Energy & Buildings 199 (2019) 472–490

Contents lists available at ScienceDirect

Energy & Buildings

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

Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning

Zhiang Zhang[a,*], Adrian Chong[b], Yuqi Pan[c], Chenlu Zhang[a], Khee Poh Lam[a,b]

[a] Center for Building Performance and Diagnostics, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 15213, USA
[b] Department of Building, School of Design and Environment, National University of Singapore, 4 Architecture Dr, 117566, Singapore
[c] Ghafari Associates LLC, 17101 Michigan Avenue, Dearborn, MI, 48126, USA

ARTICLE INFO

Article history:
Received 28 February 2019
Revised 12 July 2019
Accepted 16 July 2019
Available online 16 July 2019

Keywords:
HVAC
Energy efficiency
Whole building energy model
Optimal control
Deep reinforcement learning

ABSTRACT

Whole building energy model (BEM) is a physics-based modeling method for building energy simulation. It has been widely used in the building industry for code compliance, building design optimization, retrofit analysis, and other uses. Recent research also indicates its strong potential for the control of heating, ventilation and air-conditioning (HVAC) systems. However, its high-order nature and slow computational speed limit its practical application in real-time HVAC optimal control. Therefore, this study proposes a practical control framework (named BEM-DRL) that is based on deep reinforcement learning. The framework is implemented and assessed in a novel radiant heating system in an existing office building as a case study. The complete implementation process is presented in this study, including: building energy modeling for the novel heating system, multi-objective BEM calibration using the Bayesian method and the Genetic Algorithm, deep reinforcement learning training and simulation results evaluation, and control deployment. By analyzing the real-life control deployment data, it is found that BEM-DRL achieves 16.7% heating demand reduction with more than 95% probability compared to the old rule-based control. However, the framework still faces the practical challenges including building energy modeling of novel HVAC systems and multi-objective model calibration. Systematic study is also needed for the design of deep reinforcement learning training to provide a guideline for practitioners.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Heating, ventilation and air conditioning (HVAC) systems are the major energy consumer in non-residential buildings. In past decades, HVAC systems have experienced great improvements in energy efficiency thanks to better system design and HVAC equipment efficiency. However, inappropriate control and operation can still lead to poor energy efficiency, which is often overlooked by HVAC engineers and facility managers. Therefore, optimal control strategies for HVAC systems have become a popular research topic in recent years.

Most optimal control methods require a model. Whole building energy model (BEM) can be potentially a good candidate. BEM has been widely used in building design because of the requirements from various green and energy efficient building standards and certifications. Reusing the design-phase BEM for building control can potentially reduce the development cost of the HVAC optimal control strategies. The concept of BEM (simulation)-based building system control dates back to 2001 when Mahdavi [1] concluded that this method can deliver a complex control strategy that achieves “habitability, sustainability and feasibility” simultaneously. Zhao [2] proposed a design-build-operate energy information modeling infrastructure (DBO-EIM) in 2015 to demonstrate how a design-phase BEM can be used for HVAC control and operation. This study follows the concept of DBO-EIM, and aims to propose and evaluate a practical framework to use BEM for HVAC control.

1.1. BEM-based predictive control

Model predictive control (MPC) is one of the most popular model-based control methods for HVAC systems, and recent studies apply MPC to control supply heating/cooling power [3–8], supply water temperature [9,10], supply air temperature [11–14], supply air flow rate [11,13,15], indoor temperature setpoints [16–18], etc. However, the existing MPC solvers are not applied to high-order models and cost functions. Therefore, BEM, as a simulation program, cannot be directly used in the existing MPC methods.

[*] Corresponding author.
E-mail address: zhiangz@andrew.cmu.edu (Z. Zhang).

https://doi.org/10.1016/j.enbuild.2019.07.029
0378-7788/© 2019 Elsevier B.V. All rights reserved.

Nomenclature

General

${}^{*}_{t}$	(subscript t) any variables at control time step t
A3C	asynchronous Advantage Actor Critic algorithm
API	application Programming Interface
BAS	building Automation System
BEM	whole Building Energy Model
$BEM - DRL$	whole building energy model-based deep reinforcement learning control framework for HVAC systems
CVRMSE	coefficient of Variance of the Root Mean Square Error
DRL	deep Reinforcement Learning
GP	Gaussian Process
HVAC	heating Ventilation and Air-conditioning
NMBE	normalized Mean Bias Error
PID	proportional-integral-derivative
PMV	predicted Mean Vote
PPD	predicted Percentage of Dissatisfied
RBC	rule-based Control
RL	reinforcement Learning
TCP	transmission Control Protocol
*TMY*3	the third Typical Meteorological Year weather data

Automated calibration (Section 2.2.1)

δ	discrepancy between the simulation and observation caused by the model inadequacy
ϵ	observation error
η	simulated building performance behavior
μ_n	convex combination weight for the parameter n
ζ	true building performance behavior
L	likelihood function
P	probability distribution
T	feasible discrete set for t
t	calibration parameter
t^*	true value for the calibration parameter
x	observable input parameter
y	observed building performance behavior

Reinforcement learning (Section 2.3)

α	learning rate in gradient descent optimization
β	value loss weight
$\boldsymbol{\theta}$	weight vector in a function
γ	reward discount factor
κ	entropy weight
$\mathbb{E}$	expected value
π	control policy
A_t	control action at control time step t
d	stationary distribution
ob	user-defined agent's observation
P	probability distribution
q	action-value function
R_t	reward at control time step t
S_t	environment state at control time step t
T_∞	control time step at the infinite future
v	state-value function

Normalized energy saving performance evaluation (Section 2.5)

$\mathbf{x}$	Gaussian process model input
$E_{basedaily}$	baseline daily HVAC energy consumption
$E_{basetotal}$	baseline total HVAC energy consumption
i	one day in the new control strategy deployment period
j	one sample in the baseline total HVAC energy consumption samples
m	m days in the new control strategy deployment period
n	n samples of the baseline total HVAC energy consumption
μ	mean of a distribution
σ	standard deviation of a distribution

Case study (Section 3)

$\dot{m}$	mass flow rate of the supply hot water of the Mullion system
τ, ρ, β, λ	hyperparameters in the reward function for the case study
APD	actual Percentage of Dissatisfied collected through a smart phone App in real time
C_p	specific heat of water at constant pressure
CV	coefficient of variation
IAT	indoor air temperature
IW	the Intelligent Workplace (the case study building)
KDE	kernel density estimation
OAT	outdoor air temperature
Occp	occupancy status flag (1 or 0)
PDF	probability density function
Q_{Mull}	heating demand of the Mullion system
SP_{thres}	temperature penalty threshold in the reward function for the case study
Std	standard deviation
*T*1, *SP*1	average indoor air temperature of IW and its setpoint
*T*2, *SP*2	supply water temperature of the Mullion system and its setpoint
*T*3	return water temperature of the Mullion system

Workarounds and simplifications have to be proposed to integrate BEM into model-based predictive control framework. For example, Zhao et al. [19] proposes a real-time EnergyPlus model-based predictive control (EPMPC) method for the HVAC supply air temperature setpoint, but the prediction horizon of the case study is set to the minimum (i.e., one control time step). A successive study [20] of EPMPC finds the scalability of the method is limited because EnergyPlus is too slow for the real-time simulation. Ascione et al. [21] and May-Ostendorp et al. [22] propose the non-real-time BEM-based control method, which uses heuristic optimization to pre-calculate the optimal setpoint schedules for the next day. Aftab et al. [23] and Miezis et al. [24] propose the BEM-based predictive control method to solve a simpler control problem, the optimal start of a system. Kwak et al. [25] develops an optimization-free control algorithm based on EnergyPlus, in which the outdoor air damper at the current time step is controlled based on the EnergyPlus-predicted air handling unit supply air enthalpy of the next time step. However, all of the above-mentioned studies have only been tested in simulators, so their real-world performance, including practical feasibility and energy efficiency performance, is unknown.

