Journal of Building Engineering 74 (2023) 106852

Contents lists available at ScienceDirect

Journal of Building Engineering

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

Comparative study of model-based and model-free reinforcement learning control performance in HVAC systems

Cheng Gao [b], Dan Wang [a,*]

[a] Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China
[b] China Academy of Building Research, Beijing, 100013, China

ARTICLE INFO

Keywords:
Reinforcement learning
Model-based control
Model-free control
Control performance
HVAC systems

ABSTRACT

Reinforcement learning (RL) shows the potential to address drawbacks of rule-based control and model predictive control and exhibits great effectiveness in heating, ventilation and air conditioning (HVAC) systems. Most studies employed model-free RL to achieve building energy conservation and increase indoor comfort. However, model-free RL algorithms face the challenge of sample efficiency which causes long-time training and restricts their applications. Model-based RL is considered an alternative avenue for accelerating learning and promoting the application of RL, but it also has limitations due to modeling approaches and accuracy. In addition, few studies propose model-based RL algorithms and investigate performance gaps between model-free and model-based RL in HVAC systems. Therefore, this study conducts a comprehensive performance comparison between model-free and model-based RL to identify the current issues with RL control in HVAC systems. The open-source building optimization testing (BOPTEST) framework is employed as the virtual environment to evaluate the control performance and computational burden. Then Dueling Deep Q-Networks and Soft Actor-Critic are developed, and a state-of-the-art model-based RL framework is employed to develop their model-based versions. The comparison results showed that all RL controllers outperform the baseline control in terms of indoor temperature and operation costs. Model-based RL can achieve a control performance as good as model-free RL with a shorter training time based on its high sample efficiency. Moreover, due to massive and quickly generated data, model-based RL can accelerate the learning of RL agents, though the model is inaccurate at the early training stage. This study would provide some insights into the RL control selection and improvements in HVAC systems.

Nomenclature

Variables, parameters, and indices

A	The total floor area (m^2)
a, b, c	Weight factors
$cost$	Total cost (EUR)
$reward$	The value of reward
$tdis$	Total discomfort (°C h)
P	Probability

* Corresponding author.
E-mail addresses: wangdan9264@bjut.edu.cn, wangdan9264@foxmail.com (D. Wang).

https://doi.org/10.1016/j.jobe.2023.106852
Received 15 March 2023; Received in revised form 30 April 2023; Accepted 14 May 2023
Available online 19 May 2023

Price	Electricity price (EUR/kWh)
Q_{rad}	Solar irradiation (W/m^2)
s_z	Deviation from the bound temperature (°C)
T_{in}	Zone temperature (°C)
T_{low}	Lower bound of zone temperature (°C)
T_{out}	Outdoor dry bulb temperature (°C)
T_{up}	Upper bound of zone temperature (°C)
tdis	Total discomfort

Greek letters

γ	Discount factor
σ	Predictive standard deviation

Abbreviation

BOPTEST	Building optimization testing framework
DDPG	Deep deterministic policy gradient
DDQN	Dueling deep Q network
DNN	Deep neural network
DQN	Deep Q network
HVAC	Heating, ventilation and air conditioning
KPI	Key performance indicator
MBDDQN	Model-based DDQN
MBSAC	Model-based SAC
MBPO	Model-based policy optimization
MDP	Markov decision process
MPC	Model predictive control
MSE	Mean square error
PI	Proportional integral
PPO	Proximal policy optimization
RBC	Rule-based control
RL	Reinforcement learning
SAC	Soft actor-critic

1. Introduction

Buildings have been playing a major role in the rapid growth of energy use around the world [1,2], contributing 40% of global primary energy and 30% of CO_2 emissions [3]. Heating Ventilation and Air-Conditioning (HVAC) systems are the main energy use among all building services, which accounts for more than 50% of the building energy consumption [4]. Therefore, it is of great significance to keep zone thermal comfort while lowering the HVAC systems' energy demands.

Meanwhile, HVAC systems are also challenging to control due to the varying weather conditions, complex building and system structures, stochastic occupant behavior, and other uncertainties [5]. The common control approaches to regulate HVAC systems are based on predefined rules (Rule-based control, RBC) or physical building models (Model predictive control, MPC). RBC can be developed and applied in various building systems easily based on expert experience, which contains some static thresholds on indoor temperature or simple control loops. However, RBC is not optimal in most conditions because it is fixed regardless of operating conditions and not customized for a specific building [6]. Therefore, MPC is developed to overcome these drawbacks by making use of building models to predict disturbance and its impact on the indoor environment and energy consumption [7]. MPC performs well and robustly in many cases, however, it is also limited. Due to the complexity of indoor thermal dynamics and various influencing factors, model development is usually labor-intensive and time-consuming [8]. As a result, despite the advantages of RBC and MPC, they still have limitations and shortages in control performance and wide applications.

With the development of artificial intelligence and big data, reinforcement learning (RL) has been viewed as a promising solution in recent years [9,10]. Unlike RBC which depends on a set of rules predefined by experts, RL may keep learning and updating the control strategy during operation [11]. Additionally, RL can learn directly from operation data and does not need to develop complicated models for building and energy systems like MPC [12]. These merits facilitate RL's application in complicated and dynamic HVAC systems by ensuring its performance and generality.

1.1. Literature review

In HVAC control fields, great efforts have been paid to model-free RL due to its simplicity of usage. The model-free RL can be roughly divided into three types: value-based (act by choosing the best action in the state), policy-based (directly learn the stochastic policy function that maps state to action), and their combination. Azuatalam et al. [13] employed proximal policy optimization (PPO)

to optimize a whole-building HVAC system for demand response goals. The results showed that a maximum weekly energy reduction of up to 22% can be achieved compared to a handcrafted baseline controller. Du et al. [14] applied deep deterministic policy gradient (DDPG) to generate an optimal control strategy for a multi-zone residential HVAC system. Compared to deep Q network (DQN) and RBC, the DDPG reduced the energy consumption cost by 15% and the comfort violation by 79% and 98%, respectively. Biemann [15] carried out experiments to evaluate four actor-critic algorithms in a simulated data center. Compared to the model-based controller implemented into EnergyPlus, all applied algorithms can reduce energy consumption by at least 10% without jeopardizing occupant thermal comfort. Li et al. [16] proposed a model-free RL control strategy to adjust the temperature setpoints for the thermal storage air conditioning systems. The results showed that when compared to a non-thermal storage air conditioning system with a constant set-point, the RL agent saved 9.17% of utility costs. Yu et al. [17] proposed an energy-efficient personalized thermal comfort control algorithm based on attention-based multi-agent deep RL for office buildings. The proposed technique may simultaneously reduce average thermal comfort deviation by 64%–72% and energy usage by 0.7%–4.18% when compared to baselines. Additionally, a novel HVAC control method combining active building environment change detection and DQN is proposed by Deng et al. [18]. Their control strategy obtains stability against disturbance and generalization to an unseen building environment and reaches 13% in energy-saving and 9% improvements in thermal comfort. However, these model-free RL approaches often require sufficient explorations to converge to a stable and good policy from 70 episodes to 1000 episodes [13–18] and thus it is limited for application due to the sample inefficient [19].

To address the problem of sample inefficiency, improved model-free RL techniques have focused on creating various approximate schemes to reduce the complexity and dimension of the state-action space. Xiong et al. [20] proposed an approach to refine action space applied on DQN to control the building cooling water system. The results show that refining the action space can accelerate the convergence speed for one episode and achieve the average COP improvement rate of about 6.4%. Homod et al. [21] employed deep clustering-based methods to enhance the learning efficiency and stability of RL agents with extremely large state–action spaces. Sun et al. [22] developed an event-based Q-learning method within the Lagrangian relaxation framework to achieve energy savings in HVAC systems. The computational requirements can thus decrease significantly due to the reduction of events policy space size. Li et al. [23] proposed a multi-grid method of Q-learning which adopted a coarse model to fast converge to a good policy in early stages and a fine model to further improve the optimization result. However, the performance of these approaches heavily depends on experts' ability to classify a large state-action space and the categorization outcomes.

Model-free RL and its improved techniques demonstrate great effectiveness in HVAC systems. However, as a trial-and-error learning method, the control performance of pure model-free RL heavily relies on the data it gathers, and the control performance of improved model-free RL requires expert intervention to classify or cluster data. As a result, these techniques have a rather limited application. Model-based RL approaches are considered a potential solution for overcoming the disadvantages of model-free RL algorithms, which can gather experience from models and accelerate the learning process [24].

Model-based RL algorithms usually can be divided into two categories [25]: *Model-based RL with a learned model* (Dyna-style [26]), and *Model-based RL with a known model* (AlphaZero [27]). For HVAC systems, Zhang et al. [28] proposed a model-based RL approach for a two-room data center via neural network-based model approximation. Compared with the model-free RL approach (PPO), their approach improves the sample efficiency by 10x. Dawood et al. [29] presented model-based RL techniques to control the indoor air temperature and CO_2 concentration level with minimization of the system energy consumption. The results showed that model-based RL reduced energy consumption while keeping the indoor comfort levels within the desired ranges simultaneously. Zhang et al. [30] developed a novel model-based optimal control method for HVAC supervisory-level control based on a deep RL framework. Their method employed an EnergyPlus model and A3C algorithm. The results indicate that the control method can save about 15% heating energy while maintaining acceptable indoor thermal comfort. Arroyo et al. [31] proposed a novel algorithm called reinforced predictive control (RL-MPC) that merged the merits of two methods, namely state estimation, dynamic optimization, and learning. The results demonstrated that the RL-MPC algorithm can meet constraints and provide similar performance to MPC while enabling continuous learning and the possibility to deal with uncertain environments.

