment based on Modelica was established and the A2C algorithm was tested on this environment.

The aforementioned method of constructing the environment based on simulation software allows the agent to obtain sufficient information, as the physical environment of the building is completely known to the agent. Therefore, convergence is relatively easy. However, there is often a performance gap between the building energy simulation model and the real building, leading to significant differences between the simulated environment and the real environment. Consequently, when the agent is deployed in the real environment, it may not perform well. Additionally, the interaction between building energy simulation software and Python, for example, may remain unchanged, resulting in long training times for the agent.

To overcome such problems for RL environment construction by building simulation software, researchers began to explore a method to build RL environment with data-driven models. Christian et al. [41] pre-trained an offline RL agent in a black box model environment based on a LSTM model with an average error of 0.3246. The model could be deployed with a 19.4% cost reduction compared to traditional controllers. Mengjie et al. [42] proposed a model-free RL control method to optimize window opening behaviors of an office building. The Root Mean Squared Error (RMSE) for the experimental room temperature prediction LSTM model was 0.2 °C. The indoor air quality of the office improved by 90% with the RL controller trained in this LSTM environment compared to historical data. Shunian et al. [43] proposed a model-free optimal control method based on RL. The COP and cooling tower outlet water temperatures are simulated by random forests. The COP model achieved 1.75% MAPE and 2.56% CV(RMSE) in the test dataset. The MAPE and CV(RMSE) of the cooling tower outlet water temperature prediction model are 0.67% and 1.01%, respectively. These models were integrated with other thermodynamic models to build an RL environment. Finally, the Q-learning model trained in this environment could save 11% energy consumption compared to the basic controller. Overall, building an RL environment based on data-driven models is closer to the real environment, and the interaction between the environment and RL is more convenient and efficient. However, this type of environment is non-white-box, and there may be potential influencing factors within the historical data, which increases the difficulty of convergence during the training process of the RL agent.

To explore the applicability of data-driven methods in building an RL environment, this study conduct the following research on a residential building. In this study, data from the heating season of the building during 2020–2021 are collected, and based on this historical data, energy consumption prediction models for the HVAC system and temperature prediction models for three target rooms are trained using two different strategies based on the XGBoost model and LSTM model. In Strategy 1, the energy consumption and temperature prediction models are used to predict the HVAC system's operation status, both for turning it on and off. In Strategy 2, the energy consumption and temperature prediction models are only used to predict during the HVAC system's operation period. Rule-based control is employed to validate the RL environments under the two strategies mentioned above. Based on the energy consumption and temperature prediction models, RL agents are trained using a DQN-based model for discrete output room setpoint temperature and a Deep Deterministic Policy Gradient(DDPG)-based model for continuous output room setpoint temperature. The energy-saving effect and indoor thermal environment of the trained models are compared with rule-based control results. None of LSTM models are converged. So the XGBoost models are used further. The energy consumption prediction model performance for Strategy 1 is 0.8634 of R^2 score and 0.2324 of RMSE, and the temperature prediction model performance is around 0.99 of R^2 score and lower than 0.31 of RMSE. For Strategy 2, the energy consumption prediction model performance is 0.9181 of R^2 score and 0.1042 of RMSE, and the temperature prediction

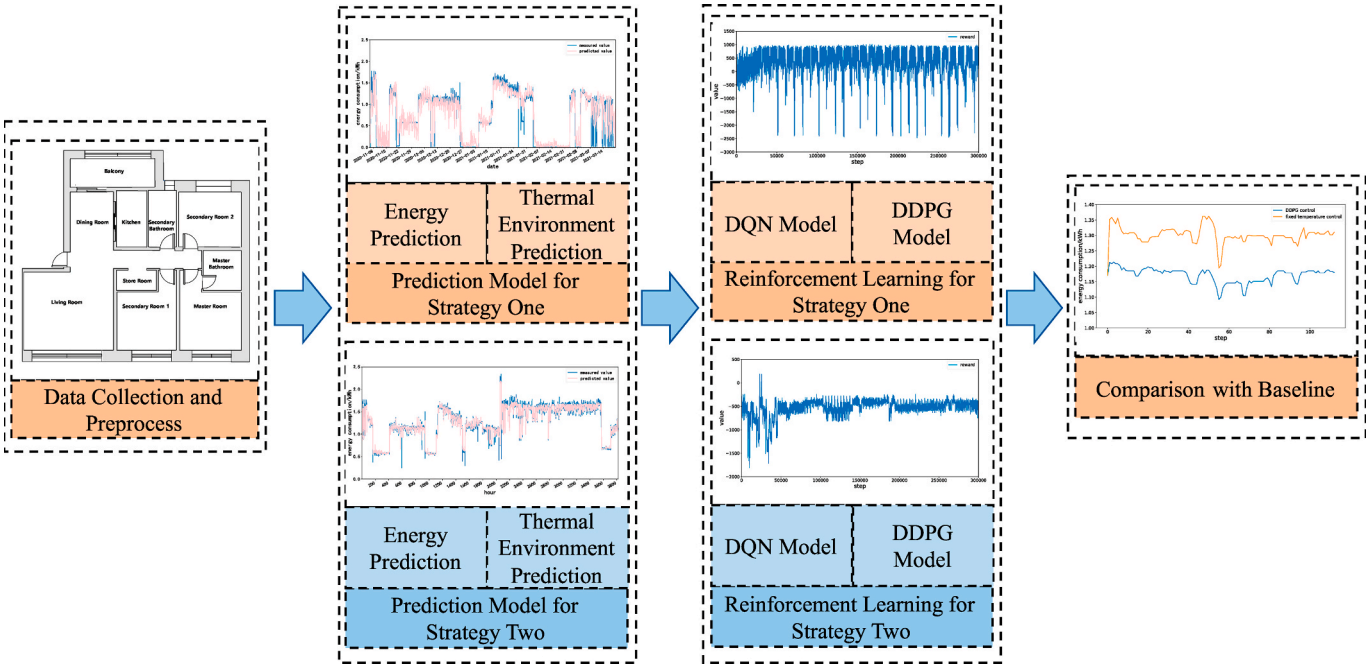


Fig. 1. Method in this essay.

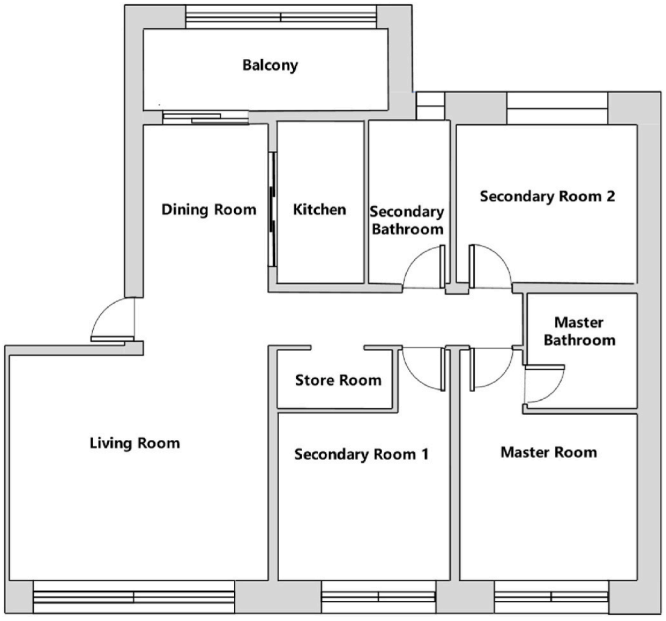


Fig. 2. House type.

Table 1
Thermal performance of envelop enclosure.

Envelop enclosure	Material	Thickness (mm)	heat transfer coefficient (W/m^2·K)
Roof	XPS	400	0.06
External wall	GEPS	250	0.17
Floor	GEPS	400	0.08

model performance is also around 0.99 of R^2 score and lower than 0.32 of RMSE. However, the DQN model does not converge in this study's RL training. Under the continuous output control temperature from DDPG, it achieves a 10.06% energy saving compared to rule-based control while ensuring thermal comfort.

This study validate the feasibility of using data-driven methods to build an RL control environment and provided some references on how to construct such an environment. Furthermore, a comparison is made between the training of RL agents using discrete output control temperature and continuous output control temperature. The DDPG algorithm achieves a 10.06% energy-saving rate while maintaining indoor thermal comfort.

2. Methodology

The technical roadmap of this paper is as shown in Fig. 1.

2.1. Data collection and preprocessing

The target building of this study is a near-zero energy residential house located in Beijing. As an experimental building, it is unoccupied for long periods, resulting in minimal disturbances from human activities in its historical data. The floor plan of this house is shown in Fig. 2. The residential building consists of a living room, a dining room, a master bedroom, two secondary bedrooms, a kitchen, two bathrooms, and a balcony. The living room and dining room are connected. In our research, we focus on the necessary air-conditioned zones, so our study includes the living room, dining room, master bedroom, and two secondary bedrooms, treating the living room and dining room as one space.

As a near-zero energy building, this residential house has excellent thermal insulation performance. The insulation materials, thickness, and heat transfer coefficients of the external walls are shown in Table 1. The inclusion of ground insulation is due to the building's location on the second floor. In order to reduce heat transfer with the first floor, insulation measures have also been applied to the floor.

The air conditioning system of the building is equipped with an air-source heat pump fresh air unit. The fresh air can be turned off, and in the historical operational state, it is known that the fresh air was not enabled. The heat pump has heat recovery capability with an efficiency of up to 70%. Each air-conditioned room is equipped with an end unit connected to the heat pump, but there is only one temperature setpoint for all rooms, meaning that individual temperature control in each room

Table 2
Collected data.

Indoor thermal temperature	Wall surface temperature	Heat pump operation parameters
Master bedroom Secondary bedroom 1 Secondary bedroom 2	Internal surface temperature Temperature between insulation layers	Fresh air temperature Supply air temperature Return air temperature Exhaust air temperature

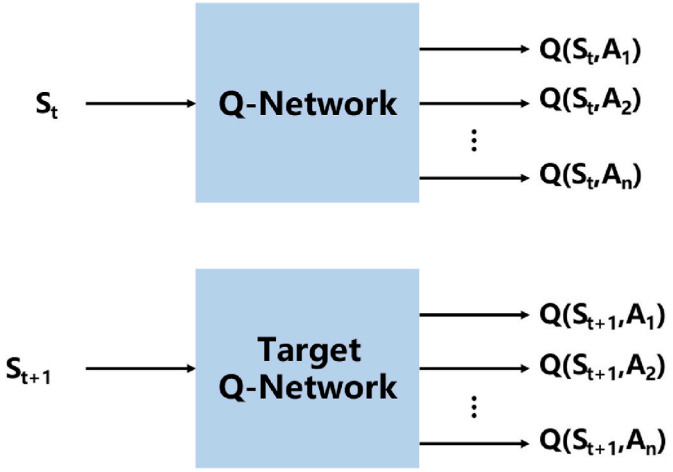


Fig. 3. DQN network structure.

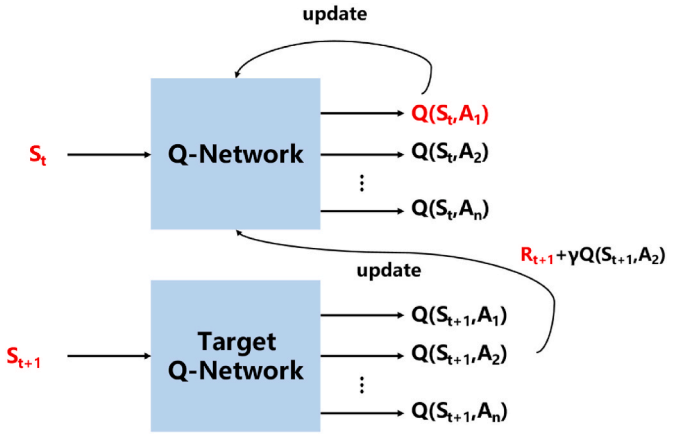


Fig. 4. DQN network update.

is not possible. The dining room and kitchen have return air outlets in the ceiling, while the other rooms only have supply air outlets. The supply air speed can be set to high, medium, or low, but to minimize noise, the historical data indicates that the supply air speed was fixed at a low speed. The system can be remotely controlled by writing to a database, and its historical operating data is also uploaded to the database.

We extract historical data for the heating seasons of 2020 and 2021 from the aforementioned database. The data are collected at hourly intervals. The energy consumption is measured in kWh, and the room temperature is in °C. The data collection period for 2020 is from November 6, 2020, to March 20, 2021. Due to data recording damage during the late period of the 2021 heating season, the data collection period for that year is from November 8, 2021, to February 24, 2022. Table 2 presents the parameters related to the indoor environment in the historical data. After organizing the data, it is discovered that the temperature of the living room is not recorded due to a sensor failure. Therefore, in subsequent research, the thermal comfort of the indoor environment is evaluated based on the temperatures of the three bedrooms, and no further processing of the temperatures in the living room and dining room is performed. Since the temperature setpoints for each room are not recorded in the heat pump parameters, and household heat pumps are controlled based on room temperature, we choose to use the return air temperature as a substitute for the room temperature setpoint during training. This is because the return air temperature can be considered an actual value of the room temperature and can largely reflect the temperature setting in that room. Based on statistics, out of 114 wall sensors, 9 sensors are installed on the walls of the three bedrooms. When training the energy consumption prediction model, based on prior knowledge, the indoor temperatures of the three rooms, the actual values of the fresh air temperature, supply air temperature, exhaust air temperature, and room temperature setpoint (return air temperature) are selected as parameters for training. Similarly, when training the room temperature prediction model, the available wall temperatures are also selected as parameters based on prior knowledge for modeling the room temperature.