1.2. Reinforcement learning control

Reinforcement learning (RL) control can be a “model-free” control method, i.e., an RL agent has no prior knowledge about the controlled process. RL learns an optimal control strategy by “trial-and-error”. Therefore, it can be an online learning method that learns an optimal control strategy during actual building operations [26–37]. However, in HVAC control, online learning may introduce unstable and poor control actions at the initial stage of the learning. In addition, it may take a long time (e.g. 20 days reported in [29], 40 days reported in [32] and over 50 days reported in [30]) for an RL agent to converge to a stable control policy for some cases. Therefore, some studies choose to use an HVAC simulator to the train the RL agent offline [27,38–42]. Unlike MPC, simulators with arbitrary high complexity (like BEM) can be directly used to train RL agents because of its “model-free” nature.

Deep reinforcement learning (DRL) has no fundamental differences with classical reinforcement learning. While classical reinforcement learning uses simple tabular settings or linear functions as the RL agent, deep reinforcement learning uses complex deep neural networks instead. The RL agent with deep neural networks has higher representational capability so it can adapt to more complex control problems. In addition, DRL can achieve “end-to-end” control, in which the input to the RL agent is high-dimensional raw sensory data without data prepossessing or feature engineering. The most famous experiment of DRL is the control of Atari video games, in which the RL agent only uses the raw game frames as the input to match or beat human’s performance [43]. Recent studies use computer simulations to show DRL can also achieve promising energy efficiency improvement for HVAC control [31,40,41,44,45].

Despite of the promising potential of RL for HVAC control, there are a limited number of studies on its practical implementation and evaluation [46]. Most existing studies are based on the simulations of box-shaped building models and simple HVAC systems (e.g. electric heater). It is known that real-world buildings and HVAC systems are usually more complex than the hypothetical computer models, so the practical implementation and evaluation are necessary to further understand the potentials and limitations of RL control. A short-term implementation study is reported in [38], in which a tabular setting RL control method was implemented in a real-life single-story test facility in the cooling season. However, the deployment experiment of this study lasts for only 5 days and the energy efficiency analysis is still based on simulations.

1.3. Objectives

This study develops a practical framework based on deep reinforcement learning to use a whole building energy model for HVAC control. The proposed framework is implemented in an existing radiant heating system of an office building as the case study. By the case study, the complete methodology of the proposed framework is demonstrated and evaluated for its practicability. The energy efficiency performance of the control method is evaluated using both simulations and deployment results.

2. Control framework

An overview of the BEM-based deep reinforcement learning control framework (BEM-DRL) [46] for HVAC systems is shown in Fig. 1, which includes four steps:

1. Building energy modeling: A building energy model is first created using a BEM engine. EnergyPlus [47] is selected in this study but any other BEM engines can also be used. The model will be used as the simulator for the offline DRL training.
2. Model calibration: The BEM built in the previous step is calibrated using the observed data to minimize the gap between simulations and observations.
3. DRL training: The calibrated BEM is used to train the RL agent off-line to develop an optimal control strategy for the target HVAC system.
4. Deployment: The trained RL agent is deployed in the building automation system to generate control signals for the target HVAC system in real-time.

The following sections will explain each step in the control framework in detail, as well as a normalized approach to evaluate the energy saving performance of the DRL control.

2.1. BEM-DRL step 1: building energy modeling

A whole building energy model of the target building should be firstly built based on the information of the building geometry, envelope, thermal zoning, HVAC systems, internal loads and operation schedules. The existing building energy model built at the building design phase can also be used.

The model will be used as a simulator that, at every control time step t, takes a control action a to predict the energy performance metrics (EnergyMetrics) and comfort performance metrics (ComfortMetrics), as shown below:

$$BEM : a_t \rightarrow \{\text{EnergyMetrics}, \text{ComfortMetrics}\}_t. \tag{1}$$

The EnergyMetrics and ComfortMetrics may be different for different control problems.

2.2. BEM-DRL step 2: model calibration

The BEM obtained in the previous step may contain errors. The errors may come from incorrect design drawings, incorrect assumptions for the unknown parameters, technical limitations of the building energy modeling engine, etc. Therefore, the BEM should be calibrated against the actual building operation data to minimize the gap between the simulation and observation. Since the control time step is usually less than one hour, the BEM should be calibrated at hourly or sub-hourly resolution.

2.2.1. Automated calibration

Automated calibration approach is adopted in this framework. This approach compares the observed and simulated building performance data, and uses various types of mathematical and statistical approach to minimize the discrepancy between the observation and simulation. The BEM should be calibrated for both EnergyMetrics and ComfortMetrics.

Two different automated calibration methods are studied in this paper, including Bayesian calibration and Genetic Algorithm optimization.

Bayesian calibration

This study adopts the method proposed by Chong and Menberg [48] for Bayesian calibration. This section briefly describes the theoretical basis of Bayesian calibration and the implementation details can be found in [48].

The statistical formulation of Bayesian calibration is

$$y(x) = \zeta(x) + \epsilon(x) = \eta(x, t^*) + \delta(x) + \epsilon(x), \tag{2a}$$

$$\text{s.t. } \delta(x) = 0, \tag{2b}$$

where y, ζ, η are the observed/true/simulated building performance behavior (i.e., calibration objectives such as HVAC energy consumption) respectively, x is the observable input parameter (i.e., the parameters that can be observed but cannot be manipulated such as weather conditions), t is the calibration parameter with the unknown true value t^* (i.e., the manipulable BEM parameters such as infiltration rate), δ is a discrepancy term to correct any model inadequacy, ϵ is the observation error. Note that in Eq. (2b), $\delta(x)$ is forced to be zero because we assume the BEM can adequately model the target HVAC system.

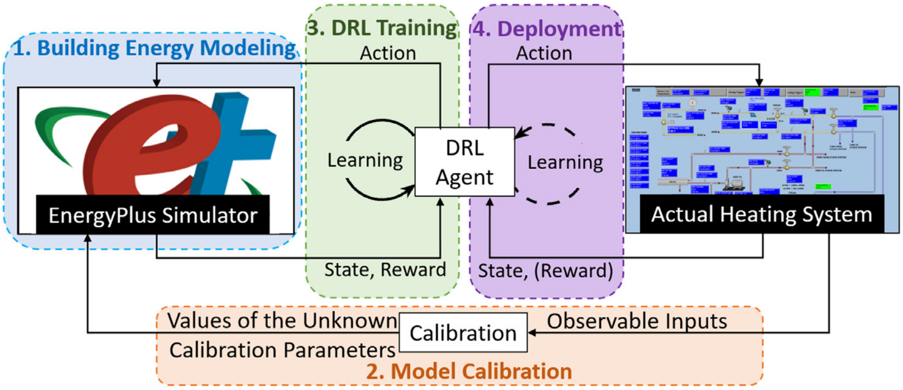

Fig. 1. BEM-based DRL control framework [46].

If we assume the distribution of ϵ is Gaussian, we can write the posterior distribution of t given y as the following based on Bayes' theorem:

$$P(t|y) \propto L\big(y|\eta(t)\big) \times P(t) \tag{3}$$

where P represents a probability distribution and L represents a likelihood function. Markov chain Monte Carlo (MCMC) sampling is then used to numerically obtain the posterior distribution $P(t|y)$ in Eq. (3). The modes of the distribution are regarded as the calibrated parameter values that are fed back into the BEM.

The computation time of Bayesian calibration increases exponentially with the size of the calibration dataset, i.e., the number of data entries of $y(x)$ and $\eta(x, t)$. To adapt Bayesian calibration for sub-hourly resolution data, we adopt the method proposed in [49] to down-sample the original calibration dataset to a smaller one.