Compared to model-free RL, model-based RL algorithms show great potential for improving sample efficiency and satisfying the constraints of room temperature. However, the performance of model-based RL depends on the accurate and robust representation of system dynamics, and this topic is not covered in existing studies. Additionally, existing studies mostly employ pre-built models or supervised approaches to train system models. As a result, this would need extra data and labor-intensive modeling like MPC and thus decreases its generality.

In summary, two main obstacles to the application of RL can be concluded from the overall literature review. *First*, there is a lack of a general and accurate modeling framework for model-based RL in HVAC systems. The majority of existing studies employed model-based RL algorithms with known models, which are developed with measured and simulated data before application. It shows great performances in terms of control stability and safety, but it limits the scalability and generalization of model-based RL. Furthermore, the setting parameters of pre-developed models are usually fixed after calibration and cannot adapt to uncertainties during operation, such as aging equipment, envelope renovation, human behavior, etc. *Second*, a thorough comparison between model-free and model-based RL is required to show performance gaps and reveal the potential improvement directions for RL control. Existing studies mostly compared their model-based RL methods with a single model-free algorithm and did not explore the performance gaps in different control scenarios. Additionally, the impact of model accuracy on the training process and control performance has also not been discussed in detail.

1.2. Objectives and contributions

In response to the above concerns and challenges, this paper conducts an in-depth comparison between model-free and model-

based RL control to quantify performance gaps under different control scenarios. The main contributions of this paper can be summarized as follows:

- Two kinds of model-free RL methods, Dueling Deep Q-Networks (DDQN) and Soft Actor-Critic (SAC), are developed to solve the discrete and continuous control problem. A state-of-the-art model-based RL framework with a learned model is introduced to develop their model-based versions.
- The performance gaps between model-free and model-based RL were demonstrated in a typical HVAC system of an open-source virtual environment. The results are discussed in terms of sample efficiency, control performance, and computational time.
- The effectiveness of model-based RL is investigated in relation to the impacts of model accuracy.

The remainder of the paper is organized as follows. Section 2 presents an overview of the HVAC system and implementation details of model-free and model-based RL algorithms. Section 3 details the results for the model-free and model-based RL, and the control behavior will be discussed. The impact of model accuracy on control performance and the limitations of this research are then discussed in Section 4. Finally, the main conclusions of this paper are summarized in Section 5.

2. Methodology

2.1. Test environment

2.1.1. Test case

As an open-source testbed for building control performance benchmarking, building optimization testing framework (BOPTEST) [32] is selected as the virtual test environment in this paper. BOPTEST is a standardized simulation environment to ensure fair evaluations of different algorithms. The functionality is enabled through API to select a test scenario, advance a simulation, and get data like measurements, forecasts, or key performance indicators (KPIs) at each control step. The framework is freely accessible at https://github.com/ibpsa/project1-boptest.

The test case "*BESTEST Hydronic Heat Pump*" is of interest in this paper. This model represents a simplified residential dwelling for a 5-member family, modeled as a single thermal zone, located in Brussels, Belgium. The building has a rectangular floor plan of 12 m by 16 m and contains 24 m^2 of windows on the south facade. The thermo-physical properties of the test model are listed in Appendix A, including exterior walls, floor, and roof. An air-to-water modulating heat pump of 15 kW nominal heating capacity extracts energy from the ambient air to heat up the floor heating system. The occupancy schedule operates before 7:00 a.m. and after 8:00 p.m. on weekdays and full-time on weekends, which represents a typical residential building schedule. The room temperature setpoint varies between 15 °C and 30 °C in an unoccupied condition to keep the air conditioner from turning on, while it is set between 21 °C and 24 °C in an occupied condition to provide good indoor thermal comfort. The detailed information of the building is summarized in Table 1.

2.1.2. Baseline control

To ensure comfort inside the building zone, a baseline proportional-integral (PI) controller is employed to determine the heat pump's compressor frequency based on the difference between the actual indoor temperature and its setpoint, as depicted in Fig. 1. The setpoint is calculated for baseline control as the heating comfort setpoint plus an offset that varies based on the occupancy schedule: the offset is set to only 0.2 °C during occupied periods and is intended to avoid discomfort from slight oscillations around the setpoint; the offset is set to 5.5 °C during unoccupied periods and is intended to compensate for the large temperature setback used during these periods. Due to the high thermal inertia of the floor heating system, the latter offset prevents the need of abrupt changes in the indoor temperature which would consequently cause discomfort. All other equipment, including a fan for the heat pump evaporator circuit and a pump for the floor heating system, is turned on when the heat pump is operating and turned off otherwise.

2.1.3. Performance metrics

A KPI calculator is embedded in the BOPTEST. It calculates the key performance metrics as a post-process after a test is complete. As our objectives are to reduce the operation cost while maintaining the indoor temperature within a reasonable range. Two main KPIs are employed in this paper, namely *tdis* and *cost*, which calculate the thermal discomfort and operation cost in a fair way respectively, as

Table 1
Information of the test building.

Construction	Floor area	192 m^2
	Height	2.7 m
	Window area	24 m^2
	Materials	See Appendix A
Occupancy	Weekdays	20:00–7:00
	Weekends	0:00–24:00
HVAC	Heating source	15 kW air-to-water modulating heat pump
	End-equipment	Floor heating system
	Heating setpoint	Occupied: 21–24 °C Unoccupied: 15–30 °C

Table 2
State variables of the reinforcement learning algorithms.

No.	Variable	Symbols	Units/Value
1	Zone temperature	T_{in}	°C
2	Outdoor dry bulb temperature	T_{out}	°C
3	Solar irradiation	Q_{rad}	W/m^2
4	Upper bound of zone temperature	T_{up}	°C
5	Lower bound of zone temperature	T_{low}	°C
6	Electricity price	*Price*	EUR/kWh

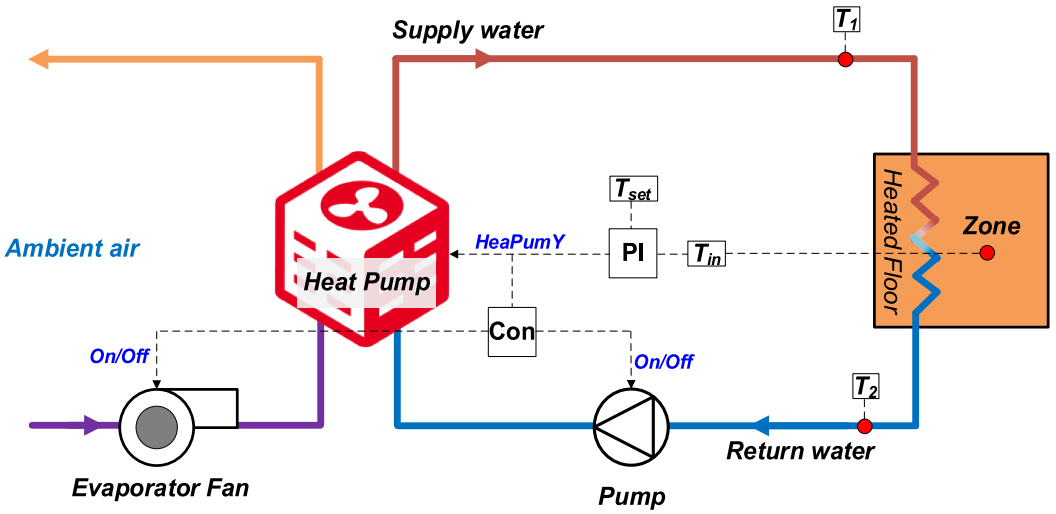

Fig. 1. A schematic view of the building system.

shown in Eqs. (1) and (2).

$$tdis = \frac{\sum_{z}^{N} \int_{t_0}^{t_f} \| s_z(t) \| \, dt}{N} \tag{1}$$

where *tdis* is the total discomfort between the initial time t_0 and the final time t_f; z is the zone index out of N zones in the building; s_z(t) is the deviation (slack) from the lower and upper bound temperatures established in zone z, with a unit of K; z and N are both set to 1 in this paper.

$$cost = \frac{\sum_{i \in \xi} \int_{t_0}^{t_f} p_i^{\tau}(t) P_i(t) dt}{A} \tag{2}$$

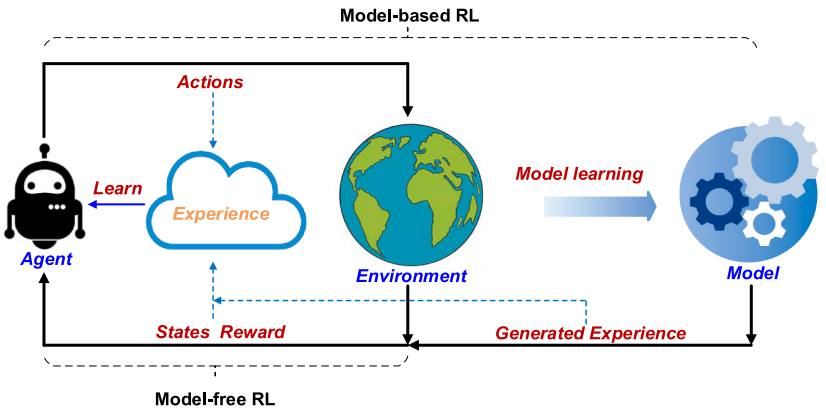

Fig. 2. The basic principle of different RL algorithms.

where *cost* is the total cost between the initial time t_0 and the final time t_f with a tariff τ; z is the price profile of equipment i with a tariff τ and units of EUR/kWh; A is the total floor area of the building, with a unit of m^2.