2.2. Methodology for building an environment for RL HVAC control

2.2.1. Principles of ML algorithms

Next, we will introduce the principles of XGBoost and RL. XGBoost is a boosting algorithm proposed by Tianqi Chen [44] in 2016. It is an ensemble learning method that combines the power of gradient boosting with advanced regularization techniques, making it highly effective in solving a wide range of supervised learning problems.

RL is an algorithmic approach that mimics the learning process of humans by continuously interacting with an environment through an agent. In RL, a training environment is provided first, which is responsible for interacting with the agent being trained. At time t, the agent receives specific environmental parameters, referred to as state S_t, from

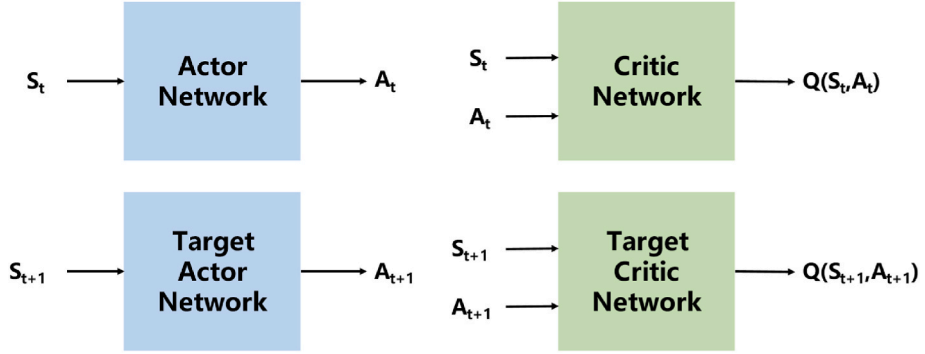


Fig. 5. DDPG network structure.

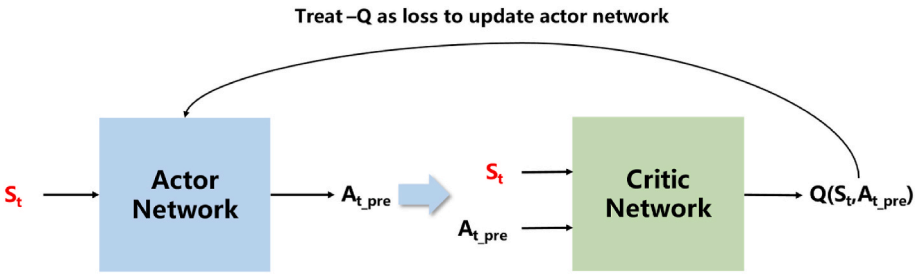


Fig. 6. DDPG actor network update.

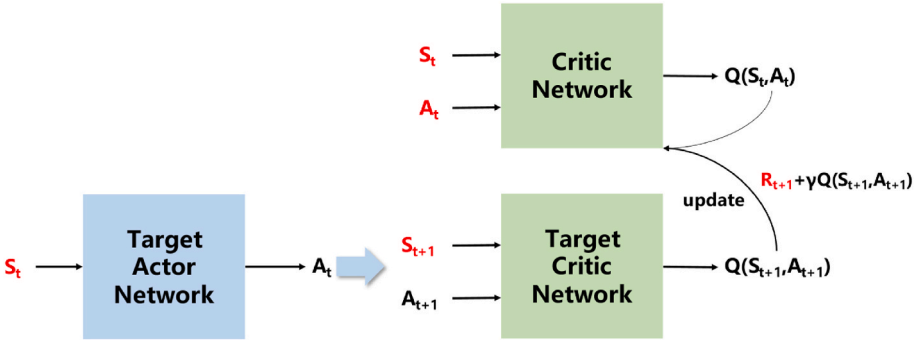


Fig. 7. DDPG critic network update.

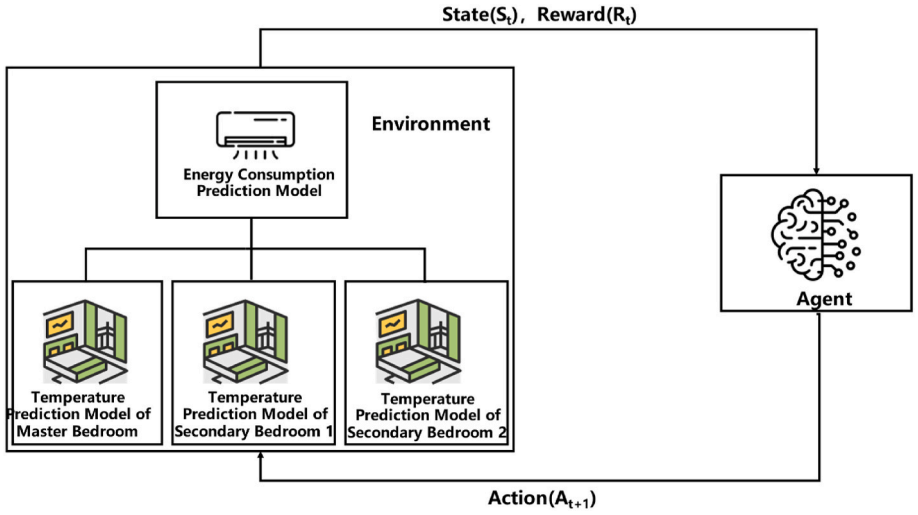


Fig. 8. Interaction process of RL with the environment.

Table 3
Ranges of parameter variations for LSTM models.

	Minimum value	Maximum value
Hidden size	16	256
Hidden layer number	2	5
Learning rate	0.00001	0.01
Time step	4	24

the environment. Based on the current environmental situation, the agent takes action A_t, which affects the environment. The environment then provides a score, known as reward R_t, for that action. Through iterative processes, the agent learns which actions yield the highest rewards, thereby acquiring the ability to make decisions based on the environment.

Table 4
Energy prediction and indoor temperature prediction performance of XGBoost models for Strategy 1.

Prediction model	Energy	Master bedroom temperature	Temperature of secondary bedroom 1	Temperature of secondary bedroom 2
R^2	0.9537	0.9905	0.9920	0.9975
RMSE	0.1454	0.2784	0.2698	0.1532

In this study, we primarily apply two RL algorithms: DQN [45] and DDPG [46]. The following sections will provide an introduction to these two algorithms.

The network structure of DQN is illustrated in Fig. 3. DQN consists of two networks: the Q-network and the target Q-network. Given a state S_t,

Table 5
Energy prediction and indoor temperature prediction performance of LSTM models for Strategy 1.

Prediction model	Energy	Master bedroom temperature	Temperature of secondary bedroom 1	Temperature of secondary bedroom 2
R^2	−0.0601	−0.0111	−0.0091	−0.0228
RMSE	0.6620	2.9742	2.9436	3.0565

the Q-network can output the Q-values for various actions, enabling the selection of the current action A_t based on a greedy policy. By applying A_t to the environment, the subsequent state $S_{(t+1)}$ and reward $R_{(t+1)}$ at time (t+1) can be obtained, thus forming a complete experience tuple (S_t, A_t, $S_{(t+1)}$, $R_{(t+1)}$). These experiences are then stored in an experience replay buffer.

In the subsequent training process, the update procedure is illustrated in Fig. 4. A batch of experiences is sampled from the experience replay buffer, where the red items represent the known values. Assuming $A_t = A_1$ and $A_{t+1} = A_2$, the current action-value $Q(S_t, A_1)$ can be determined based on the current state S_t and action A_t. Then, the next state S_{t+1} is inputted into the target Q-network to obtain the Q-values for various actions, and the action A_2 corresponding to the maximum Q-value is selected. Using $Q(S_t, A_1)$ as the predicted value and $R_{(t+1)} + \gamma Q(S_{t+1}, A_2)$ as the target value, the network is updated through the backpropagation of errors. The parameters of the Q-network are periodically copied to the target Q-network for its update.

The network structure of DDPG is depicted in Fig. 5. It consists of two actor networks whose objective is to output actions A_t that maximize the action-value $Q(S_t, A_t)$ based on the state S_t. The better the actor network performs, the larger the $Q(S_t, A_t)$ value. Additionally, there are two critic networks whose objective is to output the corresponding action-value $Q(S_t, A_t)$ given the state and action (S_t, A_t). The more accurate the $Q(S_t, A_t)$ value, the better the training results of the critic networks. The difference between the target actor/critic networks and the original actor/critic networks is similar to DQN, where the target networks undergo delayed updates. However, in DDPG, the network updates involve copying a certain percentage of the original network's parameters to the target network. This ensures smaller update magnitudes and maintains network stability.

When generating experiences, given a state St, the actor network produces an action At' which is then perturbed by adding noise N to obtain the action $A_t = A_t' + N$. The addition of noise is to ensure a certain level of exploration. Subsequently, the action At is applied to the environment, resulting in the next state S_{t+1} and reward R_{t+1}, thus forming an experience tuple (S_t, A_t, S_{t+1}, R_{t+1}).

During training, a batch of experiences is sampled from the experience replay buffer. Taking (S_t, A_t, S_{t+1}, R_{t+1}) as an example, as shown in Fig. 6, the red items represent the known values. When training the actor network, the state S_t is fed into the actor network, yielding a predicted action A_{t_pre}. This (S_t, A_{t_pre}) pair is then inputted into the critic network,

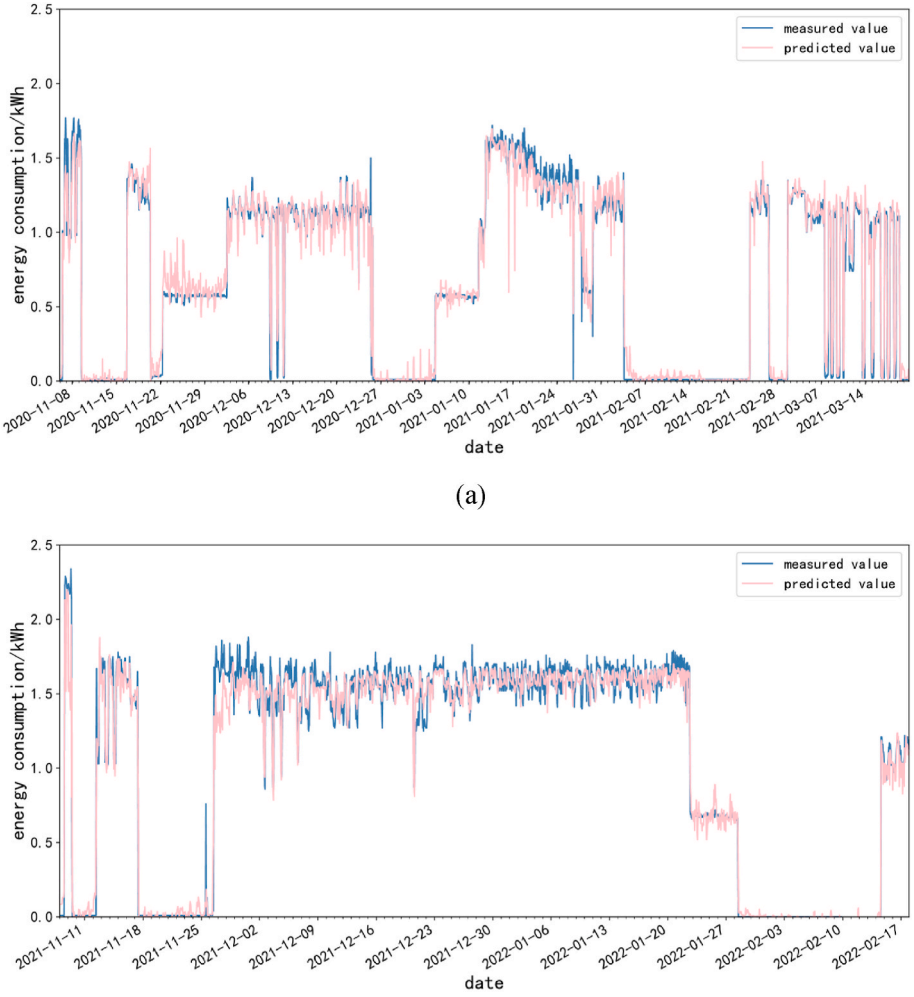
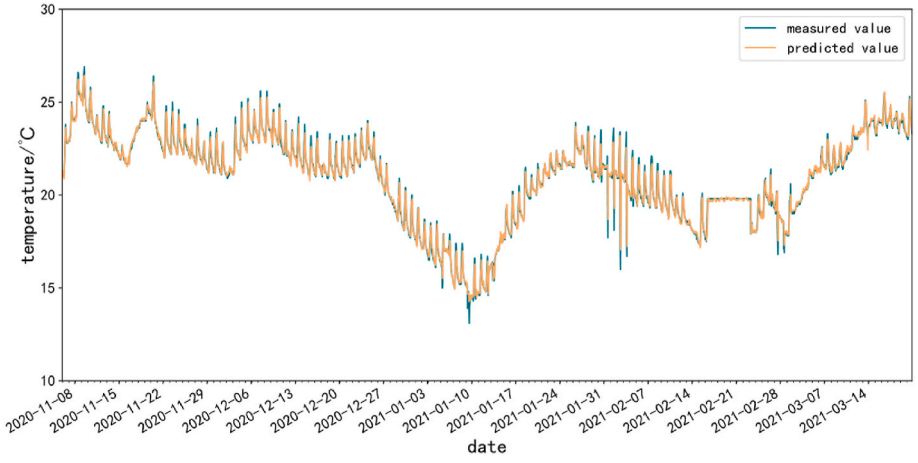
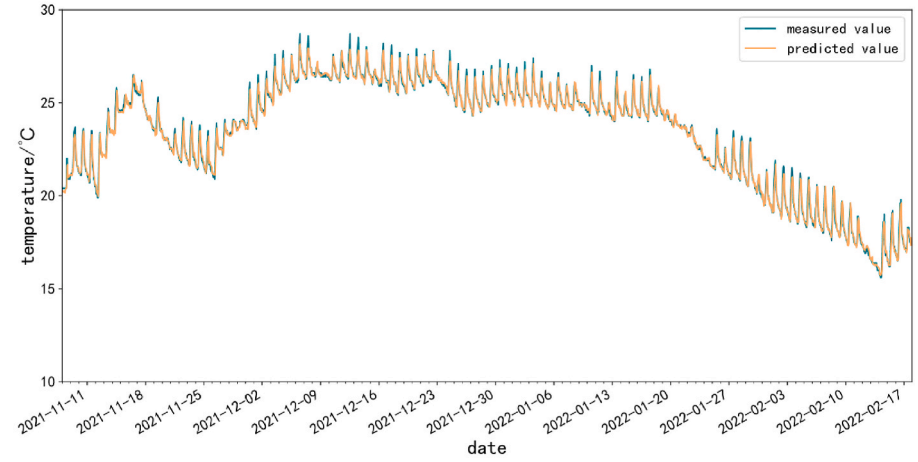