As discussed before, the model must be calibrated for multiple building performance metrics. However, the Bayesian calibration method of [48] is designed for single-objective calibration. Therefore, a convex combination method [50] is proposed to combine multiple building performance metrics into one. As a result, the y and η in Eq. (2a) are obtained by the convex combination of multiple building performance metrics, i.e.,

$$\begin{cases} y = \mu_1 y_1 + \mu_2 y_2 +, \ldots, +\mu_n y_n, \\ \eta = \mu_1 \eta_1 + \mu_2 \eta_2 +, \ldots, +\mu_n \eta_n, \end{cases} \tag{4a}$$

$$\text{s.t. } \mu_1 + \mu_2 + \ldots + \mu_n = 1, \tag{4b}$$

$$\mu_1, \mu_2, \ldots, \mu_n \geq 0, \tag{4c}$$

where $\mu_1, \mu_2, \ldots \mu_n$ are the convex combination weights, $y_1, y_2, \ldots, y_n$ and $\eta_1, \eta_2, \ldots, \eta_n$ are the observed and simulated building performance metrics.

Genetic algorithm

Genetic Algorithm (GA) is a heuristic search method that is used for multi-objective optimization. Therefore, the BEM calibration is formulated into a multi-objective optimization problem, and GA can find the calibration parameter values that minimizes the discrepancy between the observed and the simulated building performance metrics. Eq. (5a) shows the mathematical formulation of the multi-objective optimization problem,

$$\min_t \big(c_1(x, t), c_2(x, t), \ldots, c_n(x, t)\big), \tag{5a}$$

$$s.t. \; t \in T, \tag{5b}$$

where c_n is the error function representing the discrepancy between an observed and simulated building performance metric, x is the observable input parameter (e.g. weather conditions), t is the calibration parameter and T is a discrete set representing the feasible choices of t. This study uses Non-Dominated Sorting Genetic Algorithm II (NSGA-II) algorithm. It differs from other genetic algorithms by adding the crowding distance into the selection process to keep the high-ranked solutions far apart [51]. This process ensures the diversity of the solutions and helps explore the entire solution space. More details of NSGA-II can be found in [51].

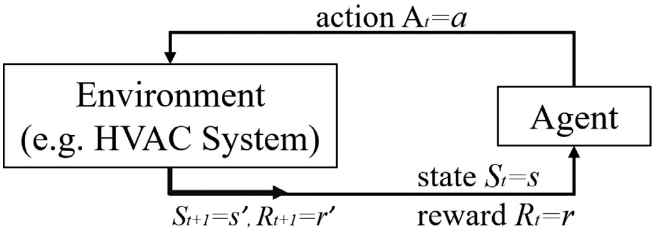

Fig. 2. A Standard Reinforcement Learning Problem [46].

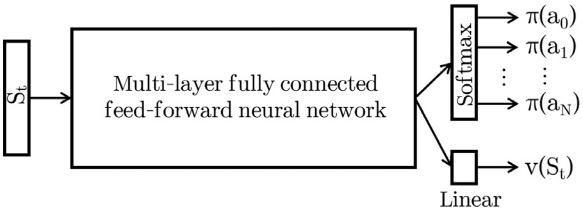

Fig. 3. Policy and state-value function architecture [46].

2.2.2. Model evaluation

The calibrated BEM is evaluated by two error metrics, normalized mean bias error (NMBE) and cumulative variation of the root mean square error (CVRMSE), using hourly or sub-hourly building performance data. ASHRAE Guideline 14 [52] suggests hourly NMBE and CVRMSE should be less than 10% and 30% respectively.

2.3. BEM-DRL step 3: DRL training

The calibrated BEM is used as a simulator to train a reinforcement learning agent offline to learn an optimal control policy for the HVAC system.

2.3.1. Technical background of deep reinforcement learning

Standard reinforcement learning problem

A standard reinforcement learning problem, as shown in Fig. 2, is formulated as a discrete control problem where, at the control time step t, the agent observes the state S_t and environment reward R_t to provide the control action A_t [53]. The agent's control policy can be written as $\pi: S_t \to A_t$ and the agent's goal is to find a control policy that maximizes the cumulative environment reward $\sum_t^{T_\infty} R_t$ at each control time step. The control policy can also be written as a distribution, that is:

$$\pi(s,a) = P(a|s), \tag{6}$$

which represents the probability of taking the action a given the state s.

A reinforcement learning problem can be defined by two closely-related value functions, including state-value function $v_\pi(s)$ and action-value function $q_\pi(s, a)$, as shown below:

$$v_\pi(s) \doteq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \tag{7a}$$

$$\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(s') | S_t = s, S_{t+1} = s'], \tag{7b}$$

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \tag{8a}$$

$$\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(s') | S_t = s, A_t = a, S_{t+1} = s'], \tag{8b}$$

where γ is the reward discount factor [53]. The value functions show the expected cumulative discounted reward of a control policy π at a state or a {state, action} tuple.

The control policy and the value functions can be represented by the parameterized functions (function approximation), i.e.,

$$v_\pi(s) \approx v_\pi(s, \boldsymbol{\theta_v}) = v_{\boldsymbol{\theta}_v}(s), \tag{9}$$

$$q_\pi(s,a) \approx q_\pi(s, a, \boldsymbol{\theta_q}) = q_{\boldsymbol{\theta}_q}(s,a), \tag{10}$$

$$\pi(s,a) \approx \pi(s, a, \boldsymbol{\theta}) = \pi_{\boldsymbol{\theta}}(s,a), \tag{11}$$

where $\boldsymbol{\theta_v}$, $\boldsymbol{\theta_q}$, $\boldsymbol{\theta}$ are weight vectors. Any parameterized functions can be used, such as linear functions or deep neural networks.

Advantage actor critic (A2C)

A standard reinforcement learning problem can be transformed into an optimization problem which aims to find $\boldsymbol{\theta}$ in $\pi_{\boldsymbol{\theta}}(s, a)$ that maximizes the average reward per control time step, that is:

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_s d_{\pi_\theta}(s) \sum_a R_s^a \pi_{\boldsymbol{\theta}}(s,a), \tag{12}$$

where $d_{\pi_\theta}(s)$ is stationary distribution for the state s of the Markov chain starting from s_0 following the policy $\pi_{\boldsymbol{\theta}}$, and R_s^a is the environment reward at the state s taking the action a [53].

Gradient descent (GD) is used to solve this optimization problem. The gradient of $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is given by [53]:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_s d_{\pi_\theta}(s) \sum_a R_s^a \pi_{\boldsymbol{\theta}}(s,a) \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(s,a)}{\pi_{\boldsymbol{\theta}}(s,a)} \tag{13a}$$

$$= \sum_s d_{\pi_\theta}(s) \sum_a R_s^a \pi_{\boldsymbol{\theta}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a) \tag{13b}$$

$$= \mathbb{E}_{\pi_\theta}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a) q_{\boldsymbol{\theta}_q}(s,a)] \tag{13c}$$

$$= \mathbb{E}_{\pi_\theta}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a)\left(q_{\boldsymbol{\theta}_q}(s,a) - v_{\boldsymbol{\theta}_v}(s)\right)\right]. \tag{13d}$$

Note that Eq. (13c) is obtained based on policy gradient theorem [53], and a zero-valued "baseline function" $\mathbb{E}_{\pi_\theta}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a) v_{\boldsymbol{\theta}_v}(s)]$ is subtracted from Eq. (13c) to reduce the variance of $q_{\boldsymbol{\theta}_q}$ for better learning stability [53]. Eq. (13d) that uses the action-value function ("actor") and state-value function ("critic") to find the gradient is called A2C, which is in the family of policy gradient methods.