2.2. Reinforcement learning implementation

RL is a type of machine learning that interacts with the dynamic environment and learns to get the optimal action sequence. Generally, RL problems can be formalized as a Markov decision process (MDP) [33], represented by a quadruple: $M = <S, A, R, P>$, where S represents the state space; A refers to the action space; R represents the reward received after transitioning from state s_t to state s_{t+1}; P represents the probability that action a_t in state s_t will lead to state s_{t+1}.

As shown in Fig. 2, RL can be mainly divided into two categories: model-free RL and model-based RL. For model-free RL, at each time step t, the agent takes an action a_t according to the state s_t and moves to the next state s_{t+1}, and then obtains a reward r_t. The agent seeks to learn the consequences of their actions through direct experience by interacting with the real environment [34]. Model-based RL, on the other hand, tries to learn its environment and develop a model for it. The model developed for model-based RL may be known before the experiments or learned through the experiments. In this paper, we mainly consider the latter for its simplicity and flexibility [35]. As shown in Fig. 2, the agent of model-based RL not only learns from the direct experience through the real environment but also learns from the generated experience from the model simulation. As a result, model-based RL may offer distinct advantages of being sample efficient and accelerating the agent's learning speed.

2.2.1. Algorithms

In this paper, two different model-free RL algorithms and their model-based algorithms are developed to study the control performance of HVAC systems, namely Dueling Deep Q-Networks (DDQN), Soft Actor-Critic (SAC), model-based DDQN (MBDDQN) and model-based SAC (MBSAC). DDQN is a value-based RL technique, which is usually used to solve the discrete control problem. SAC is a value-policy-based RL technique, which can be used to solve the continuous control problem. The choice of these two types of RL algorithms allows for a thorough examination of the system performance in discrete and continuous control scenarios.

(1) Model-free reinforcement learning

DDQN is an improved version of the traditional Q-learning algorithm, which uses two deep neural networks (DNN) to overcome the issue of dimensional explosion and Q-value overestimation [36]. The agent tries to find the action that leads to the highest Q-value in each state. The Q-value, also called the state-action value, is defined as the expected reward achieved for taking a specific action a at the given state s. As shown in Fig. 3, unlike the classical DQN which only produces a single output Q value, DDQN outputs the predictive state value function V and the prediction relative advantage function A respectively. As shown in Eq. (3), the Q-value is then calculated by adding V and A, and overestimation can thus be handled [37]. However, Eq. (3) cannot be directly used because it is unidentifiable. A common practice to address this is to force the advantage estimator to be zero at the best action for that state, as shown in Eq. (4). In this paper, we used the averaging operator in Eq. (4), which is suggested to replace the maximum operator for good stability in previous studies [38].

$$Q(s,a) = V(s) + A(s,a) \tag{3}$$

$$Q(s,a) = V(s) + A(s,a) - \frac{1}{|\mathscr{A}|}\sum_{a' \in \mathscr{A}} A(s,a') \tag{4}$$

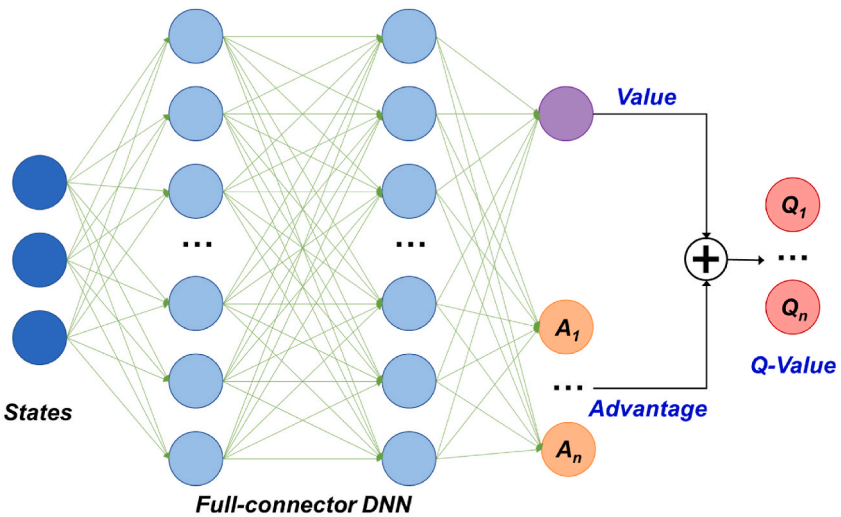

Fig. 3. The structure of DNN in DDQN.

SAC is an RL algorithm that optimizes a stochastic policy in an off-policy way. The actor-critic structure is employed in SAC, which contains an actor to select actions and a critic to evaluate the actions made by the actor. As shown in Fig. 4, it employs two different DNNs for approximating the action-value function and state-value function. The actor maps the current state based on the action that it estimates to be optimal, while the critic evaluates the action by calculating the value function. SAC maximizes the information entropy of state apart from the conventional cumulative rewards. As shown in Eq. (5), SAC prefers stochastic policies, and it does this by modifying the objective function with an additional term of the expected entropy ($\mathscr{H}$) of the policy.

$$J(\pi) = \sum_{t} \mathbb{E}_{(s_t,a_t)\rho_\pi}[r(s_t,a_t) + \alpha \mathscr{H}(\pi(\bullet|s_t))] \tag{5}$$

Here α and $\mathscr{H}(\pi(\bullet|s_t))$ is the trade-off between entropy and reward. The advantage of entropy maximization is that it can lead to policies that can explore more and prevent the policy from prematurely converging to an improper local optimum. This objective can be extended to infinite horizon problems by introducing the discount factor r and as a derivation of this objective is more involved, the reader is referred to the plenary text of [39].

(2) Model-based reinforcement learning

The model-based policy optimization framework (MBPO) which is a state-of-the-art model-based RL algorithm proposed by Janner et al. [40] is employed in this paper. MBPO is a Dyna-style algorithm that unifies planning and learning into a single framework. A model-based approach involves using a model of the environment to forecast the outcomes of states and actions, while a policy optimization approach finds a policy that maximizes the expected reward. By combining these two approaches, MBPO can provide a more efficient way of solving control problems [41] and shows good control performances.

The schematic diagram of MBPO is shown in Fig. 5. At each timestep, the agent interacts with the environment and stores the real experience. Then it samples real experience to train an environment model to generate hypothetical simulated experience. Finally, the agent updates the policy parameters by using a combination of inputs from real and simulated experiences. Therefore, it can be more efficient than learning solely from real experience, because the simulated experiences can be generated quickly and allow the agent to explore a wider range of possible states and actions [42].

The model error, known as model bias, tends to cripple the performance of model-based RL. Therefore, the model ensemble method is adopted in this paper. It shows to be effective in reducing prediction error from a single model [43]. In this paper, the ensemble model consists of five DNNs. As shown in Fig. 6, at each time step, the ensemble model samples N states s that the agent has been experienced. Then each sampled state is employed as the initial state in a model rollout and to use the agent to generate actions a. Given the s and a, DNN is randomly chosen to output a Gaussian distribution of the next states s' and reward r, as depicted in Fig. 7. The agent can then be used to generate actions a' based on s' and repeat the above process for k times. Finally, $N \times k$ pieces of hypothetical experience are generated by the ensemble model. Each DNN in the ensemble model is trained with different initializations and bootstrapped samples of the real environment data via maximum likelihood. The loss of the model employed the mean square error (*MSE*), as shown in Eq. (6). The pseudo-code of MBPO is shown in Algorithm 1.

$$MSE(\theta) = \frac{1}{N}\sum_{1}^{N}\left[\left(\mu_\theta^i - e_i\right)^2 + \sigma_i\right] \tag{6}$$

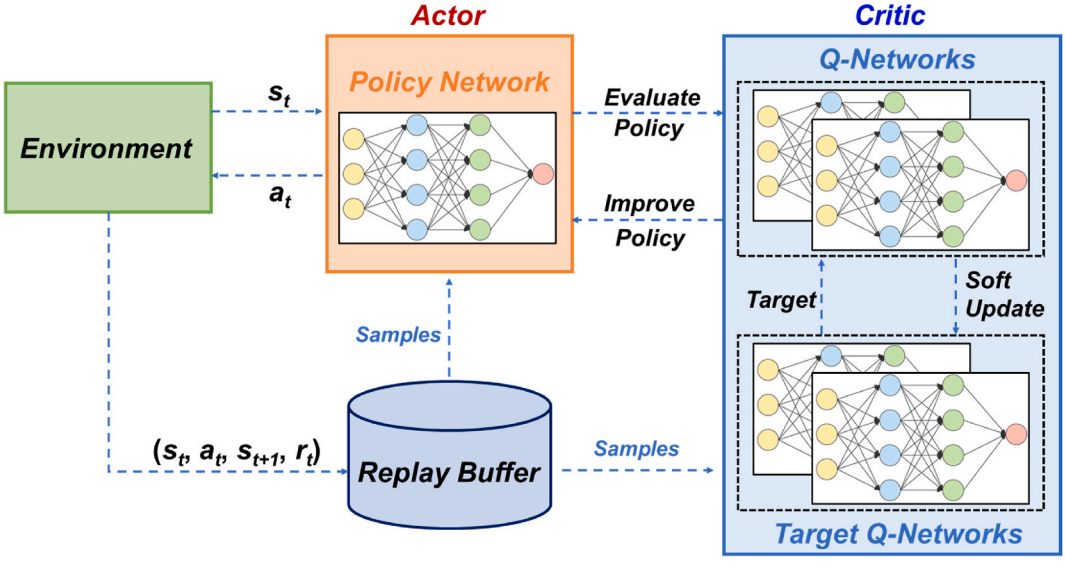

Fig. 4. The schematic diagram of SAC.