Fig. 9. Energy prediction of Strategy 1.



(a)



(b)

Fig. 10. Temperature prediction of master bedroom of Strategy 1.

which outputs an action-value $Q(S_t, A_{t\_pre})$. Since the objective of the actor network is to maximize $Q(S_t, A_{t\_pre})$. The negative of $Q(S_t, A_{t\_pre})$, $-Q(S_t, A_{t\_pre})$, is used as the loss for backpropagation to train the actor network.

For training the critic network, as shown in Fig. 7, the red items represent the known values. Using $Q(S_t, A_t)$ as the predicted value and $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$ as the target value, the difference between the predicted value and the target value is used as the loss. The loss is then backpropagated through the network to train the critic network.

These two algorithms have different requirements for the action space due to their underlying principles. DQN requires a discrete action space, while DDPG requires a continuous action space.

2.2.2. Two RL environment construction strategies

The interaction process of RL with the environment in this text is shown in Fig. 8. In this environment, it is necessary to predict energy consumption and indoor temperature. First, it is necessary to predict the indoor temperature of three bedrooms based on current indoor temperatures, the temperature set value, and other parameters. After obtaining the predicted temperatures of the three rooms, the energy consumption of the HVAC is predicted based on other operating parameters. In this study, our goal is to reduce the energy consumption of air conditioning while ensuring room thermal comfort, which will be reflected in the reward function. Once the environment is set up, the agent will start interacting with the environment. The agent will receive the current environmental state and the reward value of the previous action, make a decision on the action value for the next moment, and continue to provide it to the environment. This process will be repeated continuously during the agent's training process to train an agent capable of achieving control objectives.

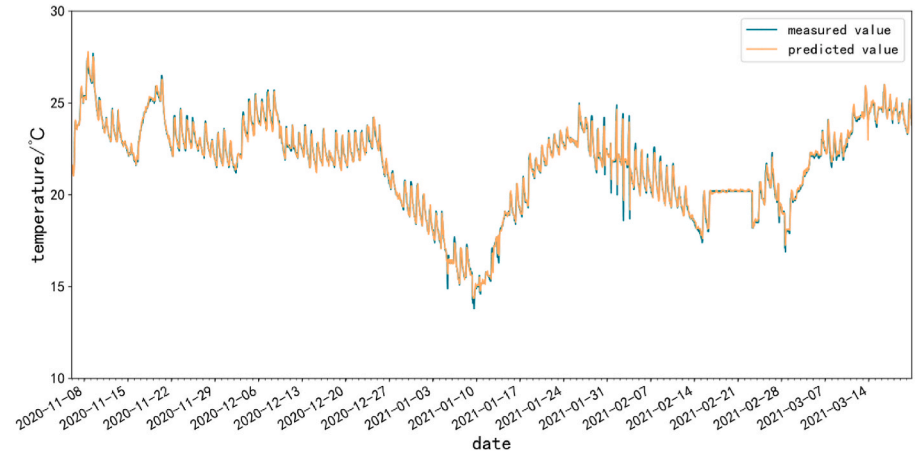
This paper proposes two data-driven strategies for constructing HVAC RL environments in buildings. The following will introduce the basic concepts of RL, states (S_t), actions (A_t), and rewards (R_t).

Strategy 1: Strategy 1 is based on an ideal completely uncontrolled HVAC system in buildings. The basic idea is to use a ML prediction model to simultaneously predict the environmental characteristics of the air conditioning system under both on and off states.

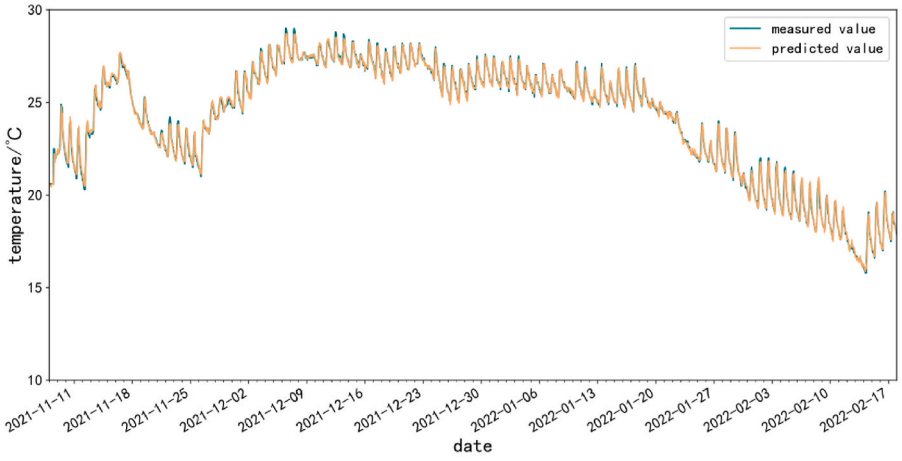
State: Assuming the current time is t, we want to obtain the power consumption and indoor environmental parameters of the air conditioning system at this time. The power consumption at this time, E_t, represents the electricity consumption within 1 h. The indoor environmental parameters, after preprocessing, mainly include the temperatures of three bedrooms: master bedroom temperature T_{mst}, secondary bedroom 1 temperature T_{scd1}, and secondary bedroom 2 temperature T_{scd2}. Therefore, the state S_t can be represented as Formula 1:

$$S_t = \{E_t, T_{mst}, T_{scd1}, T_{scd2}\} \tag{1}$$

Reward: In Ref. [47], the residential schedule is set as follows: on weekdays, people are away from 8:00 to 19:00 and indoors from 19:00 to 8:00 the next day. On weekends, people are indoors all day. The



(a)



(b)

Fig. 11. Temperature prediction of secondary bedroom 1 of Strategy 1.

reward differs when the air conditioning is on and off during these time periods.

During indoor hours, the reward is calculated based on Formulas 2-4. The reward consists of the energy consumption indicator ekpi and the indoor thermal comfort indicator tkpi. λ is a weight that can adjust the importance of ekpi and tkpi in the reward. A larger λ value means ekpi is more emphasized in the reward, and the agent tends to evolve towards energy-saving. According to our survey in China, more than half of building users consider both the environment and energy conservation equally important. Therefore, in our experiments, λ is set to 1. In practicable application, λ can be adjusted by users of the building to maintain their demands for energy saving or thermal comfort.

In tkpi, we consider the room temperatures of the three rooms. T_{exp} is the desired temperature that we expect the rooms to reach. The design heating temperature for this building is 18–24 °C. To ensure energy efficiency while considering comfort, we set the expected temperature to 20 °C for training the RL agent. Since the quadratic function is symmetric about the vertex, if we only consider energy-saving effects and set the temperature to the lower limit of comfort, it may induce temperatures below the comfortable range. For example, if the expected temperature is set to 18 °C, both 17 °C and 19 °C room temperatures will yield the same tkpi. Under the influence of ekpi, the model may prefer to choose 17 °C as the indoor temperature, which does not align well with our requirements. Theoretically, the minimum value of tkpi can be 0. η_2 is the amplification coefficient for tkpi.

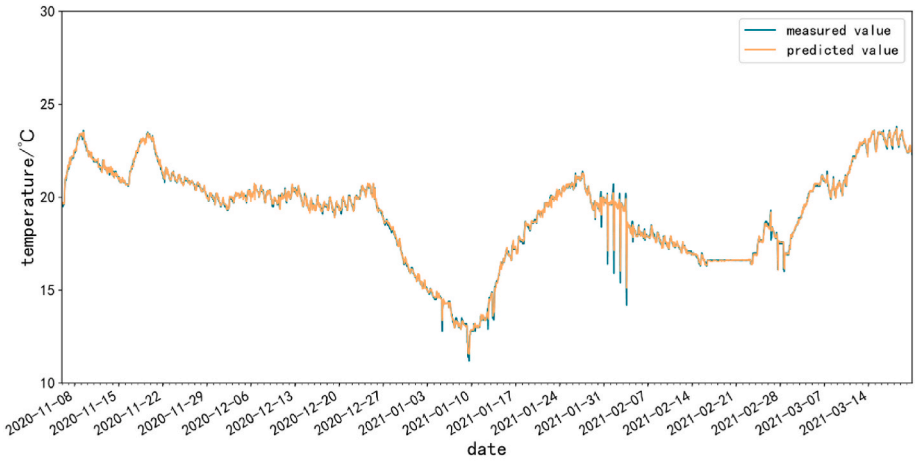
In ekpi, E_min is the minimum value obtained from energy consumption prediction by the data-driven model in pre-experiments. Due to the fluctuations in data-driven models, it is not ruled out that there may be energy consumption smaller than E_{min} in subsequent prediction results. However, this coefficient already allows ekpi to theoretically reach 0, similar to tkpi. η_1 is the amplification coefficient for ekpi. Multiplying ekpi and tkpi by different amplification coefficients is to adjust the significant difference in magnitude caused by the unit difference, so that both have a considerable impact on the agent when λ is 1. F is an activation function to facilitate the convergence of RL mathematically. in practical applications, other commonly used activation functions can be chosen based on the requirements of their problem. The introduction of η_1 and η_2 is to solve the problem that the ekpi and the tkpi have different physical quantities, measured in kWh and °C. Direct addition, subtraction, multiplication, or division of these quantities would result in a significant disparity in the magnitude of the reward values. So η_1 and η_2 are determined by the developers and cannot be adjusted further.

$$ekpi = -\eta_1 \times (f(E_t - E_{min})) \tag{2}$$

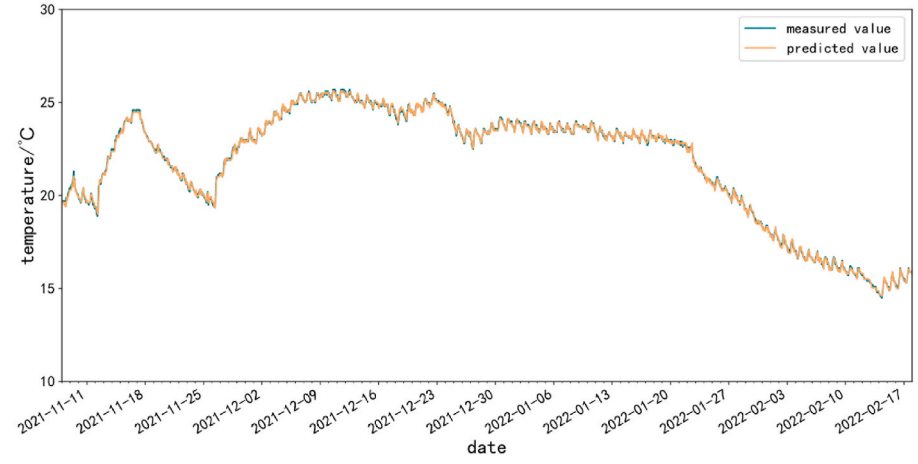
$$tkpi = -\eta_2 \times \left[(T_{mst} - T_{exp})^2 + (T_{scd1} - T_{exp})^2 + (T_{scd2} - T_{exp})^2 \right] \tag{3}$$

$$reward = \lambda \times ekpi + tkpi \tag{4}$$

During the shutdown period, the reward is 0.