After the gradient is known, the one-step learning update for $\boldsymbol{\theta}$ to solve Eq. (12) can be derived based on gradient descent method:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \mathbb{E}_{\pi_\theta}\left[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a)\left(q_{\boldsymbol{\theta}_q}(s,a) - v_{\boldsymbol{\theta}_v}(s)\right)\right] \tag{14a}$$

$$= \boldsymbol{\theta} + \alpha \mathbb{E}_{\pi_\theta}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a)\left(r' + \gamma v_{\boldsymbol{\theta}_v}(s') - v_{\boldsymbol{\theta}_v}(s)\right)], \tag{14b}$$

where α is the learning rate, r' is the actual environment reward observed at the state s and taking the action a, and s' is the resulting state from the state s and taking the action a. Note that, Eq. (14a) is derived based on the gradient shown in Eq. (13d), and Eq. (14b) is obtained based on the definition of Eq. (8b).

However, Eq. (14b) introduces an unknown variable $\boldsymbol{\theta_v}$, which is the weight in the parameterized state-value function. This parameter can also be learned using gradient descent to minimize the mean squared error between the "true" and the approximated state-value function, that is,

$$\boldsymbol{\theta_v} \leftarrow \boldsymbol{\theta_v} - \alpha \mathbb{E}_{\pi_\theta}\left[\nabla_{\boldsymbol{\theta}_v}\left(v_{true} - v_{\boldsymbol{\theta}_v}(s)\right)^2\right] \tag{15a}$$

$$= \boldsymbol{\theta_v} - \alpha \mathbb{E}_{\pi_\theta}\left[\nabla_{\boldsymbol{\theta}_v}\left(r' + \gamma v_{\boldsymbol{\theta}_v}(s') - v_{\boldsymbol{\theta}_v}(s)\right)^2\right], \tag{15b}$$

where Eq. (15b) follows from the definition of Eq. (7b). Note that during the reinforcement learning, the "true" value of the state-value function (v_{true}) is unknown. Here we use a bootstrapped state-value function $r' + \gamma v_{\boldsymbol{\theta}_v}(s')$ as a biased estimation of the "true" state-value function. This method is also called temporal difference method [53].

To reduce the complexity of the problem, the parameterized functions $v_{\boldsymbol{\theta}_v}$ and $\pi_{\boldsymbol{\theta}}$ can be partially combined with parameter sharing in a neural network [54], as shown in Fig. 3. Therefore, we have the state-value function $v_{\boldsymbol{\theta}_v} = v_\pi(s, \boldsymbol{\theta_{\pi,v}})$ and the policy distribution $\pi_{\boldsymbol{\theta}} = \pi(s, a, \boldsymbol{\theta_{\pi,v}})$.[1] In addition, to encourage the exploration of an RL agent, an entropy term H is added to the learning process as the encouragement for non-deterministic policies (the entropy is the highest for a uniform policy distribution) [54]. Therefore, the final one-step learning update of the parameterized RL agent ($\boldsymbol{\theta_{\pi,v}}$) becomes:

$$\boldsymbol{\theta}_{\boldsymbol{\pi},v} \leftarrow \boldsymbol{\theta}_{\boldsymbol{\pi},v} - \alpha \mathbb{E}_{\pi_{\boldsymbol{\theta}_{\pi,v}}}\left[\beta \nabla_{\boldsymbol{\theta}_{\pi,v}}\left(r' + \gamma v_{\boldsymbol{\theta}_{\pi,v}}(s') - v_{\boldsymbol{\theta}_{\pi,v}}(s)\right)^2\right. \tag{16a}$$

$$- \nabla_{\boldsymbol{\theta}_{\pi,v}} \log \pi_{\boldsymbol{\theta}_{\pi,v}}(s,a)\left(r' + \gamma v_{\boldsymbol{\theta}_{\pi,v}}(s') - v_{\boldsymbol{\theta}_{\pi,v}}(s)\right) \tag{16b}$$

$$\left. - \kappa \nabla_{\boldsymbol{\theta}_{\pi,v}} H(\pi_{\boldsymbol{\theta}_{\pi,v}})\right], \tag{16c}$$

where β is the value loss weight and κ is the entropy weight.

Asynchronous advantage actor critic (A3C)

To improve the learning speed of A2C, this study uses A3C [54] which is an asynchronous variation of A2C. A3C fires a number of local RL agents in parallel to "explore" different regions of a problem. Each local RL agent uses a copy of the same simulator to asynchronously perform GD update for the same global RL agent. The algorithm principle is schematically shown in Fig. 4. Compared to the conventional ϵ-greedy exploration, the asynchronous method can improve the exploration efficiency to solve the "exploitation-exploration" dilemma of reinforcement learning. Details of the algorithm can be found in [54].

[1] The notations here are slightly abused since the state-value function and the policy distribution do not share all the function parameters.

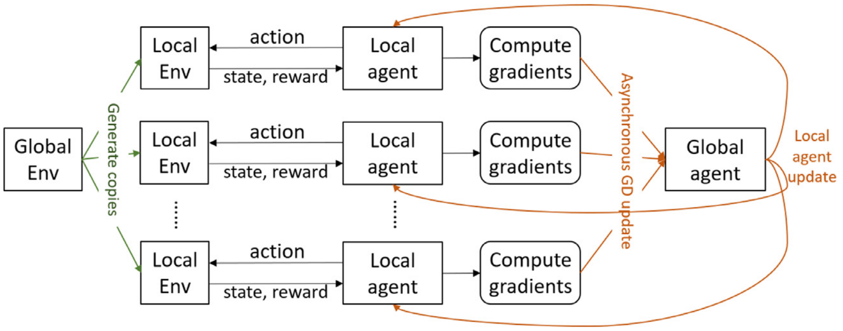

Fig. 4. Schematic diagram of A3C algorithm.

Key terminologies

- Control time step: A time interval when an RL agent observes the state and the reward, execute a control action and wait for the resulting state and reward of the next time step.
- Simulation time step (when a simulator (e.g. EnergyPlus) is used to as the environment): The time step defined in the simulator. It is independent from the control time step. For example, if the control time step is 15-min and the simulation time step is 5-min, this means an RL agent interacts with the simulator by every three simulation time steps.
- Simulator episode: The simulation time period of a simulator. During the RL training, the simulation will be repeated for multiple episodes.
- Interaction times: The number of times that the RL agents (including all the local RL agents of A3C) interact with their environments (one interaction means an RL agent finishes one control time step).

2.3.2. State, action and reward

A reinforcement learning problem must explicitly define the state, action and reward.

The state represents the RL agent's observation of the environment. Since building structures may have slow thermal response, the state should be a stack of the current and historical observations, as shown below:

$$S_t = \left\{ob_t, ob_{t-1}, \ldots, ob_{t-n}\right\}, \quad (17)$$

where *ob* is the user-defined agent's observation at a control time step, *t* is the current control time step, *n* is the history window to be considered [46].

The action is a discrete set of control choices, as shown below,

$$A = \left\{a_1, a_2, \ldots, a_n\right\}, \quad (18)$$

where a_n is a control action choice.

The reward is a scalar value in the range of [0, 1] representing how good is the state and/or the action. The reward is user-defined. The reward value should be small when the energy consumption is high and/or the thermal comfort performance is low, and should be large when the energy consumption is low and the thermal comfort performance is acceptable.