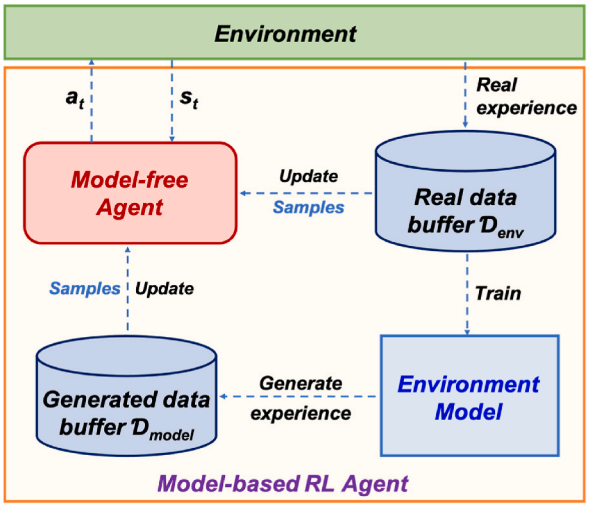

Fig. 5. The framework of MBPO.

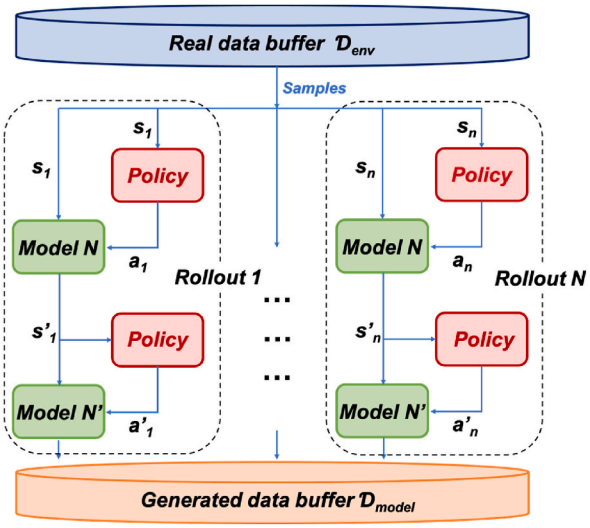

Fig. 6. The process of model rollout.

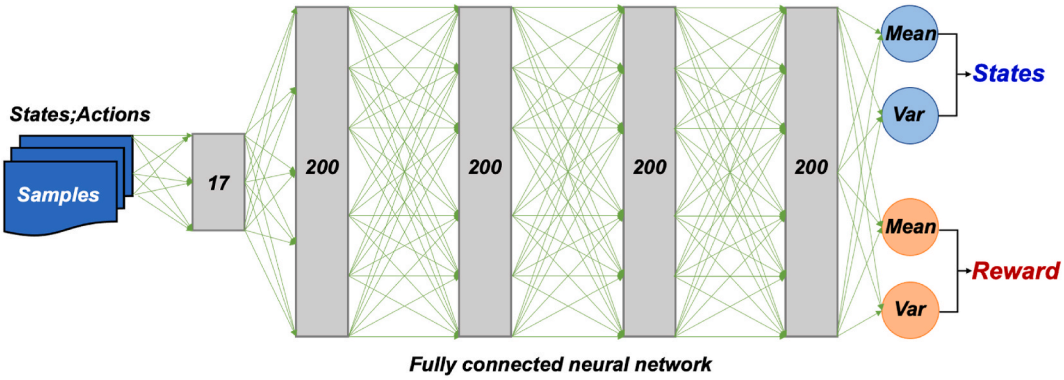

Fig. 7. The DNN of the environment model.

where θ is the parameters of DNN; N is the size of samples; μ_θ is the predictive mean value by DNN; e is the ground truth of state and reward values; σ is the predictive standard deviation.

Algorithm 1. Model-based Policy Optimization (MBPO)

Algorithm 1 Model-based Policy Optimization (MBPO)

1. Initialize policy π_φ, model p_θ, environment replay buffer D_{env}, model replay buffer D_{model}
2. **for** N episodes **do**
3. **for** E steps **do**
4. Take an action in the environment according to π_φ; add to D_{env}
5. **for** per M steps **do**
6. Train p_θ based on D_{env}
7. **end for**
8. **for** L model rollouts **do**
9. Sample state s_t form D_{env}
10. Perform k-step model rollout starting from s_t; add to D_{model}
11. **end for**
12. **for** G updates **do**
13. Update π_φ parameters using the combination of model data and environment data
14. **end for**
15. **end for**

2.2.2. Design of state

Proper variables should be used to create the state space to capture the full information of the system and provide the agent with enough knowledge to make informed decisions. To prevent overfitting and the dimensionality curse, it is also suggested to remove redundant and irrelevant data. For the test case, the state space should reflect the operation cost of the building system and zone thermal comfort. Therefore, as shown in Table 2, the zone temperature, upper and lower bounds of zone temperature, and electricity price are considered. As the ambient environment would also affect the indoor temperature, two major influence parameters, the outdoor dry bulb temperature, and solar irradiation, are introduced into the state space.

To reduce operation costs and maintain the zone temperature for a long period, the control signal made in time t should consider the future state. Therefore, we extend the dimension of the state variable: T_{out}, Q_{rad}, T_{up}, T_{low}, $Price$, by incorporating future prediction within a time window (set to 2 h in this paper). Furthermore, the state variables are all normalized within a range of [0, 1] to accelerate training.

2.2.3. Design of action

The action is the control variable which can be determined by the agent to change the state of the environment. In this paper, the heat pump modulating signal u is regarded as the action. The range of action space, the heat pump modulating signal, is [0, 1]. In SAC, the action space is continuous from 0 to 1. In DDQN, the action space is partitioned into 10 uniform intervals, $u = \{0, 0.1, 0.2, \ldots, 1\}$, because it works in the discrete action space.

2.2.4. Design of reward function

A well-designed reward function can accelerate the convergence of the agent and guide the agent to take proper actions under different situations. Since our objective is to minimize thermal discomfort and operation cost, the reward function is designed as a weighted sum of these two goals. The two KPIs described in Section 2.1.3 are employed to construct the reward function. The mathematical representation of the objective function is shown in Eqs. (7) and (8).

$$r(t) = a \bullet tdis + b \bullet cost \tag{7}$$

$$reward(t) = c \bullet [r(t-1) - r(t)] \tag{8}$$

where $r(t)$ represents the performance of operation cost and zone temperature at t time; $reward(t)$ represents the value of reward given to the agent at t time; a, b, c are weight factors, which are set to 10, 1, 0.05, respectively.

2.2.5. Hyperparameters

In this paper, we manually tuned the hyperparameters of RL algorithms since BOPTEST does not support parallel computing. The hyperparameters of four algorithms are listed in Table 3. It should be noted that the hyperparameters of model-based algorithms are consistent with their model-free algorithms, respectively. This allows to compare the performance gaps between model-based RL and model-free RL in a fair way. Additionally, the rollout size of model-based RL algorithms is set to 90% of batch size. Compared to model-free RL, model-based RL algorithms only employ 10% real experience and 90% generated experience to update the RL agent, which highly improves the sample efficiency.

3. Results

3.1. Convergence

In this paper, all tests are run on an Intel processor with 16 GB RAM to evaluate the performance of different RL algorithms. RL algorithms are all implemented in Python 3.7 and OpenAI Gym [44] is used to interface with the BOPTEST simulation environment.

Fig. 8 depicts the evolution of cumulative return for each episode of different RL algorithms. A clear rising trend in the episodic reward can be seen with more training episodes. It demonstrates that RL agents can successfully learn through interacting with the environment. Additionally, MBDDQN and MBSAC both exhibit significantly higher episodic rewards and lower reward deviation between different trials than model-free algorithms. This indicates that model-based RL algorithms can achieve more stable and better training outcomes, compared to their original model-free RL algorithms.

As shown in Fig. 8 a), MBDDQN approaches convergence at around the 23rd episode, while DDQN reaches convergence at around the 37th episode, which saves nearly 14 episodes (4700 timesteps). The reward of MBDDQN in the 21st episode matches that of DDQN in the 37th episode. Fig. 8 b) suggests that while SAC approaches convergence around the 47th episode, MBSAC converges at the 35th episode, which saves nearly 12 episodes (4000 timesteps). The performance of MBSAC at the 31st episode is identical to that of SAC at the convergence. Therefore, model-based RL can achieve a faster convergence speed and a higher reward at the convergence when compared to model-free RL.

Comparing Fig. 8 a) and b), the model-based RL approach provides different improvements for model-free RL algorithms under different control scenarios. During the convergence period (40–50 episodes), MBDDQN achieves approximately the same operation cost reductions as DDQN and a lower thermal deviation, while MBSAC reaches the same thermal deviation as SAC and saves more operation costs. As a result, at convergence, MBDDQN yields a reward that is 2.10% higher than DDQN, while MBSAC yields a reward that is 6.53% higher than SAC. This illustrates that the continuous control scenario may benefit more from the model-based RL than the discrete control scenario. Additionally, DDQN and MBDDQN converge more quickly than SAC and MBSAC which demonstrates that continuous control learning is more challenging and takes longer time to converge than discrete control.

3.2. Control performance

To describe the performance of the control during the test period, the highest reward trial out of five random seeds is selected. Fig. 9 displays KPIs for different RL algorithms at the 50th episode. All RL control methods outperform the baseline control in terms of thermal comfort, operation cost and energy consumption, although they add various amounts of computational burden.