(a)



(b)

Fig. 12. Temperature prediction of secondary bedroom 2 of Strategy 1.

Table 6
Energy prediction and indoor temperature prediction performance of XGBoost models for Strategy 1 after dimensionality reduction.

Prediction model	Energy	Master bedroom temperature	Temperature of secondary bedroom 1	Temperature of secondary bedroom 2
R^2	0.8634	0.9887	0.9913	0.9953
RMSE	0.2423	0.3051	0.2864	0.2087

Table 7
Energy prediction and indoor temperature prediction performance of LSTM models for Strategy 1 after dimensionality reduction.

Prediction model	Energy	Master bedroom temperature	Temperature of secondary bedroom 1	Temperature of secondary bedroom 2
R^2	−0.0342	−0.0273	−0.0247	−0.0061
RMSE	0.6526	2.8503	3.1221	3.0457

Action: Our action target is the room temperature setpoint, which is also the most common control point for residential air conditioning. Since we will train RL agents using both discrete action space and continuous action space algorithms, when training agents with a discrete action space, the action space is defined as Formula 5.

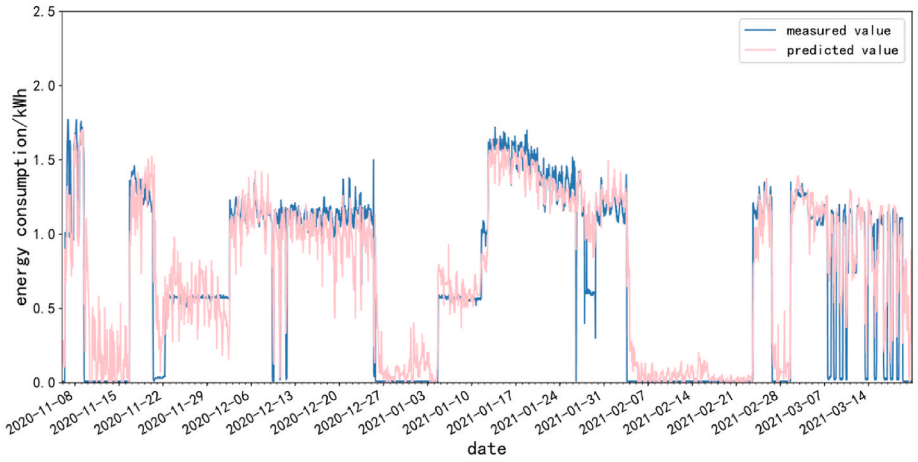
$$Action\ Space = \{18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28\} \quad (5)$$

When training agents with a continuous action space, the action space is defined as formula 6.

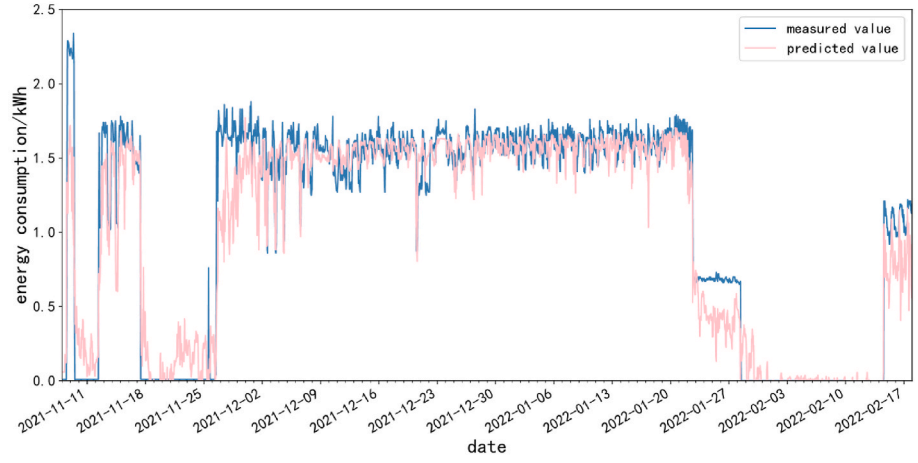
$$Action\ Space = [18, 28] \quad (6)$$

The temperature set point in both equations is in °C. Due to the difficulty in accurately predicting data that has never been seen before with data-driven methods, the action space selected in Equation 5is slightly smaller than the actual range that can be controlled by the household air conditioner, and is based on the distribution of data in the historical data.

Strategy 2:Strategy 2 only models the period when the air conditioner compressor is turned on. In application, the on/off of the air conditioner needs to be manually controlled, and whether the compressor is turned on in the on state of the air conditioner can be determined by the air conditioner's self-control system. Therefore, when training the energy consumption prediction model and room temperature prediction model, only the compressor-on periods in the historical data were extracted and the rest were excluded. The definition of the state and action in Strategy 2 is the same as in Strategy 1, but there is no off period, and the reward is calculated using Equations 1-3during the training process.



(a)



(b)

Fig. 13. Energy prediction of Strategy 1 after dimensionality reduction.

2.3. Comparison with the original control logic

To compare the effect of the control logic generated by RL, we analyze the historical data of the room temperature set point (return air temperature) and selecte the room temperature set point with the highest frequency of occurrence. We input this temperature into the environment we constructe and obtaine the energy consumption and indoor temperature under this temperature in the reserved test set, and compare it with the energy consumption and indoor temperature under the control logic generate by RL.

3. Evaluation

3.1. Comparison of the results of building the environment with two strategies

3.1.1. XGBoost energy prediction model and indoor temperature prediction model

In this section, we will compare the energy consumption prediction model and the room temperature prediction model generated by the two strategies. As described earlier, each strategy will produce one energy consumption prediction model and three room temperature prediction models. In the energy consumption prediction of Strategy 1, the selected feature parameters include the current temperature in the master bedroom, the current temperature in the secondary bedroom 1, the current temperature in the secondary bedroom 2, the current exhaust temperature, the current supply temperature, the current fresh air temperature, the room temperature set point (return air temperature) for the next time step, and the outdoor temperature for the next time step. In the room temperature prediction, we selecte the current indoor temperature, the current wall temperature, and the current insulation layer temperature, the room temperature set point (return air temperature) for the next time step and the outdoor temperature. We obtaine a total of 5854 h of data over two years. To prevent data leakage during RL agent training, we use the week from February 18, 2022 to February 24, 2022 as the training set for the RL agent, and a total of 5686 h of data are used to train the energy consumption prediction model and indoor temperature prediction model.

R^2 score and RMSE are the indicators to evaluate the performance of the XGBoost models and LSTM models. All LSTM models have performed parameter tuning using a grid search method. The ranges of parameter variations for the grid are shown in Table 3. The R^2 score and RMSE of the four models are shown in Table 4 and Table 5. As can be seen from Table 4, the R^2 of the XGBoost model for energy consumption prediction we trained can reach 0.9537 and its RMSE is 0.1454, and the R^2 of the XGBoost models for room temperature prediction are all above 0.99 and their RMSE are less than 0.3. Figs. 9–12 show the results of the XGBoost models of air conditioning energy consumption and room temperature prediction of Strategy 1 in the winter of 2020 and 2021. They also show that the model captures the changes in energy

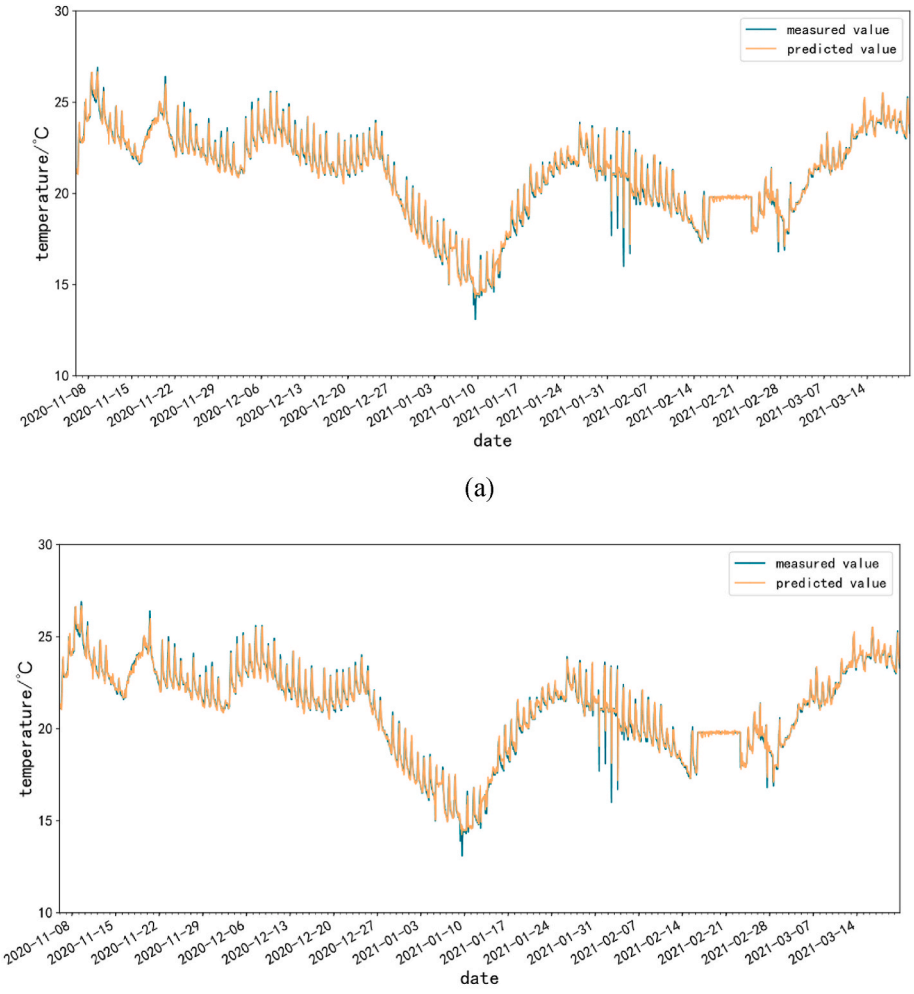


Fig. 14. Temperature prediction of master bedroom of Strategy 1 after dimensionality reduction.

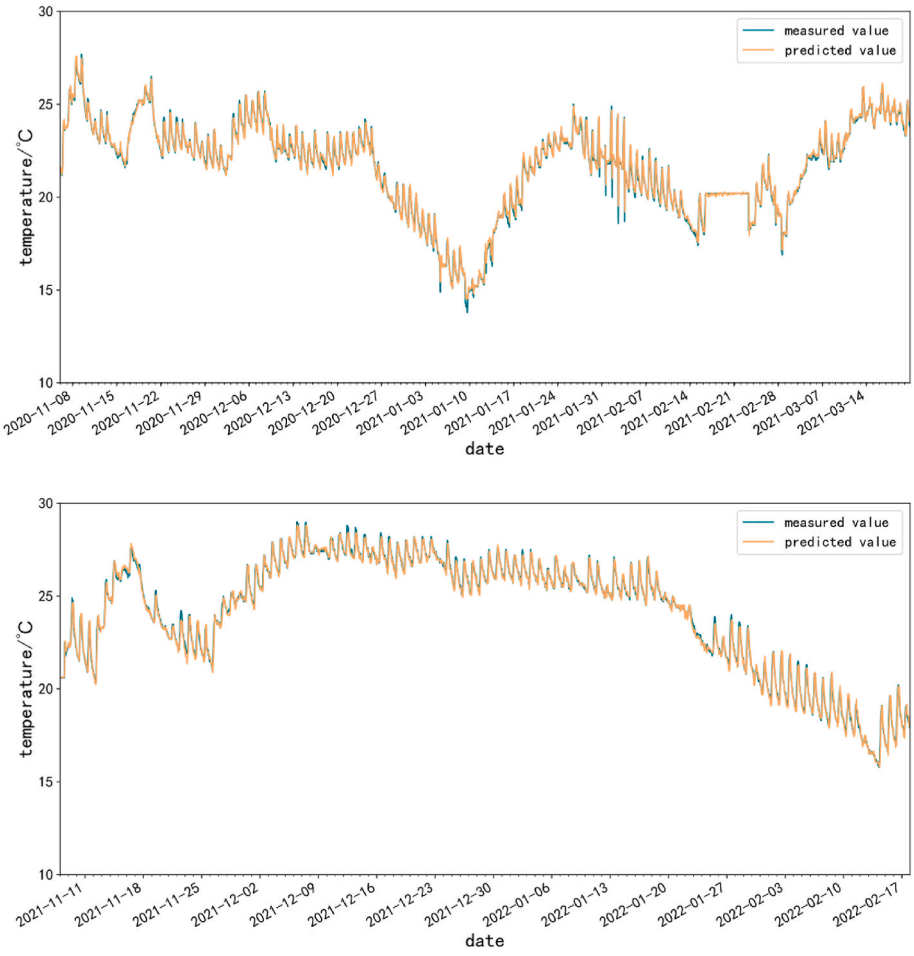
consumption and room temperature well. However, the R^2 score and RMSE of LSTM models for energy consumption prediction and indoor air temperature prediction indicate the training process of these models is not converged, which cannot be used further.