2.3.3. EnergyPlus simulator for reinforcement learning (EPRL)

We have developed an EnergyPlus Simulator for Reinforcement Learning (EPRL) that wraps an EnergyPlus model in a Python-based OpenAI Gym interface [55]. EPRL creates a standard reinforcement learning environment (simulator), so that an RL agent can make interactions with an EnergyPlus model to learn an optimal control policy. EPRL is based on ExternalInterface function of EnergyPlus and BCVTB middleware [56] and its architecture is shown in Fig. 5. The main component of this architecture is an OpenAI gym interface, which is a Python object with three functions: object constructor (*__init__*), *reset()* and *step(a)*. The operation sequence of EPRL is as the followings:

1. Upon the time that the OpenAI gym interface object (gym object) is initiated (by calling the object constructor), a server socket for the inter-program communication is created.
2. The function *reset()* should be called by the RL agent when the learning starts. When it is called, an EnergyPlus instance is created using the input definition file (IDF) and data exchange file (with the.cfg extension) stored in the local drive.
3. The gym object then creates a TCP connection with the EnergyPlus instance, in which ExternalInterface of EnergyPlus performs as a client through BCVTB.
4. The gym object uses the TCP connection to read the initial simulation output from the EnergyPlus instance, and returns the simulation output to the RL agent as the raw state observation. The raw state observation should be processed by the RL agent to extract the state and reward.
5. The RL agent calls the function *step(a)* at each control time step. When it is called, the gym object uses the TCP connection to send the action **a** to the EnergyPlus instance and read the resulting simulation output. The gym object then returns the simulation output to the RL agent as the raw state observation. The raw state observation should be processed by the RL agent to extract the state and reward.

EPRL is available at https://github.com/zhangzhizza/Gym-Eplus.

2.4. BEM-DRL step 4: deployment

The trained RL agent can be statically or dynamically deployed:

- Static deployment: the RL agent is deployed as a static function without any learning going on. This type of deployment requires less computing power than the dynamic deployment. However, the BEM may need to be re-calibrated (i.e. continuous calibration [57]) and the RL agent may need to be re-trained when the key characteristics of the HVAC system changes.
- Dynamic deployment: the RL agent is deployed as a continuous-learning agent. At each control time step, the RL agent obtains the state observation from the HVAC system, computes the control action, observes the resulting next state and reward value, and updates itself as in the DRL training stage. This type of deployment requires much more computing power than the previous type, but the RL agent has the ability to adapt to the changing system characteristics. Note that the dynamic deployment may make the control policy unstable.

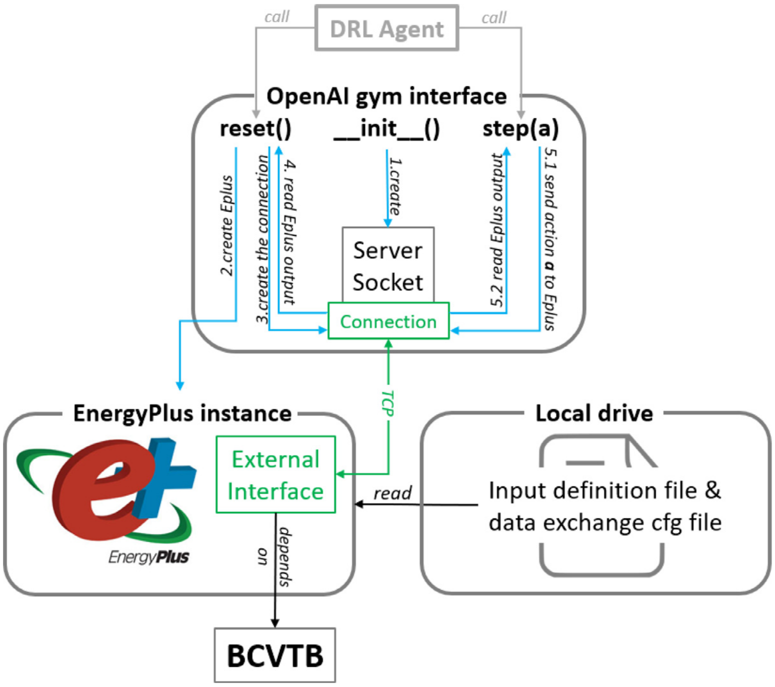

Fig. 5. Architecture of the EnergyPlus simulator for reinforcement learning.

2.5. Normalized energy saving performance evaluation

Energy consumption of an HVAC system is influenced by multiple factors, such as the control strategy, outdoor air temperature, solar radiation, indoor air temperature, etc. Therefore, to evaluate the energy saving of a new control strategy compared to the old one, all the energy-influencing factors must be constant. This can be easily realized in computer simulations but is difficult in real-world implementations.

This study proposes a stochastic data-driven approach to perform the normalized energy saving evaluation of a new control strategy using the real-world operation data [46]. The approach is similar to Weather Normalized Energy analysis method in ENERGY STAR [58,59], but it is extended to include multiple energy-influencing factors, non-linear input-output relationships and stochasticity. The workflow of this approach is shown in Fig. 6, which is divided into two parts, model fitting and sampling:

- Model fitting: This part fits a Gaussian process (GP) [60] model using the historical HVAC operation data in the old control period. This GP model is treated as the *baseline daily HVAC energy consumption* model. The GP model's form is:

$$GP(\mathbf{x}) = \mathcal{N}(\mu, \sigma), \tag{19}$$

  where $\mathbf{x}$ is the independent variables, μ and σ are the mean and standard deviation of the daily HVAC energy consumption.

  The GP model fitting follows a standard machine learning process, including cross validation for feature selection and model testing. If the testing accuracy (based on the mean of the prediction of the GP model) passes a predefined threshold, the GP model can be used for the sampling part.
- Sampling: This part uses the fitted GP model to create a sampling distribution of the *baseline total HVAC energy consumption at the new control period*. Each sample of the baseline total HVAC energy consumption ($E_{basetotal}$) is generated using the following equation:

$$E_{basetotal,j} = \sum_{i}^{m} \left(E_{basedaily,i} \sim GP(\mathbf{x}_i)\right),$$
$$j \in \{1, 2, \ldots, n\}, \tag{20}$$

  where $E_{basedaily}$ is the *baseline daily HVAC energy consumption at the new control period* sampled from the fitted GP model, i represents a day in the new control period of m days, j represents a sample of the n baseline samples.

  After sampling for n times using Eq. (20), a set of n values (i.e., $\{E_{basetotal,1}, E_{basetotal,2}, \ldots, E_{basetotal,n}\}$) is obtained representing the sampling distribution of the *baseline total HVAC energy consumption at the new control period*. Thus, the distribution functions can be approximated, and it can be compared with the *observed total HVAC energy consumption at the new control period* to create a statistically solid energy saving conclusion.

3. Implementation of the case study

BEM-DRL control framework is implemented in a real-life office building for its radiant heating system. This section describes the case study building and the implementation details.

3.1. Building and system

The case study building, the Intelligent Workplace (IW, shown in Fig. 7), is a one-level 600 m^2 office building in Pittsburgh, PA, USA. It was built in 1997 on the roof of an existing building. The building has about 20 regular occupants and a 30-person conference room.

The major heating system of IW is a novel water-based radiant heating system called "Mullion" system. It integrates the hot water pipes with the window mullions, as shown in Figs. 8 and 9.