In the discrete control scenario, MBDDQN achieves the same cost savings as DDQN while reducing the zone thermal deviation by 42.86%. It should be noted that MBDDQN achieves the same cost savings as DDQN and higher energy consumption, leading to greater flexibility in the HVAC system. For the continuous control scenario, MBSAC achieves a lower operation cost and energy consumption than SAC with a reduction of 10.98% and 11.78% respectively. However, MBSAC has a higher zone thermal deviation than SAC with an increase of 61.60%. In terms of computational time, MBDDQN needs 81.74% more time than DDQN, and MBSAC takes 88.14% more time than SAC. This is due to the increased computational load of model training and hypothetical experience simulating in model-based RL.

Figs. 10 and 11 depict the detailed indoor temperature variations and control signals of DDQN and MBDDQN during the test period. The test period lasts for two weeks. As seen, the baseline PI controller is well-tuned and has a good control performance, because it can keep the indoor temperature at the lower bound of the comfortable range. Even when compared to this high-standard baseline, DDQN

Table 3
Hyperparameters of different RL algorithms.

Hyperparameter	DDQN	SAC	MBDDQN	MBSAC
Learning rate	0.003	0.0005	0.003	0.0005
Learning rate (Critic)	–	0.002	–	0.002
Target entropy	–	−1	–	−1
Discount factor	0.95	0.99	0.95	0.99
Exploration rate	0.03	–	0.03	–
Target update rate	10	–	10	–
DNN	64 × 64	64 × 64	64 × 64	64 × 64
Batch size	1024	1024	1024	1024
Rollout size	–	–	1024 × 0.9	1024 × 0.9
Rollout length	–	–	1	1
G updates	–	–	5	5
Model DNN	–	–	200 × 200 × 200 × 200	200 × 200 × 200 × 200

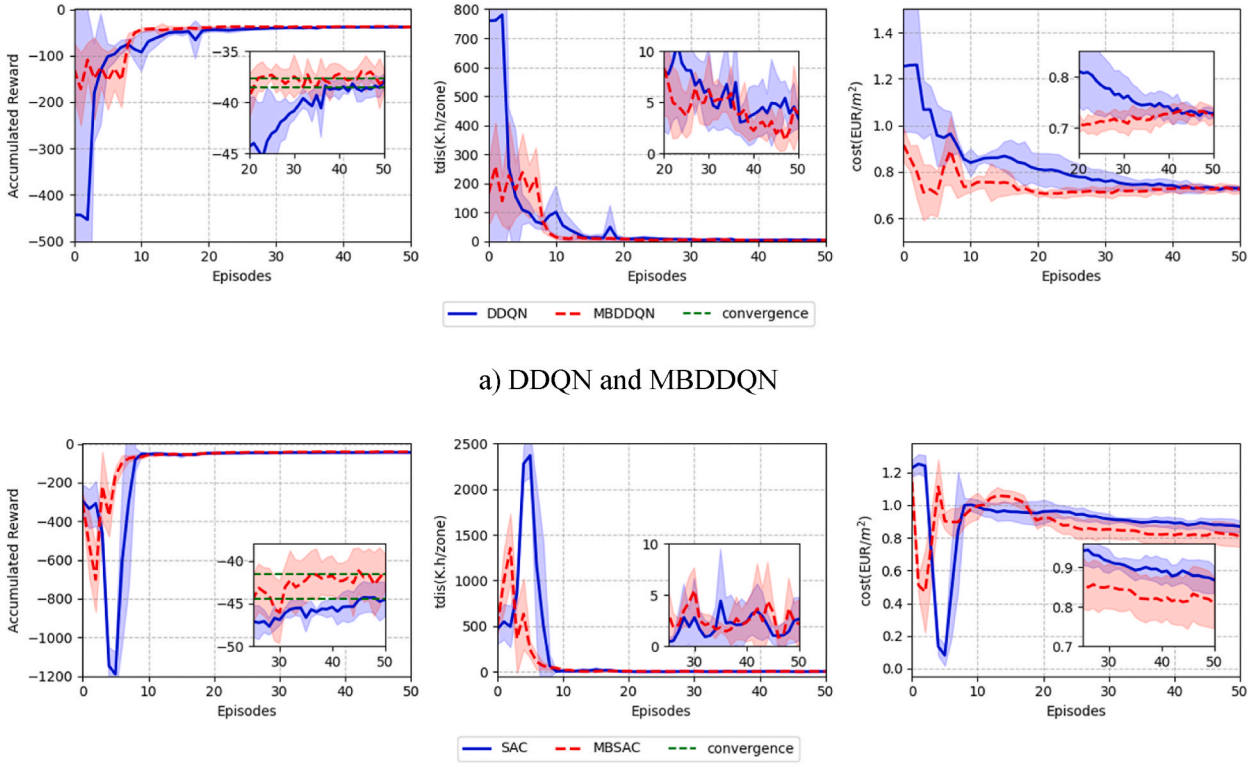

Fig. 8. Training curves of DDQN, MBDDQN, SAC, and MBSAC. (The solid lines indicate the mean value and shaded areas indicate the standard deviation of five trails over different random seeds).

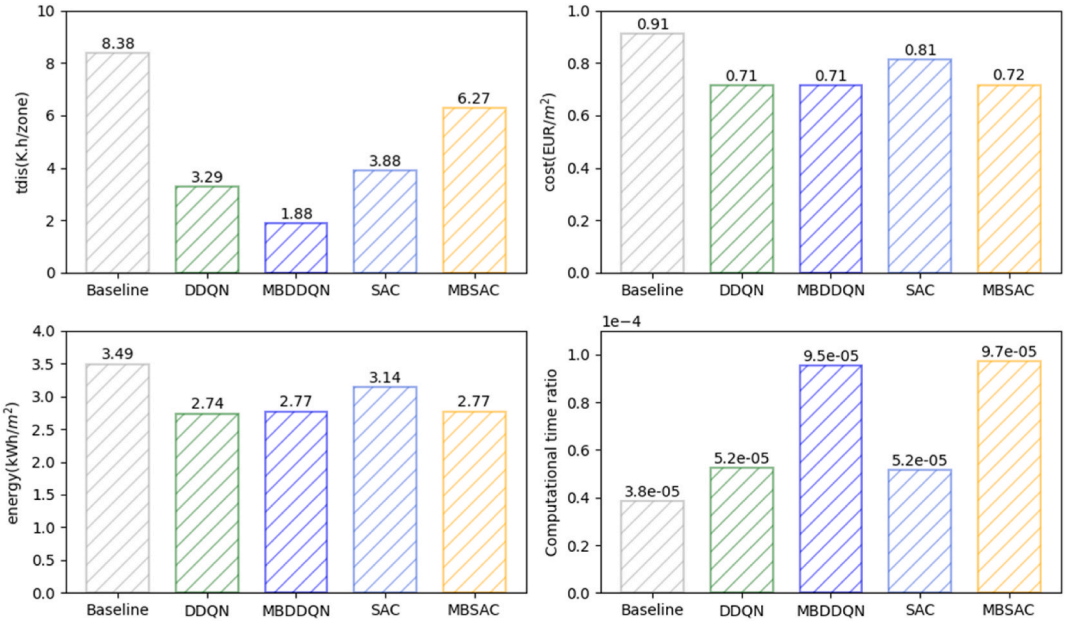

Fig. 9. KPI values of baseline and different RL algorithms at the convergence.

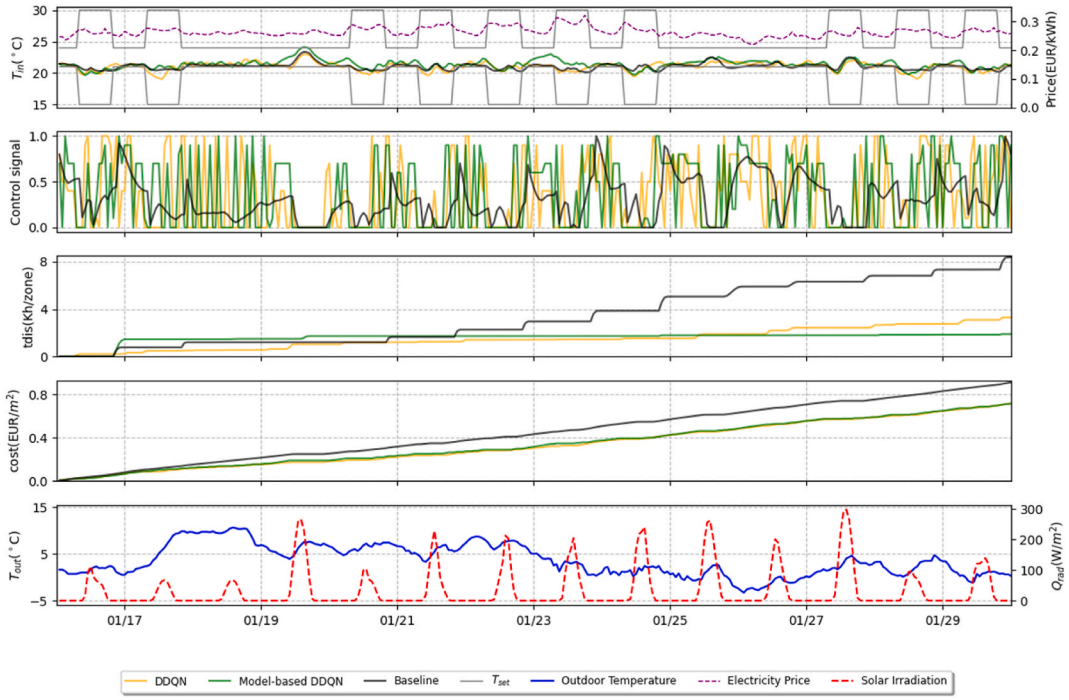

Fig. 10. Control performances of DDQN and MBDDQN under the test period.