However, in the subsequent environment construction, we find that when building the RL environment, all parameters within that environment must be iterable. If not, training will be difficult. Although based on prior knowledge, we know that wall temperature and insulation layer temperature have a certain impact on the prediction of indoor temperature, and operational parameters of the system such as supply temperature, exhaust temperature, and return air temperature will also have an impact on air conditioning energy consumption prediction. While these parameters are easily obtainable in a real measurement environment, they are difficult to acquire in a data-driven environment due to the challenges in iteration. Therefore, in the subsequent energy consumption prediction model training, we exclude the supply temperature, exhaust temperature, and fresh air temperature of the system, and only retain the current indoor temperatures of the three bedrooms, the next time step's room temperature set point (return air temperature), and the outdoor temperature. For indoor temperature prediction, we exclude the wall temperature and insulation layer temperature, and only retain the current indoor temperature, the next time step's room temperature set point (return air temperature), and the outdoor temperature. Tables 6 and 7 and Figs. 13–16 demonstrate the changes in model performance after excluding the aforementioned parameters.

According to Table 6 and it can be observed that the performance of the XGBoost model for energy consumption prediction is significantly affected after excluding the operational parameters of the system. The model's performance decreased by approximately 0.09 of R^2 score and increase 0.1 of RMSE. It is unable to accurately identify the on/off states, which can potentially interfere with the training of the RL agent. On the other hand, the performance of the indoor temperature prediction model is minimally affected. The prediction performance of the master bedroom temperature even slightly increase after dimensionality reduction. This indicates that wall temperature has little impact on the prediction of room temperature in that particular room. This can be attributed to the thick insulation layer in nearly zero-energy buildings, which provides strong insulation against external heat disturbances and results in minimal temperature difference between the wall and the indoor environment. The slight increase in performance for the master bedroom may be attributed to the different random seeds used during the training of the XGBoost model. Figs. 13–16 displayed the prediction results of these XGBoost models.

In Table 7, the R^2 score and RMSE of the LSTM models for energy consumption prediction and indoor air temperature prediction are rather poor. These models are still not converged.

Although excluding certain parameters has a significant impact on energy consumption prediction, for the sake of model iteration, we can only perform the aforementioned treatment in this experiment.



(b)

Fig. 15. Temperature prediction of secondary bedroom 1 of Strategy 1 after dimensionality reduction.

However, there may be potential benefits in training the RL model with the excluded parameters.

In the subsequent modeling of Strategy 2, we only conducte modeling based on dimensionality reduction. Since Strategy 2 focuses solely on modeling the periods when the air conditioning compressor is activated, we exclude the periods when the compressor is not in operation during the modeling of the energy consumption and temperature prediction models for Strategy 2. As a result, we retaine 3966 time hours, with a consecutive set of 144 h reserved as the test set for the RL agent model. A total of 3822 h are used for training the energy consumption and temperature prediction models. Tables 8 and 9 and Figs. 17–20 present the results of the energy consumption and temperature prediction models for Strategy 2.

From Table 8 and it can be observed that after excluding the periods when the air conditioning is not activated, the R^2 score of XGBoost model for energy consumption prediction improve to over 0.9 and its RMSE is 0.1042 even without the device operating parameters. But the LSTM models are not converged in Table 9.

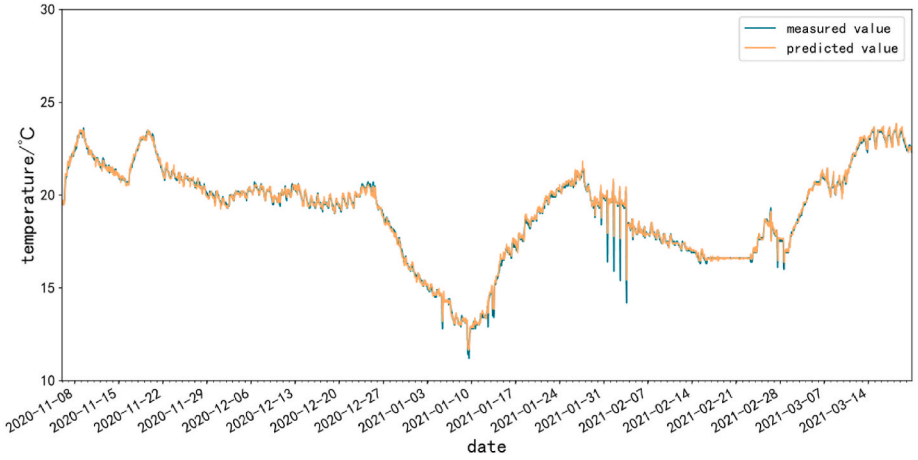
Fig. 17 also demonstrates good fitting performance for energy consumption, with the prediction error reduced by only considering the energy consumption during the air conditioning activation periods.

As for indoor temperature prediction, Strategy 2 achieve similar performance to Strategy 1. Both strategies demonstrated high prediction performance, which provide a foundation for training the RL agent.

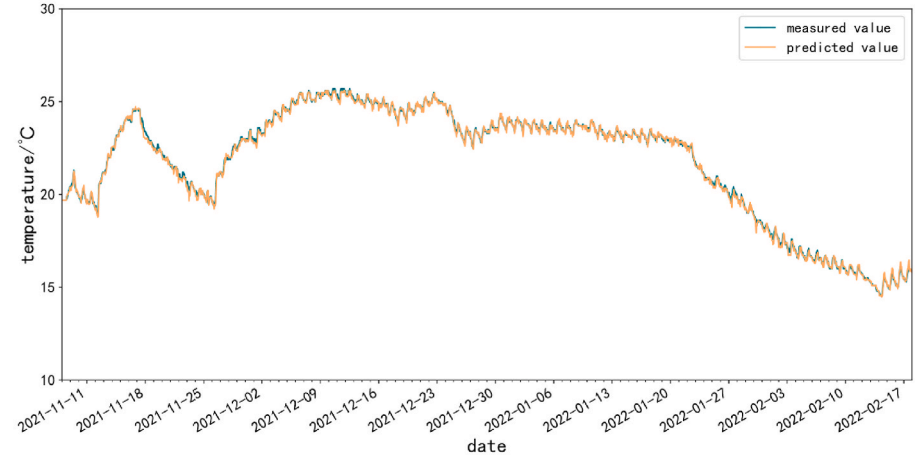
3.1.2. Training results of DQN and DDPG agent models

We encapsulate the XGBoost energy consumption and temperature prediction models generated from the above two strategies into an OpenAI Gym environment and test the DQN and DDPG models trained on these environments. The training step is set to 300,000 steps, and the following are the test results. The hyperparameters of DQN and DDPG training are described in Table 10 after hyperparameter tuning. In the DQN model, our ε value in greedy policy is dynamic. The value of ε linearly changes from 1 to 0.05. This is because, during the training process of RL, a higher exploration probability is needed in the early stages of training to explore the action space. Reducing the exploration probability later in training aids the convergence of the model. DDPG algorithm uses the policy gradient network to replace the ε-greedy policy so there is not an ε in it. The training dataset of RL consists of the training data of the aforementioned XGBoost model, while the testing dataset consists of additional data that is not used to train the XGBoost model in order to prevent data leakage. The variation in data corresponds to differences in initialization conditions, thus making our training and testing environments distinct. Furthermore, since the data utilize in the testing environment has never been employed during training, our testing environment effectively reflects the control performance of the RL agent in a novel setting. In each episode, the model will be tested after training. And the action values of the observed values in the training and testing sets will be recorded. At the beginning of each episode, the OpenAI Gym environment is initialized for the next training iteration.

Firstly, DQN is trained for Strategy 1. Fig. 21 displays the reward, ekpi, and tkpi for Experiment 1 under DQN. It can be observed that during the training of DQN, rewards and ekpi/tkpi fluctuate within a



(a)



(b)

Fig. 16. Temperature prediction of secondary bedroom 1 of Strategy 1 after dimensionality reduction.

Table 8
Energy prediction and indoor temperature prediction performance of XGBoost models for Strategy 2.

Prediction model	Energy	Master bedroom temperature	Temperature of secondary bedroom 1	Temperature of secondary bedroom 2
R^2	0.9181	0.9884	0.9916	0.9958
RMSE	0.1042	0.3101	0.2739	0.2006

Table 9
Energy prediction and indoor temperature prediction performance of LSTM models for Strategy 2.

Prediction model	Energy	Master bedroom temperature	Temperature of secondary bedroom 1	Temperature of secondary bedroom 2
R^2	−0.0151	−0.0337	−0.0176	−0.0370
RMSE	0.3463	3.1423	3.0944	3.0301

large range, without showing a stable convergence trend. However, after stabilization, periodic fluctuations become evident. DQN does not yield satisfactory training results in this environment.

Fig. 22 displays the training results of the DDPG agent for Strategy 1. Similar to DQN, DDPG also exhibits a periodic pattern in the later stages of training, but with a reduced fluctuation range compared to DQN.

Considering the training results of both agents mentioned above, it can be observed that Strategy 1 fails to guide the training of a stable and useable DQN or DDPG agent. There could be two possible reasons for this phenomenon. Firstly, in Strategy 1, when training the energy consumption and temperature prediction models, we require a model to provide predictions for both the air conditioning on and off states. However, when entering the training of the RL agent, the model cannot accurately determine the air conditioning's current state, resulting in a significant impact on the predictions of energy consumption and temperature. This is highly detrimental to the training of the RL agent model.

Secondly, during the off-state, we set the reward at any given time step to be 0. This implies that the model's training does not stop during the off-state. However, since the model realizes that any room temperature setting will yield the same reward feedback, it may randomly guess the desired air conditioning temperature. The guessed action values may be related to the training cycle, which could explain the periodic changes observed in the later stages of training for the DDPG model.

In summary, due to the issues in the strategy design, Strategy 1 fails to properly induce the training of the RL agent. The data-driven RL environment cannot achieve automatic control by modeling multiple states using a single model.

Fig. 23 illustrates the training results of the DQN agent model in the Strategy 2 environment. From Fig. 23, it can be observed that after an

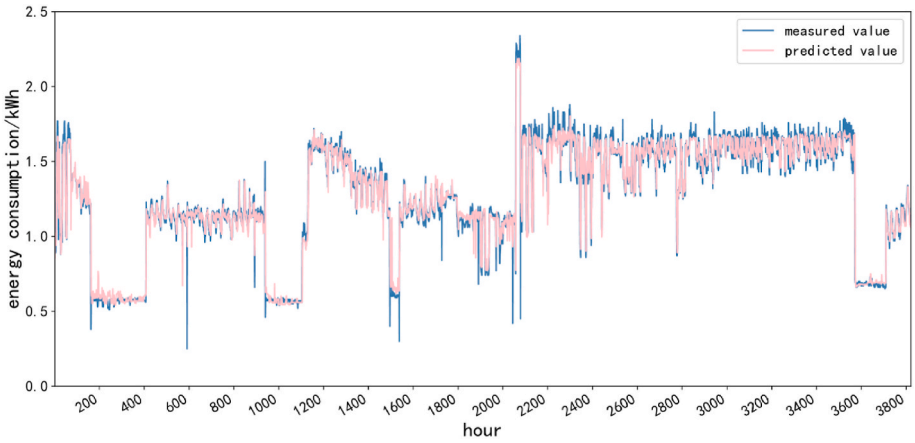


Fig. 17. Energy prediction of Strategy 2.

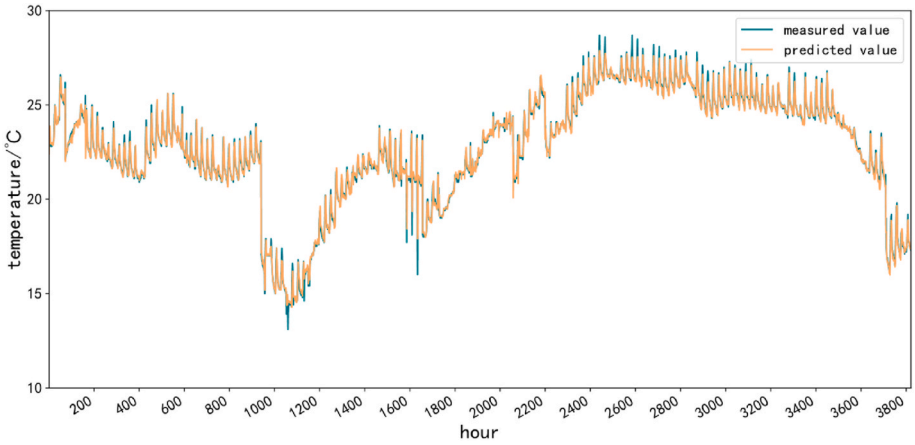


Fig. 18. Temperature prediction of master bedroom of Strategy 2.