3.1.1. Baseline control logic

Fig. 10 shows the existing control logic and the major sensor and actuator points of the Mullion system. With a constant hot

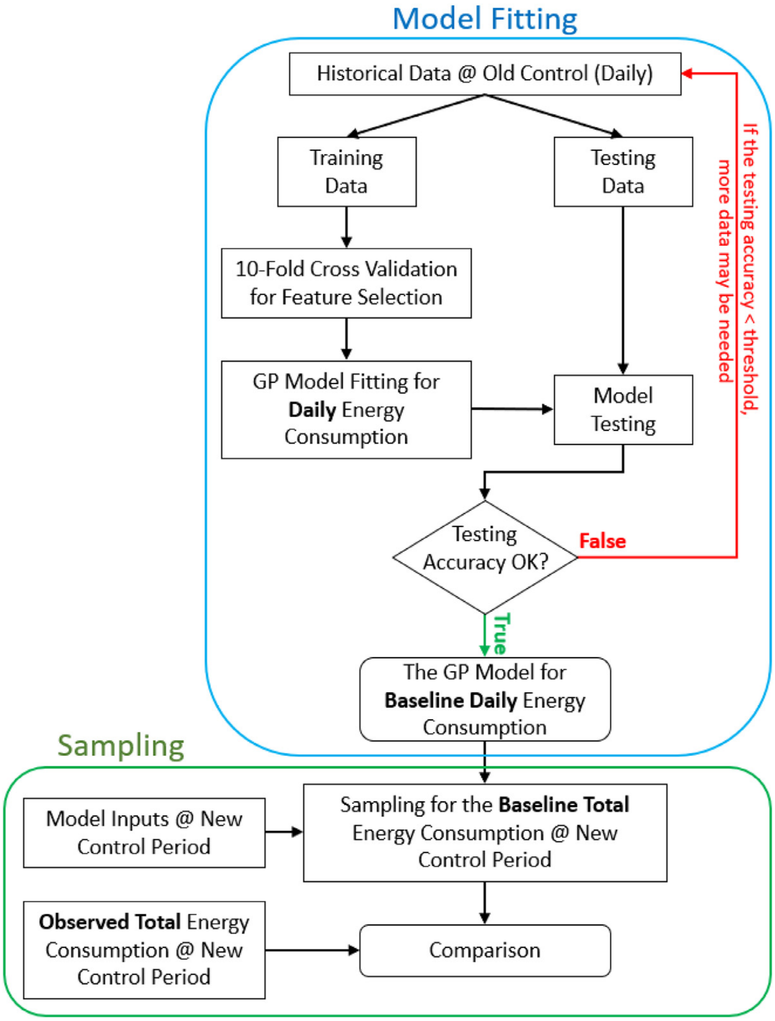
Model Fitting
Historical Data @ Old Control (Daily)
Training Data
Testing Data
10-Fold Cross Validation for Feature Selection
GP Model Fitting for Daily Energy Consumption
Model Testing
Testing Accuracy OK?
False
If the testing accuracy < threshold, more data may be needed
True
The GP Model for Baseline Daily Energy Consumption
Sampling
Model Inputs @ New Control Period
Sampling for the Baseline Total Energy Consumption @ New Control Period
Observed Total Energy Consumption @ New Control Period
Comparison

Fig. 6. Workflow of the normalized energy saving performance evaluation approach.

Fig. 7. The Intelligent Workplace (IW).

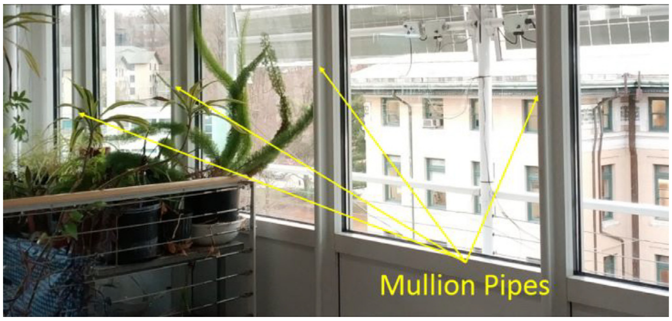

Fig. 8. “Mullion” radiant heating system in IW.

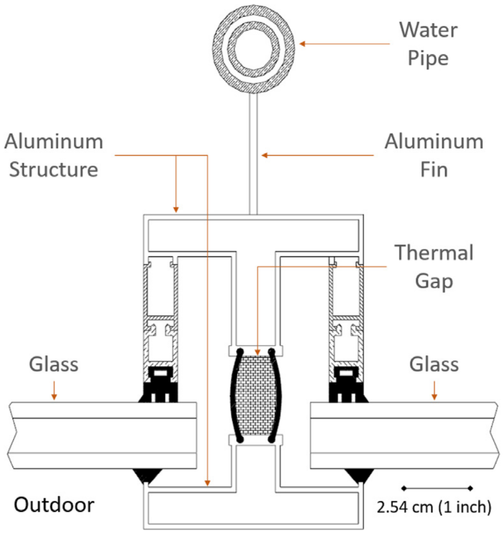

Fig. 9. Details of the “Mullion” radiant heating system in IW (top view) [61].

water flow rate, the Mullion system adjusts its supply hot water temperature to respond to different indoor heating demands. A proportional-integral-derivative (PID) feedback controller (PID1) calculates the Mullion supply water temperature setpoint (SP2)

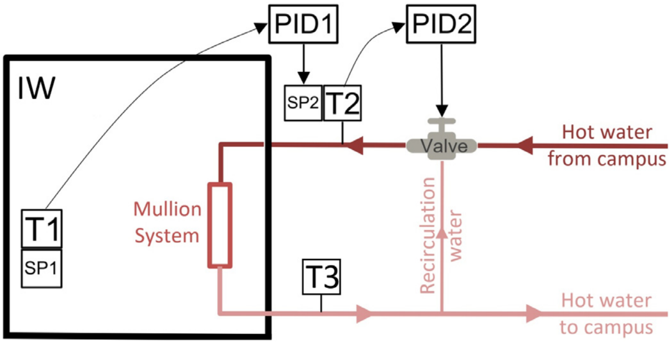

Fig. 10. The existing control principle of the heating system in IW [46].

based on the error between the IW average indoor air temperature (T1) and its setpoint (SP1). Then, another PID controller (PID2) adjusts the open state of a mixture valve based on the error between the Mullion supply water temperature (T2) and its setpoint (SP2). The open state of the mixture valve determines the mixture ratio between the hot water from the campus and the recirculation water, so the Mullion supply water temperature (T2) can be changed. In addition, the control logic shuts off the flow of the hot water from the campus when the outdoor air temperature is below 10 °C.

3.1.2. Energy metric

Since the hot water is from a district heating system of the campus, the facility manager of IW uses the system's heating demand as the energy metric, which is calculated by:

$$Q_{Mull} = C_p \dot{m}(T2 - T3), \tag{21}$$

where C_p is the specific heat of water at constant pressure, $\dot{m}$ is the mass flow rate of the supply water of the Mullion system. Note that this equation does not consider the transient behaviors of the system so its output may not be accurate when there is a sudden change in the system operation (e.g., when the supply water temperature suddenly increases). This may cause the large variability in the heating demand data.

3.2. Optimal control objective

This case study aims to develop an optimal control policy to reduce the Mullion system heating demand and maintain an acceptable overall thermal comfort quality. In the DRL training, the thermal comfort metric is the calculated predicted percentage of dissatisfied (PPD) based on Fanger's model [62]. Even though PPD may not represent the actual thermal comfort profile of the IW occupants, it is still used for the training because of the unavailability of the thermal comfort data. In the deployment, the overall thermal comfort quality is obtained based on the real-time feedback from the IW occupants.

3.3. Building energy modeling of IW

This is the first step of BEM-DRL control framework, where the IW BEM is created mainly using EnergyPlus v8.3.

3.3.1. Model structure

As specified in Eq. (1), the BEM should take some control actions as the input and output the energy performance metrics and comfort performance metrics.

Based on the optimal control objective of this case study, the BEM has three types of output, including:

1. The Mullion heating demand (HeatingDemand) as the energy performance metric;
2. The average indoor air temperature (IAT) as one comfort metric;
3. The average PPD as another comfort metric.

There are two supervisory setpoints that are available as the control input, including:

- SP1: Average indoor air temperature setpoint;
- SP2: Mullion supply water temperature setpoint.

The selection of the control point depends on the BEM calibration results, which will be described below.