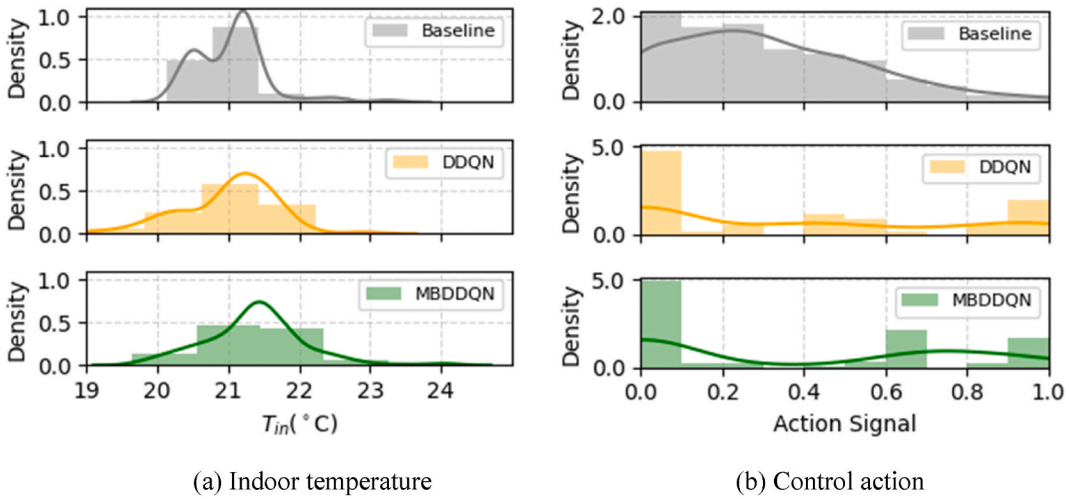

Fig. 11. Distributions of indoor temperature and control signal of DDQN and MBDDQN.

and MBDDQN both perform better in terms of indoor thermal comfort and energy consumption. Additionally, MBDDQN exhibits a higher indoor temperature compared with baseline control and DDQN. As shown in Fig. 11 a), the indoor temperature of MBDDQN mainly varies above 21 °C, while that of DDQN mainly varies between 20 °C and 22 °C.

As seen in Figs. 10 and 11 b), three controls have different control behaviors during the test period. We can see DDQN and MBDDQN regulate the system at significantly higher frequencies than baseline control, while baseline control signals shift more gradually and have a more stable distribution from 0 to 1. The control signals of DDQN and MBDDQN are both concentrated in 0 or 1, however, MBDDQN distributes more control signals in the middle range of [0.6, 0.7] than DDQN does. Moreover, notably on January 23rd, MBDDQN can pre-heating the room by taking advantage of electricity price fluctuation, which decreases utility costs while maintaining thermal comfort. As a result, compared with baseline control and DDQN, MBDDQN achieves a lower zone thermal deviation while reducing operation costs.

Figs. 12 and 13 depict the detailed indoor temperature and control signals of SAC and MBSAC during the test period. Due to their higher indoor temperatures than the baseline control during the occupied time, SAC and MBSAC experience less thermal discomfort.

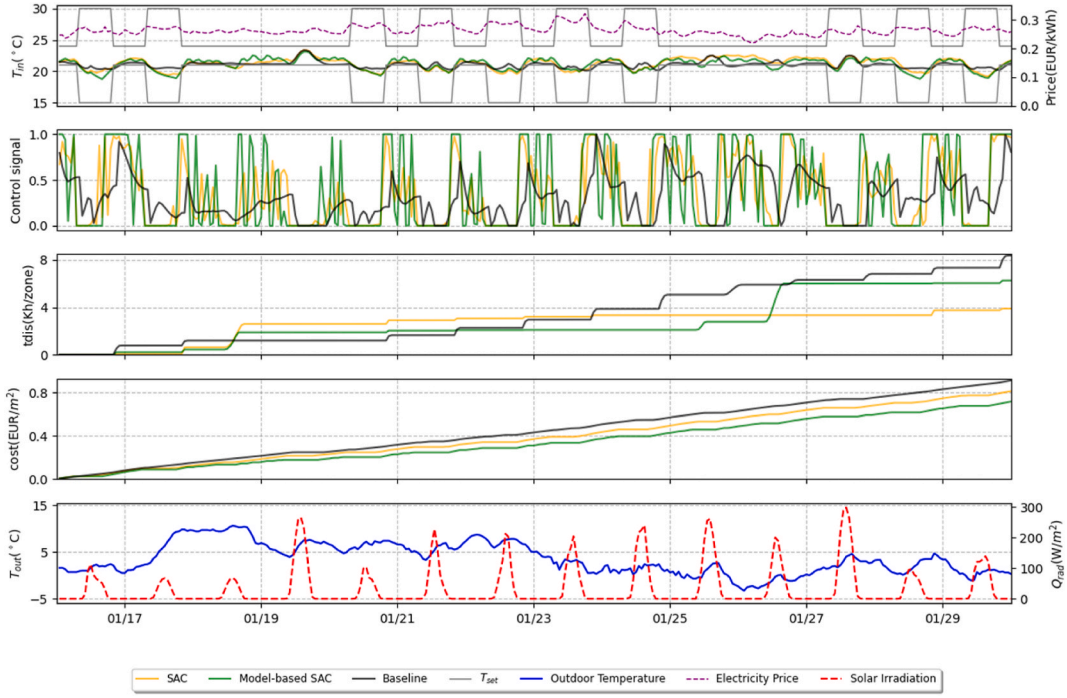

Fig. 12. Control performances of SAC and MBSAC under the test period.

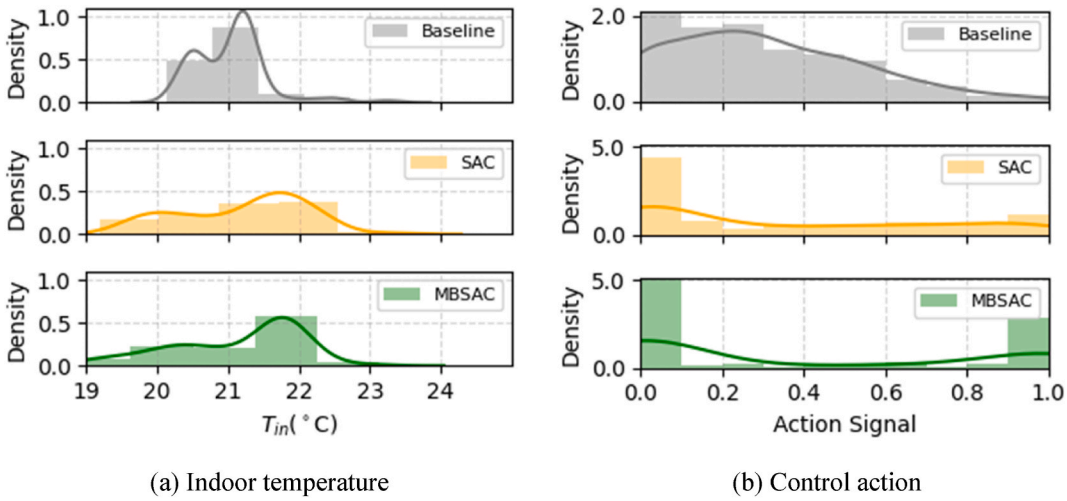

Fig. 13. Distributions of indoor temperature and control signal of SAC and MBSAC.

Additionally, Fig. 13 a) demonstrates that the indoor temperature of MBSAC is mostly concentrated at 22 °C, whereas the indoor temperature of the SAC is more evenly distributed between [19 °C, 22 °C]. However, during some unoccupied time, the indoor temperature of MBSAC is lower than SAC, which makes MBSAC unable to respond in time between occupied and unoccupied periods as SAC can. On January 26th, MBSAC is unable to control the indoor temperature within the comfort range, resulting in MBSAC's *tdis* exceeding that of SAC. These are reasons that MBSAC achieves a higher indoor thermal deviation than SAC.

As depicted in Fig. 13 b), the control signals of MBSAC generally alternate between 0 and 1, while the control signals of SAC mainly alternate between 0 and [0.1, 1]. MBSAC tends to turn off the heat pump during the unoccupied time, whereas SAC would permit the heat pump to run at a low frequency. Therefore, SAC can achieve a higher indoor temperature during the unoccupied time and a smoother control signal fluctuation through the test period. At the same time, however, this has led to increased operating costs for SAC compared to MBSAC.

4. Discussion and future research

4.1. Discussions

A crucial concern of this study is how the model accuracy of model-based RL algorithms fluctuates and influences the training process. To answer this question, the *MSE* of the model during the training process is shown in Fig. 14. As seen, the models are inaccurate before the 10th episode for MBDDQN and MBSAC. Even though, when comparing Fig. 8 with Fig. 14, model-based RL still achieves a faster learning speed than model-free RL based on sufficient model-generated data at the early training stage. However, the rewards of MBDDQN and MBSAC fluctuate around a high level as a result of erroneous model-generated data rather than showing a clearly rising trend. After 10 episodes, the *MSE* of the model converges within 0.05 rapidly, indicating strong model predictive accuracy. Then, the rewards significantly increase and model-based RL agents converge quickly. Therefore, quickly and massively generated data can speed up the learning process at the early training stage, but it is also limited to inaccuracy. Additionally, it should be noticed that the *MSE* of MBSAC is larger than that of MBDDQN at the beginning of training. This indicates that the continuous control displays more similar actions than the discrete control, which makes training the model more challenging.