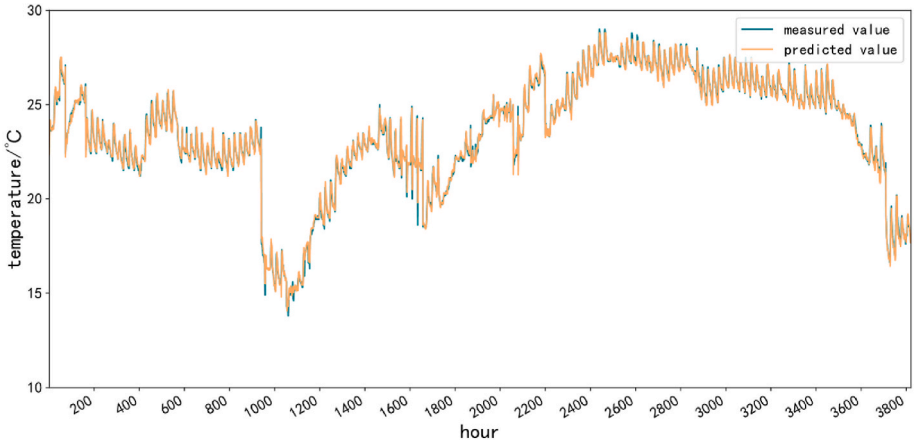


Fig. 19. Temperature prediction of secondary bedroom 1 of Strategy 2.

initial large oscillation, the training process exhibits periodic oscillations that persist for a while before expanding into a larger range. This indicates that the model eventually diverges, and the environment fails to train an effective DQN agent model.

Fig. 24 presents the training progress of the DDPG agent model in Strategy 2. By considering the rewards, ekpi, and tkpi, we can observe that after an initial period of intense action oscillations, the three metrics gradually increase and stabilize, indicating the model's convergence. However, around step 110,000, the model starts to overfit, leading to a slight increase in the oscillation range for the three metrics. Therefore, according to Fig. 24, the model achieves optimal performance at approximately 110,000 steps. Thus, we will select the observed values and action values of the testing process at around 110,000 steps for further analysis.

Fig. 25 shows the temperatures of the master bedroom, secondary bedroom 1, secondary bedroom 2, room temperature setpoint, and

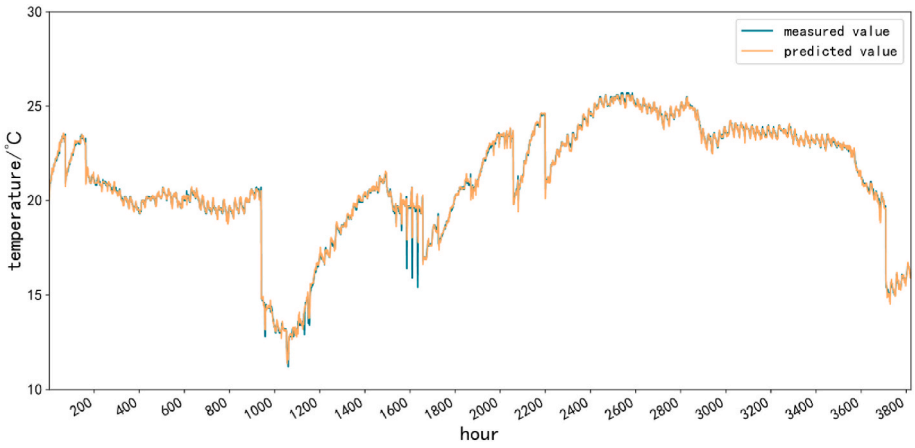


Fig. 20. Temperature prediction of secondary bedroom 2 of Strategy 2.

Table 10
Hyperparameters of DQN and DDPG training.

	Policy	Learning rate	Discount factor	initial value of ε	final value of ε
DQN	Mlp	0.0005	0.99	1	0.05
DDPG	Mlp	0.0005	0.99	–	–

outdoor temperature at step 110,000 of the test set. At this point, the master bedroom temperature remains within the range of 18.4 °C–19.0 °C, while secondary bedroom 1 ranges from 18 °C to 18.5 °C, which is closer to the desired temperature of 20 °C set during the training process. However, secondary bedroom 2 exhibits lower temperatures, ranging from 16.5 °C to 17.5 °C, with a larger deviation from the desired temperature.

Comparing the historical data in Figs. 18–20, it can be observed that at the same moment, the temperature in secondary bedroom 2 is generally lower than that in the master bedroom and secondary bedroom 1, with a difference of approximately 1.5 °C–2 °C. Referring to the floor plan in Fig. 2, this is because secondary bedroom 2 is adjacent to a non-air-conditioned area, namely the bathroom, while the master bedroom and secondary bedroom 1 are adjacent to each other and surrounded by air-conditioned areas. Secondary bedroom 2 is more influenced by the non-air-conditioned area, resulting in higher heat dissipation compared to the other two rooms.

Additionally, consistent with the theoretical analysis in section 2.2.2, when we set the desired temperature to 20 °C, none of the three rooms fully reach 20 °C, but they generally meet the lower limit of the heating comfort standard at 18 °C. This confirms that when we assigned equal weights to ekpi and tkpi during training, the agent tended to prioritize energy efficiency over comfort to some extent.

Furthermore, comparing the DQN and DDPG algorithms, it can be observed that DQN still fails to converge in Strategy 2, possibly due to its small action space of only 11 values. This limited range of options in the DQN model is insufficient for addressing the complexity of our actual problem, resulting in difficulty in convergence. DQN performs poorly on this problem.

Moreover, the control strategy of the DDPG agent does not exhibit significant fluctuations within the given period. This is because this study focuses on the heating condition of a nearly zero-energy building during winter, where the building has good insulation and therefore experiences minimal fluctuations in indoor temperature. In conventional buildings, the adjustability of the RL agent model is likely to be more effective.

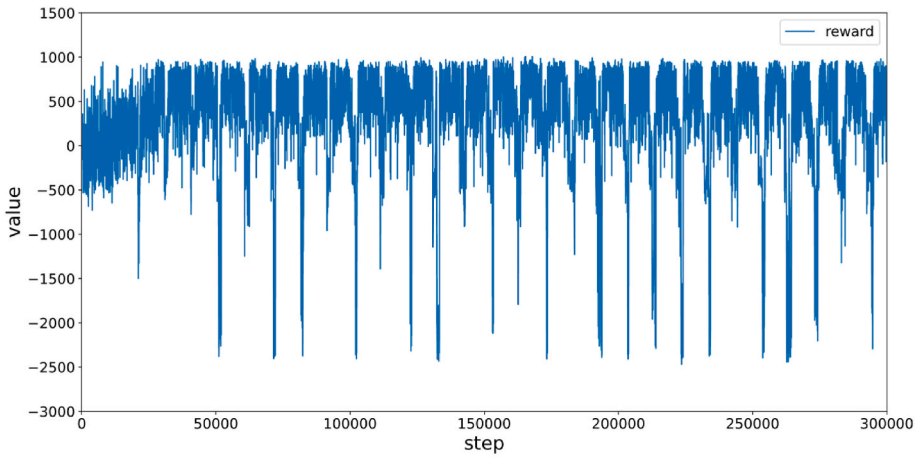
3.2. Comparison with the original control logic

Fig. 26 shows the probability density curve of the room temperature setpoint (return air temperature) in the historical data. From the curve, it can be observed that the original control logic had the highest probability density at a room temperature setpoint of 26.2 °C. Table 11 presents the energy consumption comparison in the test set under DDPG control compared to the 26.2 °C control. According to Table 11, the energy consumption in the building under DDPG control was 10.06% lower compared to the 26.2 °C control during the test set time range.

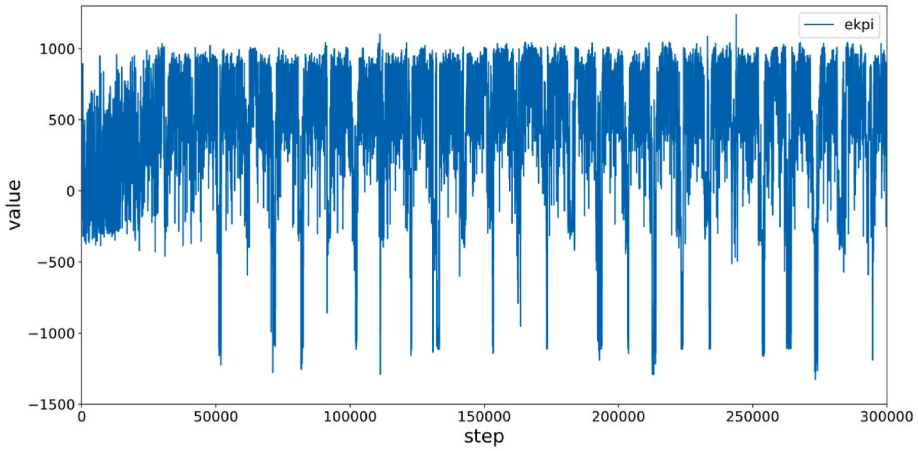
4. Conclusion

In this study, two strategies for energy-saving control of a residential building are established, and XGBoost models for energy consumption prediction and indoor temperature prediction are chosen after comparsion with LSTM model under these strategies, because none of LSTM models converged. Under the aforementioned strategies, two RL models, DQN with a discrete action space and DDPG with a continuous action space, are trained. The main conclusions are as follows:

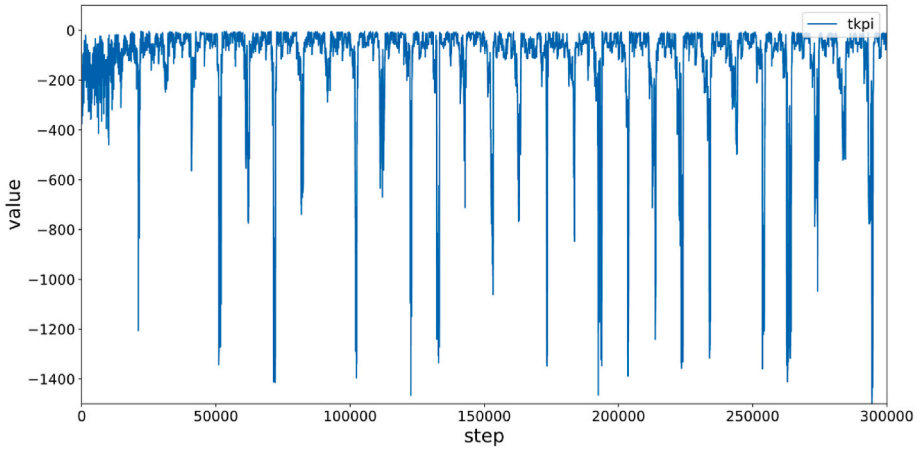
- Without considering the iterability of data, the performance of the energy consumption prediction model trained under Strategy 1 is 0.9537 of R^2 score and 0.1454 of RMSE, and the performance of the temperature prediction models for the master bedroom, secondary bedroom 1, and secondary bedroom 2 is 0.9905 of R^2 score and 0.2784 of RMSE, 0.9920 of R^2 score and 0.2698 of RMSE, 0.9975 of R^2 score and 0.1532 of RMSE, respectively. However, there are non-iterable parameters in the training parameters of the above models, such as equipment operating parameters and room wall temperatures. After excluding these parameters, the performance of the energy consumption model under Strategy 1 is 0.8634 of R^2 score and 0.2423 of RMSE., and the performance of the temperature prediction for the three rooms is 0.9887 of R^2 score and 0.3051 of RMSE, 0.9913 of R^2 score and 0.2864 of RMSE, 0.9953 of R^2 score and 0.2087 of RMSE, respectively. The energy consumption prediction performance under Strategy 2 is 0.9181 of R^2 score and 0.1042 of RMSE, and the temperature prediction performance for the three rooms is 0.9884 of R^2 score and 0.3101 of RMSE, 0.9916 of R^2 score and 0.2739 of RMSE, 0.9958 of R^2 score and 0.2006 of RMSE. Equipment operating parameters have a significant impact on the performance of air conditioning energy consumption prediction.
- Strategy 1 does not induce a useable RL agent in both DQN and DDPG training. This is because a single model is used to represent two processes, making it difficult for the RL agent to determine the current on/off state. Additionally, during the off state, the model training do not stop, but the rewards are all 0, causing the RL agent to