Therefore, two BEMs with different control input are built, including *SP1Model*:

$$SP1Model : SP1_t \rightarrow \{HeatingDemand, IAT, PPD\}_t, \tag{22}$$

and *SP2Model*:

$$SP2Model : SP2_t \rightarrow \{HeatingDemand, IAT, PPD\}_t. \tag{23}$$

The major challenge of creating *SP1Model* and *SP2Model* is the PID controllers in this system. As shown in Fig. 10, two PID controllers are involved in the heating control, including one (PID1) determines the Mullion supply water temperature setpoint and the other one (PID2) determines the mixture valve open state. However, PID controllers cannot be directly modeled in EnergyPlus because the simulation time step is much longer than the PID physical sampling time step. Direct modeling of PID controllers in EnergyPlus may result in unstable and unrealistic control outputs.

In addition, the mixture valve cannot be explicitly modeled in EnergyPlus. In EnergyPlus, users can only define the outlet water temperature setpoint and the simulation engine delivers the water at the setpoint temperature if the capacity allows. In the real-world situation, the actual water temperature cannot be exactly at the setpoint value due to the non-ideal water mixture and the instability caused by the PID controller.

Therefore, a workaround method is proposed to implement *SP1Model* and *SP2Model*, as shown below:

$$SP1Model(SP1) = EplusModel\big(PIDModel(\mathbf{PIDx})\big), \quad \text{s.t.} SP1 \in \mathbf{PIDx}, \tag{24}$$

and

$$SP2Model(SP2) = EplusModel(T2), \tag{25}$$

where

$$PIDModel : \mathbf{PIDx}_t \rightarrow T2_t, \tag{26}$$

$$EplusModel : T2_t \rightarrow \{HeatingDemand, IAT, PPD\}_t, \tag{27}$$

In this method, *PIDModel* is a data-driven model outside EnergyPlus to mimic the functions of the PID controllers (will be described in Section 3.3.3). It models the relationship between the relevant observable inputs (**PIDx**) and the observed Mullion supply water temperature (T2). *EplusModel* is an EnergyPlus model that uses the observed Mullion supply water temperature (T2) to predict the energy performance metric and comfort performance metrics. Note that the effective input for *SP2Model* is T2, rather than its setpoint (SP2), because of the inability of EnergyPlus to model an imperfect mixture valve. This may affect the actual deployment of the RL control policy because we can only control SP2 rather than T2. The effect can be minimized by using a long control time step (e.g. 15-min) since T2 can be close to SP2 given a sufficient settling time period.

*3.3.2. EnergyPlus modeling (*EplusModel*)*

The IW building envelope, thermal zones and the heating system are modeled in EnergyPlus. However, the Mullion system cannot be directly modeled because of its uniqueness. As a

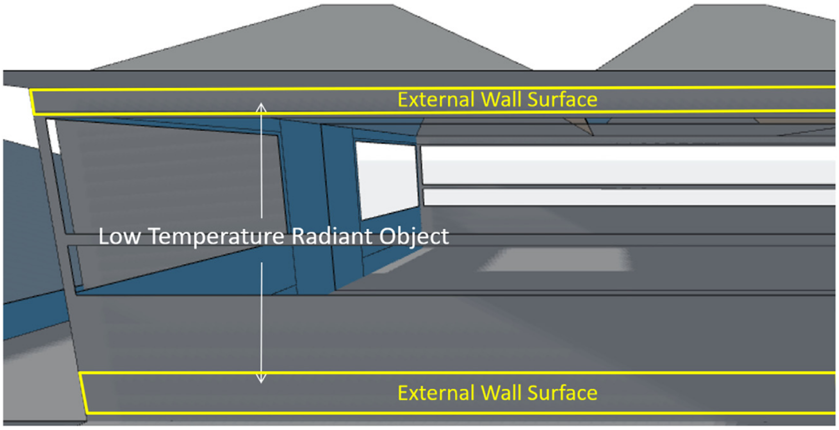

Fig. 11. Modeling the mullion system as the external wall surfaces with the "Low Temperature Radiant" objects of EnergyPlus (rendered by BuildSimHub[63]).

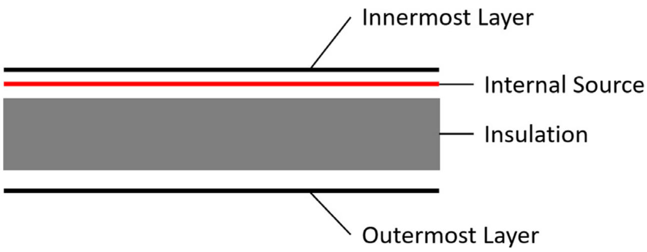

Fig. 12. Cross-section of the mullion radiant surface (Outermost layer is closed to the outdoor environment).

workaround,"low temperature radiant" object of EnergyPlus is selected to model this system. However, it should be noted that the "low temperature radiant" object is designed for modeling the radiant surfaces with heavy thermal mass, such as concrete underfloor heating.

The Mullion system modeled in EnergyPlus is schematically shown in Fig. 11. The top and bottom surfaces of the external walls are modeled as the "low temperature radiant" objects of EnergyPlus (named "Mullion radiant surface" in the later sections). Fig. 12 shows the cross-section of the Mullion radiant surface, where the internal source is an abstraction of the hot water pipes in EnergyPlus simulation. The location of the Mullion radiant surfaces (in this case, the top and the bottom of the external wall) does not affect the simulation results because EnergyPlus (and most other BEM simulation engines) assumes well-mixed air and 1-D heat conduction across envelopes.

As a workaround for the Mullion system modeling, the characteristics of the Mullion radiant surface is significantly different from the actual Mullion system. The modeling parameters related to the Mullion radiant surface (such as the radiant surface area and insulation R-value) cannot be determined directly. These parameters will be found in the calibration section to ensure the simulated thermal behavior matches the observation.

3.3.3. *PID controller modeling (*PIDModel*)*

Random Forest [64] is used to create *PIDModel*. Since the PID controllers include "if-then-else" rules, decision-tree-based Random Forest regression is suitable for modeling such input-output relationships.

The *PIDModel* input includes:

$$\mathbf{PIDx_t} = \{OAT_t, (SP1_t - T1_t), OCCP_t\}, \tag{28}$$

where *OAT* is the outdoor air temperature, *T*1 and *SP*1 are the average indoor air temperature and its setpoint, *OCCP* is the scheduled occupancy flag. The inputs are determined based on the actual control inputs of the PID controllers.

3.4. *Model calibration of IW BEM:*SP2Model

This is the first part of the second step of BEM-DRL control framework. The calibration methods described in Section 2.2 is implemented for the IW EnergyPlus model (equivalently the *SP2Model*). This model takes T2 as the control input to predict the energy and comfort performance metrics.

3.4.1. *Calibration objective*

The calibration objective is to minimize the gap between the simulated and observed Mullion heating demand and average indoor air temperature at sub-hourly resolution. Note that, even though PPD is one of the comfort performance metrics defined in Section 3.3.1, it cannot be calibrated due to the unavailability of the related data.

3.4.2. *Selection of the calibration parameters*

Initially, 12 calibration parameters are manually selected based on the authors' judgment, including: (1) insulation thickness of the pitched roof, (2) insulation thickness of the flat roof, (3) insulation thickness of the external wall, (4) infiltration rate, (5) U-value of the external window, (6) solar heat gain coefficient (SHGC) of the external window, (7) heat conductivity of the innermost layer of the Mullion radiant surface, (8) insulation thickness of the Mullion radiant surface, (9) total area of the Mullion radiant surfaces, (10) air mixing rate across the zones, (11) electric equipment power density, (12) internal mass area. We select 4 calibration parameters out of the 12 using the sensitivity analysis (Morris method [65]) and manual trial-and-error tests. The selected calibration parameters are listed in Table 1 with their calibration ranges. The ranges are determined based on the experience. Note that the four selected calibration parameters are not entirely determined based on the sensitivity analysis. With the sensitivity analysis results as a reference, a number of manual experiments are also conducted.