Another intriguing challenge is how to strike a balance between sample efficiency and training outcomes, because the model accuracy has a significant impact on the training process. Since the results shown in Section 3 used 10% real experience and 90% model-generated experience, we carried out two further experiments to train RL agents using 50% real experience and 50% model-generated experience. As illustrated in Fig. 15, model-based RL algorithms trained with different ratios of real and simulated experience can accomplish quicker learning speeds and higher training outcomes in contrast to model-free RL algorithms. Model-based RL algorithms employing 90% simulated data, as opposed to those using 50% simulated data, converged more quickly because of a high sample efficiency. MBDDQN with 90% simulated experience converges at the 22nd episode while MBDDQN with 50% simulated experience converges at the 32nd episode; MBSAC with 90% simulated data converges at the 35th episode while MBSAC with 50% simulated experience converges at the 40th episode. Meanwhile, model-based RL algorithms with 50% simulated experience have slower convergence rates but better convergence outcomes, because more real and accurate experience has been sampled from the environment. As a result, MBDDQN with 50% simulated experience achieves a reward that is 1.59% higher than it does with 90% simulated data, while MBSAC with 50% simulated experience achieves a reward that is 4.82% higher than it does with 90% simulated experience. Therefore, a higher proportion of generated to real experience will increase sampling efficiency and accelerate the learning speed, but it will also decrease the control performance because of erroneous experience data.

4.2. Limitations and future research

One of the main limitations of this study is that all RL algorithms are all tested in the virtual environment. In the virtual environment, uncertainties are not included in the experiment, including prediction error, measurement error, and mismatch between simulation and reality. These uncertainties can influence the performance of RL algorithms which is strongly related to the data quality [45,46]. Additionally, it may decrease the accuracy of the learned environment model, which may lead to a performance overestimation of model-based RL [47]. Hence, to investigate the real performance gaps of RL algorithms, it is important to apply the RL algorithms in the real world.

Another limitation of this research is that we only compare the Dyna-style MBPO with the model-free algorithms. However, the model in the model-based RL also can be employed to predict the state in advance and allow the agent to make decisions that fit within the indoor temperature limits [24]. It should be noted that the thermal deviation at the early training stage may be too large to be accepted in reality. In the future, we plan to focus on combining the Dyna-style framework with safe RL frameworks to help the agent learn more safely and quickly and thus enable the direct application of RL control.

5. Conclusion

RL shows great potential to enhance energy efficiency and maintain indoor comfort in HVAC systems. In this paper, the performance gaps between model-free and model-based RL algorithms are investigated in terms of indoor temperature, energy consumption, operational cost, data efficiency, and computational time. The open-source BOPTEST framework is employed as the virtual environment and the test case "*BESTEST Hydronic Heat Pump*" is selected. Two different model-free RL algorithms, DDQN and SAC, and a

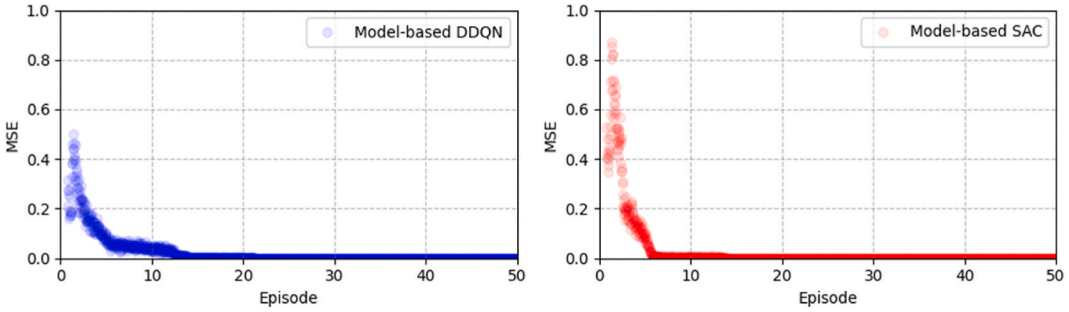

Fig. 14. The *MSE* of the model during the training process.

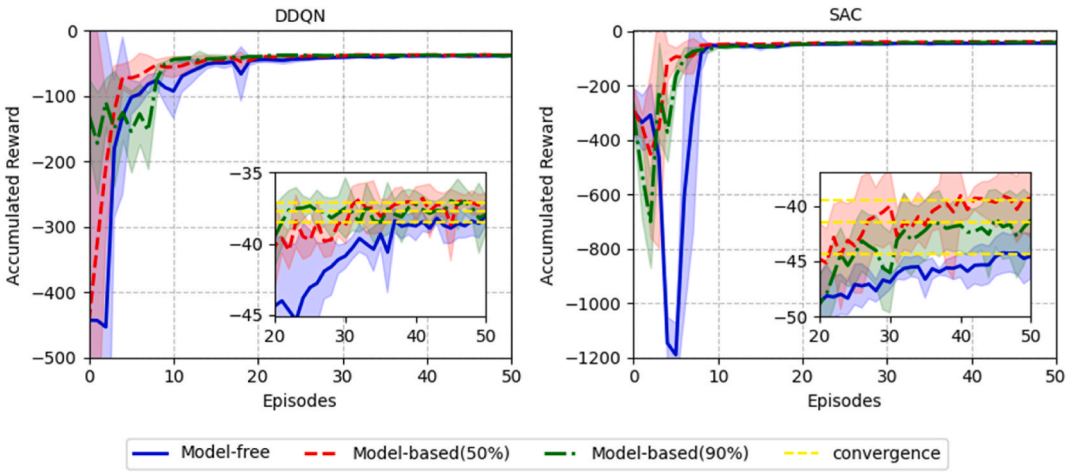

Fig. 15. Learning curves of RL algorithms with different combinations of real experience and hypothetical simulated experience (The solid lines indicate the mean and shaded areas indicate the standard deviation of five trials over different random seeds).

model-based RL framework (MBPO) are employed as representatives for both discrete and continuous control problems. The major findings and conclusions of this paper are listed as follows.

- With the same training dataset, MBDDQN and MBSAC can both achieve faster convergence than DDQN and SAC by 14 and 12 episodes ahead respectively, which saves nearly 5000 timesteps. Furthermore, MBDDQN and MBSAC exhibit significantly higher episodic rewards and lower reward deviation between different trials, which illustrates that model-based RL can produce more stable and better results with less training time, compared to model-free RL.
- During the test period, model-free and model-based RL controls can both outperform the baseline control in terms of indoor temperature, operation cost, and energy consumption. MBDDQN achieves the same cost savings as DDQN but manages to achieve a lower zone thermal deviation with a reduction of 42.86%, while MBSAC yields a lower operation cost than SAC with a reduction of 10.98% but increases the thermal deviation of 61.60%.
- Due to massive and quickly model-generated data, model-based RL can speed up the learning process even though the model is inaccurate at the early training stage. Additionally, the ratio of generated to real experience is positively connected with learning speed and negatively associated with control performance.

In a nutshell, this paper demonstrates that model-based RL has the potential to outperform model-free RL in terms of sample efficiency and control performance even without pre-built models. Our findings can provide a thorough understanding of different RL control techniques which may shed some light on the selection and enhancement of RL controls for HVAC systems.

Author statement

Cheng Gao: Conceptualization, Methodology, Software, Writing - Original Draft, Investigation, Formal analysis, Visualization, Data Curation.

Dan Wang: Conceptualization, Writing - Review & Editing, Validation, Visualization, Resources, Software.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 52208093). This work was supported in part by the Beijing Postdoctoral Research Foundation (No.2022-ZZ-095) and the Urban Carbon Neutral Science and Technology Innovation Funding from Beijing University of Technology (No. 040000514122610). The authors are grateful for the support of these sponsors.

Appendix A. . Thermo-physical properties of the test model

Name		Thickness (m)	Thermal conductivity (W/m·K)	Specific Heat Capacity (J/kg·K)	Density (kg/m^3)
Exterior Wall	Layer1 (Wood siding)	0.009	0.14	900	530
	Layer2 (Insulation)	0.0615	0.04	1400	10
	Layer3 (Concrete block)	0.1	0.51	1000	1400
Floor	Layer1 (Concrete)	0.15	1.4	840	2100
	Layer2 (Insulation)	0.20	0.02	1470	30
	Layer3 (Screed)	0.05	0.6	840	1100
	Layer4 (Tile)	0.01	1.4	840	2100
Roof	Layer1 (Roof deck)	0.019	0.14	900	530
	Layer2 (Fiber glass)	0.1118	0.04	840	12
	Layer3 (Plaster board)	0.01	0.16	840	950

References

[1] A.A. Al-Shargabi, A. Almhafdy, D.M. Ibrahim, et al., Buildings' energy consumption prediction models based on buildings' characteristics: research trends, taxonomy, and performance measures, J. Build. Eng. 54 (2022), 104577, https://doi.org/10.1016/j.jobe.2022.104577.

[2] D. Wang, X. Pang, W. Wang, et al., Evaluation of the dynamic energy performance gap of green buildings: case studies in China, Build. Simulat. 13 (6) (2020) 1191–1204, https://doi.org/10.1007/s12273-020-0653-y.

[3] O.F. Yildiz, M. Yilmaz, A. Celik, Reduction of energy consumption and CO2 emissions of HVAC system in airport terminal buildings, Build. Environ. 208 (2022), 108632, https://doi.org/10.1016/j.buildenv.2021.108632.

[4] H. Wang, Q. Chen, Impact of climate change heating and cooling energy use in buildings in the United States, Energy Build. 82 (2014) 428–436, https://doi.org/10.1016/j.enbuild.2014.07.034.

[5] S. Taheri, P. Hosseini, A. Razban, Model predictive control of heating, ventilation, and air conditioning (HVAC) systems: a state-of-the-art review, J. Build. Eng. (2022), 105067, https://doi.org/10.1016/j.jobe.2022.105067.

[6] X. Li, T. Zhao, P. Fan, et al., Rule-based fuzzy control method for static pressure reset using improved Mamdani model in VAV systems, J. Build. Eng. 22 (2019) 192–199, https://doi.org/10.1016/j.jobe.2018.12.005.