(a) Reward of DQN of Strategy 1



(b) Ekpi of DQN of Strategy 1

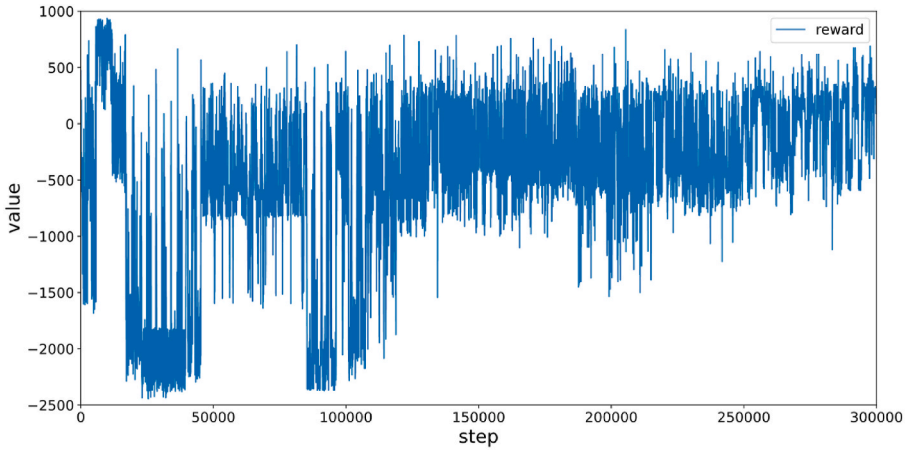


(c) Tkpi of DQN of Strategy 1

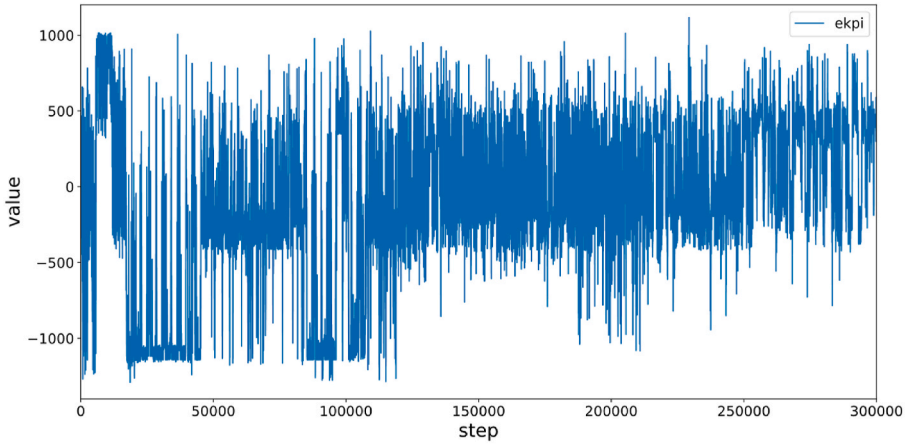
Fig. 21. DQN training result of Strategy 1.

make random guesses about the air conditioning temperature setpoints.

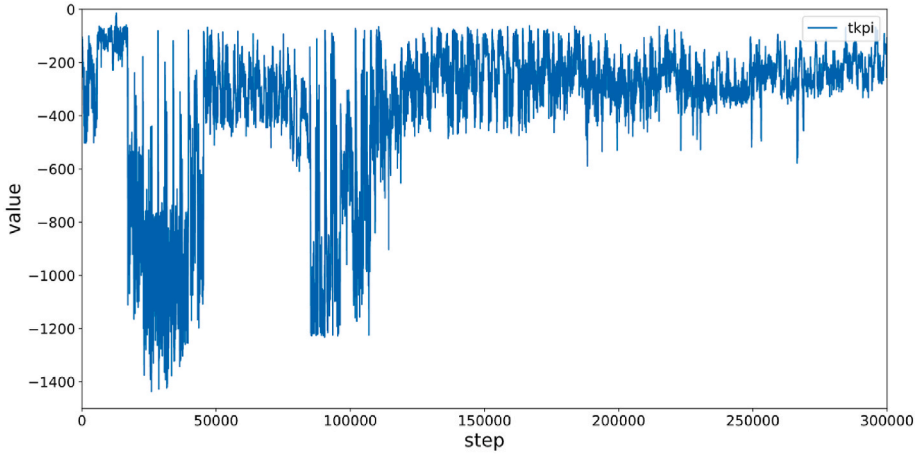
- In the DQN training of Strategy 2, a useable agent is not induced. However, in the DDPG training, the RL algorithm converge and generate a useable agent model. This is because the action space of the DQN algorithm is relatively small for the problem at hand, and the model failes to converge.
- In the test set, the DDPG agent trained under Strategy 2 achieve 10.06% energy savings compared to the original fixed temperature control at 26.2 °C, while ensuring indoor thermal comfort.



(a) Reward of DDPG of Strategy 1



(b) Ekpi of DDPG of Strategy 1



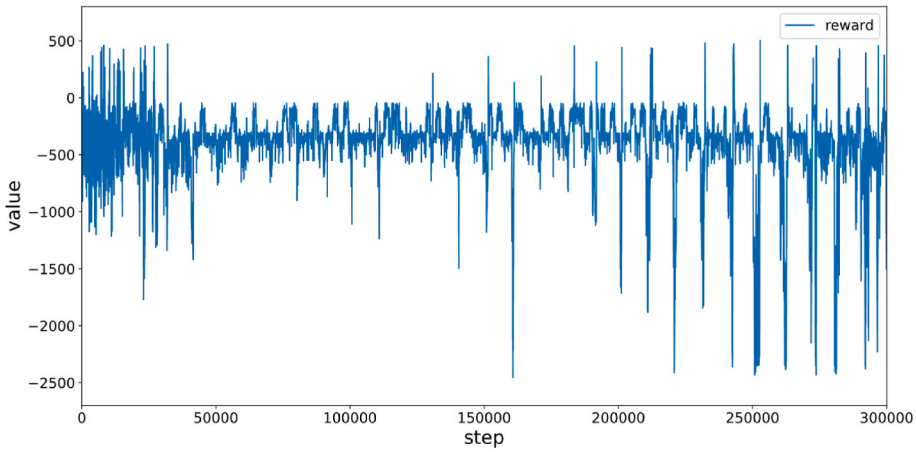
(c) Tkpi of DDPG of Strategy 1

Fig. 22. DDPG training result of Strategy 1.

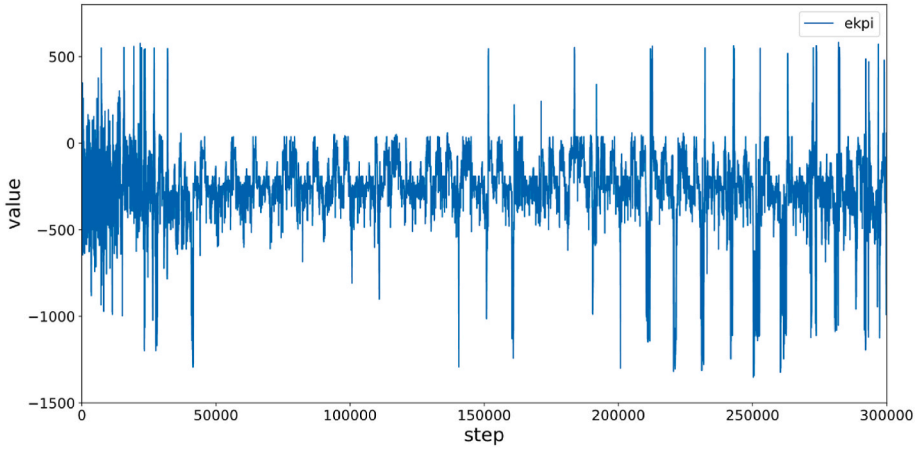
5. Discussion

In this study, a data-driven approach is used to construct an RL environment for an experimental nearly zero-energy building and verify its feasibility. However, this approach may encounter the following issues in practical applications.

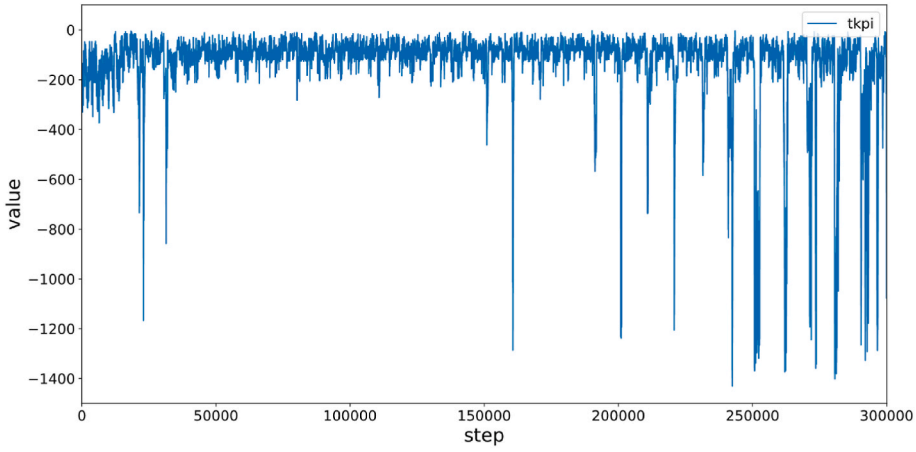
First, when training the RL agent model using this approach, the action space cannot exceed the range of actions observed in the



(a) Reward of DQN of Strategy 2



(b) Ekpi of DQN of Strategy 2

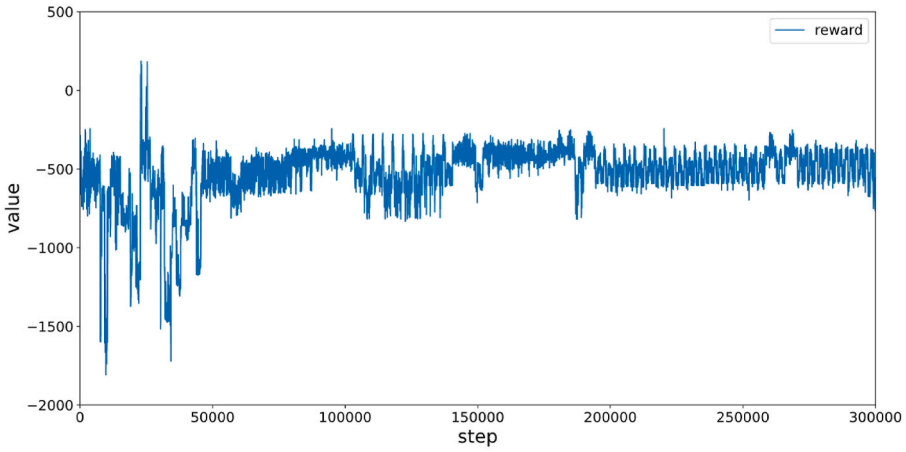


(c) Tkpi of DQN of Strategy 2

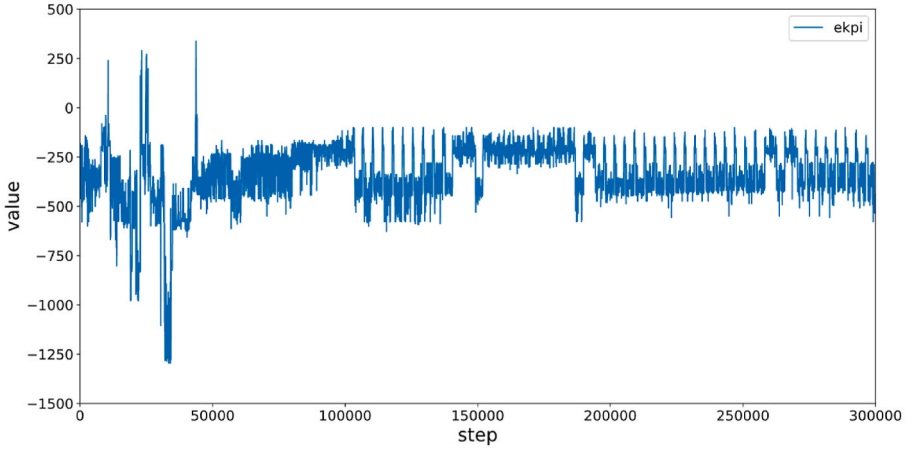
Fig. 23. DQN training result of Strategy 2.

historical data. In this case, the range of room temperature setpoints cannot include data that has not appeared in the historical data. This is because data-driven methods learn based on available data and may not perform well on unseen data.

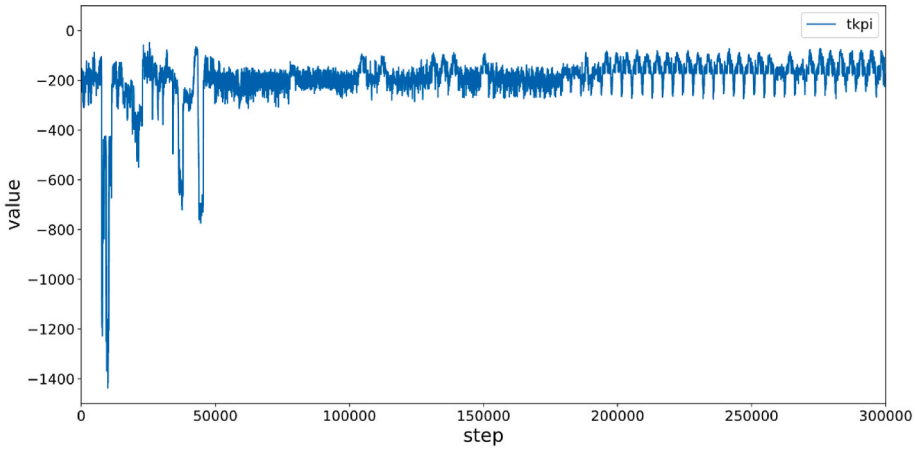
Second, the iterability of parameters needs to be considered in the selection of observation values. In this case, although the operating parameters of the air conditioning system can positively impact energy consumption prediction, we cannot obtain these parameters during iterative data-driven training. Therefore, they have to be excluded, resulting in a sacrifice of prediction accuracy. However, in an experimental environment, this problem can be solved, and its impact on the training of RL agent models will be further reduced.