[7] D. Wang, Y. Chen, W. Wang, et al., Field test of Model Predictive Control in residential buildings for utility cost savings, Energy Build. (2023), 113026, https://doi.org/10.1016/j.enbuild.2023.113026.

[8] H.S. Ganesh, K. Seo, H.E. Fritz, et al., Indoor air quality and energy management in buildings using combined moving horizon estimation and model predictive control, J. Build. Eng. 33 (2021), 101552, https://doi.org/10.1016/j.jobe.2020.101552.

[9] Z. Wang, T. Hong, Reinforcement learning for building controls: the opportunities and challenges, Appl. Energy 269 (2020), 115036, https://doi.org/10.1016/j.apenergy.2020.115036.

[10] D. Wang, C. Gao, Y. Sun, W. Wang, S. Zhu, Reinforcement Learning Control Strategy for Differential Pressure Setpoint in Large-Scale Multi-Source Looped District Cooling System, Energy & Buildings, 2023, https://doi.org/10.1016/j.enbuild.2023.112778.

[11] A. Dmitrewski, M. Molina-Solana, Arcucci R. CntrlDA, A building energy management control system with real-time adjustments. Application to indoor temperature, Build. Environ. 215 (2022), 108938, https://doi.org/10.1016/j.buildenv.2022.108938.

[12] D. Wang, W. Zheng, Z. Wang, et al., Comparison of reinforcement learning and model predictive control for building energy system optimization, Appl. Therm. Eng. 228 (2023), 120430, https://doi.org/10.1016/j.applthermaleng.2023.120430.

[13] D. Azuatalam, W.L. Lee, F. de Nijs, et al., Reinforcement learning for whole-building HVAC control and demand response, Energy and AI 2 (2020), 100020, https://doi.org/10.1016/j.egyai.2020.100020.

[14] Y. Du, H. Zandi, O. Kotevska, et al., Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning, Appl. Energy 281 (2021), 116117, https://doi.org/10.1016/j.apenergy.2020.116117.

[15] M. Biemann, F. Scheller, X. Liu, et al., Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control, Appl. Energy 298 (2021), 117164, https://doi.org/10.1016/j.apenergy.2021.117164.

[16] Z. Li, Z. Sun, Q. Meng, et al., Reinforcement learning of room temperature set-point of thermal storage air-conditioning system with demand response, Energy Build. 259 (2022), 111903, https://doi.org/10.1016/j.enbuild.2022.111903.

[17] L. Yu, Z. Xu, T. Zhang, et al., Energy-efficient personalized thermal comfort control in office buildings based on multi-agent deep reinforcement learning, Build. Environ. 223 (2022), 109458, https://doi.org/10.1016/j.buildenv.2022.109458.

[18] X. Deng, 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, https://doi.org/10.1016/j.buildenv.2021.108680.

[19] T. Wang, X. Bao, I. Clavera, et al., Benchmarking Model-Based Reinforcement Learning, 2019, https://doi.org/10.48550/arXiv.1907.02057 arXiv preprint arXiv:1907.02057.

[20] Q. Xiong, Z. Li, W. Cai, et al., Model free optimization of building cooling water systems with refined action space, Build. Simulat. 16 (2023) 615–627, https://doi.org/10.1007/s12273-022-0956-2.

[21] R.Z. Homod, H. Togun, A.K. Hussein, et al., Dynamics analysis of a novel hybrid deep clustering for unsupervised learning by reinforcement of multi-agent to energy saving in intelligent buildings, Appl. Energy 313 (2022), 118863, https://doi.org/10.1016/j.apenergy.2022.118863.

[22] B. Sun, P.B. Luh, Q.S. Jia, et al., Event-based optimization within the Lagrangian relaxation framework for energy savings in HVAC systems, IEEE Trans. Autom. Sci. Eng. 12 (4) (2015) 1396–1406, https://doi.org/10.1109/TASE.2015.2455419.

[23] B. Li, L. Xia, A multi-grid reinforcement learning method for energy conservation and comfort of HVAC in buildings, in: 2015 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, 2015, pp. 444–449, https://doi.org/10.1109/CoASE.2015.7294119.

[24] W. Sun, N. Jiang, A. Krishnamurthy, et al., Model-based rl in contextual decision processes: pac bounds and exponential improvements over model-free approaches, Conference on learning theory. PMLR (2019) 2898–2933.

[25] T.M. Moerland, J. Broekens, C.M. Jonker, Model-based Reinforcement Learning: A Survey, 2020, https://doi.org/10.1561/2200000086 arXiv preprint arXiv: 2006.16712.

[26] B. Peng, X. Li, J. Gao, et al., Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning, 2018, https://doi.org/10.48550/arXiv.1801.06176 arXiv preprint arXiv:1801.06176.

[27] Q. Huang, Model-based or model-free, a review of approaches in reinforcement learning, in: 2020 International Conference on Computing and Data Science (CDS), IEEE, 2020, pp. 219–221, https://doi.org/10.1109/CDS49703.2020.00051.

[28] C. Zhang, S.R. Kuppannagari, R. Kannan, et al., Building HVAC scheduling using reinforcement learning via neural network based model approximation, in: Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, 2019, pp. 287–296, https://doi.org/10.1145/3360322.3360861.

[29] S.M. Dawood, A. Hatami, R.Z. Homod, Trade-off decisions in a novel deep reinforcement learning for energy savings in HVAC systems, Journal of Building Performance Simulation 15 (6) (2022) 809–831, https://doi.org/10.1080/19401493.2022.2099465.
[30] Z. Zhang, C. Zhang, K.P. Lam, A Deep Reinforcement Learning Method for Model-Based Optimal Control of HVAC Systems, 2018, https://doi.org/10.14305/ibpc.2018.ec-1.01.
[31] J. Arroyo, C. Manna, F. Spiessens, et al., Reinforced model predictive control (RL-MPC) for building energy management, Appl. Energy 309 (2022), 118346, https://doi.org/10.1016/j.apenergy.2021.118346.
[32] D. Blum, J. Arroyo, S. Huang, et al., Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings, Journal of Building Performance Simulation 14 (5) (2021) 586–610, https://doi.org/10.1080/19401493.2021.1986574.
[33] Y. Wang, Y. Wu, Y. Tang, et al., Cooperative energy management and eco-driving of plug-in hybrid electric vehicle via multi-agent reinforcement learning, Appl. Energy 332 (2023), 120563, https://doi.org/10.1016/j.apenergy.2022.120563.
[34] Y. Gao, Y. Matsunami, S. Miyata, et al., Operational optimization for off-grid renewable building energy system using deep reinforcement learning, Appl. Energy 325 (2022), 119783, https://doi.org/10.1016/j.apenergy.2022.119783.
[35] A.S. Polydoros, L. Nalpantidis, Survey of model-based reinforcement learning: applications on robotics, J. Intell. Rob. Syst. 86 (2) (2017) 153–173, https://doi.org/10.1007/s10846-017-0468-y.
[36] M. Esrafilian-Najafabadi, F. Haghighat, Towards Self-Learning Control of HVAC Systems with the Consideration of Dynamic Occupancy Patterns: Application of Model-free Deep Reinforcement Learning, Building and Environment, 2022, 109747, https://doi.org/10.1016/j.buildenv.2022.109747.
[37] B. Hu, J. Li, S. Li, et al., A hybrid end-to-end control strategy combining dueling deep Q-network and PID for transient boost control of a diesel engine with variable geometry turbocharger and cooled EGR, Energies 12 (19) (2019) 3739, https://doi.org/10.3390/en12193739.
[38] B. Peng, Q. Sun, S.E. Li, et al., End-to-End autonomous driving through dueling double deep Q-network, Automotive Innovation 4 (3) (2021) 328–337, https://doi.org/10.1007/s42154-021-00151-3.
[39] T. Haarnoja, A. Zhou, P. Abbeel, et al., Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International Conference on Machine Learning, PMLR, 2018, pp. 1861–1870.
[40] M. Janner, J. Fu, M. Zhang, et al., When to trust your model: model-based policy optimization, Adv. Neural Inf. Process. Syst. 32 (2019).
[41] T. Yu, G. Thomas, L. Yu, et al., Mopo: model-based offline policy optimization, Adv. Neural Inf. Process. Syst. 33 (2020) 14129–14142.
[42] L. Kaiser, M. Babaeizadeh, P. Milos, et al., Model-based Reinforcement Learning for Atari, arXiv preprint arXiv:1903.00374, 2019, https://doi.org/10.48550/arXiv.1903.00374.
[43] T. Kurutach, I. Clavera, Y. Duan, et al., Model-ensemble trust-region policy optimization, arXiv preprint arXiv:1802.10592, https://doi.org/10.48550/arXiv.1802.10592, 2018.
[44] G. Brockman, V. Cheung, L. Pettersson, et al., Openai Gym, 2016, https://doi.org/10.48550/arXiv.1606.01540 arXiv preprint arXiv:1606.01540.
[45] 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, https://doi.org/10.1016/j.buildenv.2019.106535.
[46] J.Y. Park, T. Dougherty, H. Fritz, et al., LightLearn: an adaptive and occupant centered controller for lighting based on reinforcement learning, Build. Environ. 147 (2019) 397–414, https://doi.org/10.1016/j.buildenv.2018.10.028.
[47] R.Z. Homod, Z.M. Yaseen, A.K. Hussein, et al., Deep clustering of cooperative multi-agent reinforcement learning to optimize multi chiller HVAC systems for smart buildings energy management, J. Build. Eng. 65 (2023), 105689, https://doi.org/10.1016/j.jobe.2022.105689.