(a) Reward of DDPG of Strategy 2



(b) Ekpi of DDPG of Strategy 2



(c) Tkpi of DDPG of Strategy 2

Fig. 24. DDPG training result of Strategy 2.

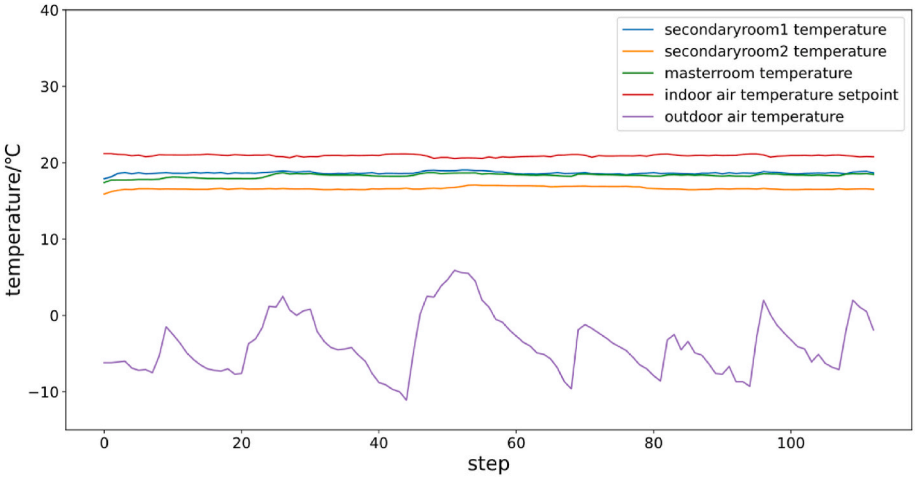


Fig. 25. Indoor and outdoor temperature of DDPG of Strategy 2.

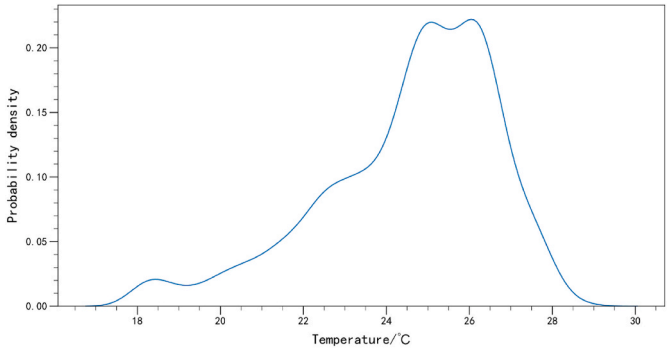


Fig. 26. Probability indensity of return air temperature in historical data.

Table 11
Comparison between DDPG controller and original temperature setpoint.

	DDPG controller	Original temperature setpoint	Energy saving
Accumulative energy consumption of heat pump (kWh)	132.26	147.07	10.06%

Thirdly, although the research in this paper is based on a nearly zero-energy residential building, the research methodology can theoretically be extended to conventional residential buildings. The generalization of this method would require more case studies and data from conventional residential buildings.

CRediT authorship contribution statement

Man Wang: Writing – original draft, Visualization, Methodology, Investigation. **Borong Lin:** Writing – review & editing.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

Data availability

The data that has been used is confidential.

Acknowledgments

The study was supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No.51825802).

References

[1] A. Afram, F. Janabi-Sharifi, Theory and applications of HVAC control systems–A review of model predictive control (MPC), Build. Environ. 72 (2014) 343–355.
[2] S. Carlucci, G. Cattarin, F. Causone, L. Pagliano, Multi-objective optimization of a nearly zero-energy building based on thermal and visual discomfort minimization using a non-dominated sorting genetic algorithm (NSGA-II), Energy Build. 104 (2015) 378–394.
[3] F.P. Chantrelle, H. Lahmidi, W. Keilholz, M. El Mankibi, P. Michel, Development of a multicriteria tool for optimizing the renovation of buildings, Appl. Energy 88 (4) (2011) 1386–1394.
[4] Y. Bichiou, M. Krarti, Optimization of envelope and HVAC systems selection for residential buildings, Energy Build. 43 (12) (2011) 3373–3382.
[5] K.F. Fong, V.I. Hanby, T.T. Chow, HVAC system optimization for energy management by evolutionary programming, Energy Build. 38 (3) (2006) 220–231.
[6] R. Ooka, K. Komamura, Optimal design method for building energy systems using genetic algorithms, Build. Environ. 44 (7) (2009) 1538–1544.
[7] V. Congradac, F. Kulic, HVAC system optimization with CO2 concentration control using genetic algorithms, Energy Build. 41 (5) (2009) 571–577.
[8] W. Huang, H.N. Lam, Using genetic algorithms to optimize controller parameters for HVAC systems, Energy Build. 26 (3) (1997) 277–282.
[9] L. Lu, W. Cai, L. Xie, S. Li, Y.C. Soh, HVAC system optimization—in-building section, Energy Build. 37 (1) (2005) 11–22.
[10] N. Nassif, S. Kajl, R. Sabourin, Optimization of HVAC control system strategy using two-objective genetic algorithm, HVAC R Res. 11 (3) (2005) 459–486.
[11] S.R. West, J.K. Ward, J. Wall, Trial results from a model predictive control and optimisation system for commercial building HVAC, Energy Build. 72 (2014) 271–279.
[12] D. Blum, Z. Wang, C. Weyandt, D. Kim, M. Wetter, T. Hong, M.A. Piette, Field demonstration and implementation analysis of model predictive control in an office HVAC system, Appl. Energy 318 (2022), 119104.
[13] D. Sturzenegger, D. Gyalistras, M. Morari, R.S. Smith, Model predictive climate control of a swiss office building: implementation, results, and cost–benefit analysis, IEEE Trans. Control Syst. Technol. 24 (1) (2015) 1–12.
[14] S. Yang, M.P. Wan, B.F. Ng, S. Dubey, G.P. Henze, S.K. Rai, K. Baskaran, Experimental study of a model predictive control system for active chilled beam (ACB) air-conditioning system, Energy Build. 203 (2019), 109451.

[15] J. Široký, F. Oldewurtel, J. Cigler, S. Prívara, Experimental analysis of model predictive control for an energy efficient building heating system, Appl. Energy 88 (9) (2011) 3079–3087.
[16] Y. Ma, J. Matuško, F. Borrelli, Stochastic model predictive control for building HVAC systems: complexity and conservatism, IEEE Trans. Control Syst. Technol. 23 (1) (2014) 101–116.
[17] P. Li, D. Vrabie, D. Li, S.C. Bengea, S. Mijanovic, Z.D. O'Neill, Simulation and experimental demonstration of model predictive control in a building HVAC system, Sci. Technol. Built Environ. 21 (6) (2015) 721–732.
[18] B. Dong, K.P. Lam, February). A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting, in: Building Simulation, vol. 7, Springer Berlin Heidelberg, 2014, pp. 89–106.
[19] A. Abida, P. Richter, HVAC control in buildings using neural network, J. Build. Eng. 65 (2023), 105558.
[20] S. Taheri, A. Ahmadi, B. Mohammadi-Ivatloo, S. Asadi, Fault detection diagnostic for HVAC systems via deep learning algorithms, Energy Build. 250 (2021), 111275.
[21] M. Esrafilian-Najafabadi, F. Haghighat, Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings, Energy Build. 252 (2021), 111377.
[22] S. Xu, Y. Wang, Y. Wang, Z. O'Neill, Q. Zhu, November). One for many: transfer learning for building hvac control, in: Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, 2020, pp. 230–239, and transportation.
[23] Z. Wang, T. Hong, M.A. Piette, Data fusion in predicting internal heat gains for office buildings through a deep learning approach, Appl. Energy 240 (2019) 386–398.
[24] Y. Du, H. Zandi, O. Kotevska, K. Kurte, J. Munk, K. Amasyali, F. Li, Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning, Appl. Energy 281 (2021), 116117.
[25] 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.
[26] Z. Zou, X. Yu, S. Ergan, Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network, Build. Environ. 168 (2020), 106535.
[27] Z. Jiang, M.J. Risbeck, V. Ramamurti, S. Murugesan, J. Amores, C. Zhang, K. H. Drees, Building HVAC control with reinforcement learning for reduction of energy cost and demand charge, Energy Build. 239 (2021), 110833.
[28] Z. Deng, Q. Chen, Reinforcement learning of occupant behavior model for cross-building transfer learning to various HVAC control systems, Energy Build. 238 (2021), 110860.
[29] V. Taboga, A. Bellahsen, H. Dagdougui, An enhanced adaptivity of reinforcement learning-based temperature control in buildings using generalized training, IEEE Transact. Emerging Topics Comput. Intell. 6 (2) (2021) 255–266.
[30] D. Wan, M. Chi, Q. Peng, Y. Yu, Z.W. Liu, May). Energy scheduling strategy of ice storage air conditioning system based on deep reinforcement learning, in: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), IEEE, 2021, pp. 846–851.
[31] B. Chen, Z. Cai, M. Bergés, Gnu-rl: a practical and scalable reinforcement learning solution for building hvac control using a differentiable mpc policy, Frontiers Built Environ 6 (2020), 562239.
[32] X. Yuan, Y. Pan, J. Yang, W. Wang, Z. Huang, Study on the application of reinforcement learning in the operation optimization of HVAC system, in: Building Simulation, vol. 14, Tsinghua University Press, 2021, February, pp. 75–87.
[33] X. Zhang, Z. Li, Z. Li, S. Qiu, H. Wang, Differential pressure reset strategy based on reinforcement learning for chilled water systems, in: Building Simulation, vol. 15, Tsinghua University Press, Beijing, 2022, February, pp. 233–248. No. 2.
[34] A.H. Hosseinloo, A. Ryzhov, A. Bischi, H. Ouerdane, K. Turitsyn, M.A. Dahleh, Data-driven control of micro-climate in buildings: an event-triggered reinforcement learning approach, Appl. Energy 277 (2020), 115451.
[35] Q. Fu, X. Chen, S. Ma, N. Fang, B. Xing, J. Chen, Optimal control method of HVAC based on multi-agent deep reinforcement learning, Energy Build. 270 (2022), 112284.
[36] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, W. Li, X. Shi, Cross temporal-spatial transferability investigation of deep reinforcement learning control strategy in the building HVAC system level, Energy 263 (2023), 125679.
[37] Y. Lei, S. Zhan, E. Ono, Y. Peng, Z. Zhang, T. Hasama, A. Chong, A practical deep reinforcement learning framework for multivariate occupant-centric control in buildings, Appl. Energy 324 (2022), 119742.
[38] T. Moriyama, G. De Magistris, M. Tatsubori, T.H. Pham, A. Munawar, R. Tachibana, Reinforcement learning testbed for power-consumption optimization, in: Methods and Applications for Modeling and Simulation of Complex Systems: 18th Asia Simulation Conference, AsiaSim 2018, vol. 18, Springer Singapore, Kyoto, Japan, 2018, pp. 45–59. October 27–29, 2018, Proceedings.
[39] C. Lork, W.T. Li, Y. Qin, Y. Zhou, C. Yuen, W. Tushar, T.K. Saha, An uncertainty-aware deep reinforcement learning framework for residential air conditioning energy management, Appl. Energy 276 (2020), 115426.
[40] J. Arroyo, C. Manna, F. Spiessens, L. Helsen, An OpenAI-gym environment for the building optimization testing (BOPTEST) framework, in: Proceedings of the 17th IBPSA Conference, 2021, September.
[41] C. Blad, S. Bøgh, C.S. Kallesøe, Data-driven offline reinforcement learning for HVAC-systems, Energy 261 (2022), 125290.
[42] M. Han, R. May, X. Zhang, X. Wang, S. Pan, Y. Da, Y. Jin, A novel reinforcement learning method for improving occupant comfort via window opening and closing, Sustain. Cities Soc. 61 (2020), 102247.
[43] S. Qiu, Z. Li, Z. Li, J. Li, S. Long, X. Li, Model-free control method based on reinforcement learning for building cooling water systems: validation by measured data-based simulation, Energy Build. 218 (2020), 110055.
[44] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System. The 22nd ACM SIGKDD International Conference, ACM, 2016.
[45] 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.
[46] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Wierstra, Continuous Control with Deep Reinforcement Learning, 2015 arXiv preprint arXiv: 1509.02971.
[47] C. Peng, D. Yan, R. Wu, C. Wang, X. Zhou, Y. Jiang, Quantitative description and simulation of human behavior in residential buildings, in: Building Simulation, vol. 5, Tsinghua Press, 2012, June, pp. 85–94.