3.4.3. *Calibration datasets*

This study focuses on the control for heating seasons. Therefore, the calibration is conducted using the three-month observed data from Jan 1st, 2017 to Mar 31th, 2017. The calibrated model is then tested using the one-month observed data from No. 1st, 2017 to No. 30th, 2017. The time resolution of all datasets is 5 min. Table 2 shows the items contained in the datasets.

Table 1
Selected four calibration parameters [50].

Parameter	Range
Insulation (Polyisocyanates) thickness of the Mullion radiant surfaces†	1–5 mm
Total area of the Mullion radiant surfaces†	6.7–26.7% of the external wall
Internal mass area*	200–1000 m^2/zone
Infiltration rate	0.01–0.30 ACH

Note: †The Mullion system is modeled as the “low temperature radiant” surfaces in the EnergyPlus model. *Internal mass is modeled in the EnergyPlus model as 5 cm thick concrete.

Table 2
Items in the calibration datasets for *SP2Model*.

Item	Type
Outdoor air temperature (°C)	Inputs to *SP2Model*
Outdoor air relative humidity (%)	
Direct solar radiation (W/m^2)	
Diffuse solar radiation (W/m^2)	
Wind speed (m/s)	
Wind direction (degree from North)	
Mullion supply water temperature (T2) (°C)	
Mullion supply water mass flow rate (kg/s)	
Average indoor air temperature (°C)	Calibration objectives
Mullion system heating demand (kW)	

3.4.4. Implementation of Bayesian calibration

We implement the Bayesian calibration method shown in Section 2.2.1 for the EnergyPlus model. The input parameters (x of Eq. (2a)) are shown in Table 2 (“Inputs to *SP2Model*” of the table) and the calibration parameters (t of Eq. (2a)) are shown in Table 1. The two calibration objectives are combined into one using the convex combination method shown in Eq. (4a). y_1, y_2 of Eq. (4a) are the observed average indoor air temperature and observed Mullion system heating demand, and η_1, η_2 of Eq. (4a) are the simulated average indoor air temperature and simulated Mullion system heating demand. Different combinations of μ_1 and μ_2 are tested.

3.4.5. Implementation of genetic algorithm

The genetic algorithm method shown in Section 2.2.1 is implemented for the EnergyPlus model. The input parameters (x of Eq. (5a)) and the calibration parameters (t of Eq. (5a)) are the same as the Bayesian calibration. The ranges of the calibration parameters (shown Table 1) are discretized as the feasible set T of t. Two error functions (c_n of Eq. (5a)) are defined, including 5-min CVRMSE for the average indoor air temperature as c_1 and the hourly NMBE for the heating demand as c_2. The two error functions are determined based on the trial-and-error tests. The multi-objective optimization setting can generate the Pareto-front of the solutions. A solution is selected if both objectives have acceptable errors.

3.4.6. Bayesian calibration results

As discussed in Section 3.4.4, two calibration objectives are convexly combined into one with the weights μ_1 (for the average indoor air temperature) and μ_2 (for the heating demand). Different combinations of the weights are tested with the modeling errors shown in Table 3.

It is interesting to find that the weights combination $\mu_1 = 0, \mu_2 = 1$ (i.e., calibrate solely on the heating demand data) gives the obviously better results than the other combinations in terms of all the error metrics. A slight different combination (e.g. $\mu_1 = 0.2, \mu_2 = 0.8$) obviously deteriorates the results. This is possibly because there is a strong correlation between the heating demand and the average indoor air temperature. Therefore, the accurate prediction of the heating demand leads to the accurate prediction of the average indoor air temperature. Such strong correlation is shown in Fig. 13, where an almost linear relationship (Pearson r $= -0.98$) can be found in a specific region. However, this cannot explain why the combination $\mu_1 = 1, \mu_2 = 0$ (i.e., calibrate solely on the indoor air temperature data) cannot achieve the similar performance as the combination $\mu_1 = 0, \mu_2 = 1$. Therefore, the proposed convex combination method cannot achieve the goal of the multi-objective calibration in this case. The convex combination of the two objectives may cause significant information loss that fails the proposed method. Further study should be conducted for multi-objective Bayesian calibration of BEM.

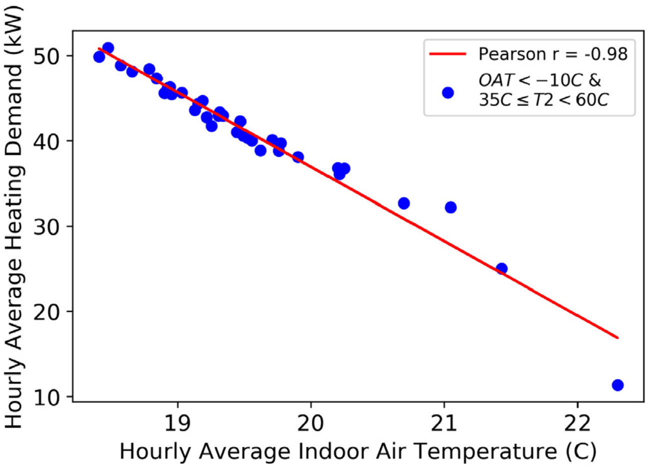

Fig. 13. Relationship between the average indoor air temperature and the heating demand when the outdoor air temperature (OAT) < 10 °C and 35 °C ≤ the mullion supply water temperature (T2) < 60 °C.

Another observation in Table 3 is that the average indoor air temperature can achieve much better accuracy than the heating demand for all different combinations of the weights. This may be because the indoor air temperature has much less variability than the heating demand. It is shown in Table 4 that, the 5-min average indoor air temperature has much smaller Coefficient of Variation (CV) than the hourly heating demand.

Table 5 shows the testing modeling errors of the best calibrated model (with $\mu_1 = 0, \mu_2 = 1$) of the Bayesian calibration. The testing dataset is a unseen dataset that is different from the one used for the Bayesian calibration (as specified in Section 3.4.3). It can be found that, while the indoor air temperature still has low errors, the heating demand errors obviously increase. This may be caused by the system operation change of the Mullion system (the supply water flow rate has been reduced by 40% since the start of November 2017 after the system maintenance by the facility management). The increased errors on the testing dataset means the calibrated model is slightly over-fitted to the calibration dataset. Out of the 5 error metrics shown in Table 5, hourly CVRMSE of the heating demand has the largest value. Fig. 14 shows the hourly and 5-min comparison between the observed and simulated heating demand using the testing dataset. It can be seen that, even though

Table 3
SP2Model modeling errors on the calibration dataset for the different weights combinations of Bayesian calibration (IAT: indoor air temperature).

Objective	Metric	Weights combinations (μ_1, μ_2)					
		0.0,1.0	0.2,0.8	0.4,0.6	0.6,0.4	0.8,0.2	1.0,0.0
Average IAT (C)	5-min NMBE	0.52%	−4.55%	−2.61%	−1.51%	−1.66%	−1.44%
	5-min CVRMSE	4.82%	7.23%	5.47%	5.09%	5.16	5.09%
Heating Demand (kWh)	Hourly NMBE	0.43%	−13.42%	−14.50%	−11.01%	−11.54%	−11.47%
	Hourly CVRMSE	35.93%	39.23%	39.27%	37.48%	37.82%	37.80%
	Daily CVRMSE	10.46%	18.02%	18.79%	15.53%	16.11%	16.01%

Table 4
Summary statistics of the hourly heating demand and the 5-min average indoor air temperature in the calibration dataset.

	Hourly Heating demand	5-min average Indoor air temperature
Mean	20.10 kW	21.90 °C
Std	19.19 kW	1.69 °C
CV	0.95	0.08

Table 5
Modeling errors after calibration using Bayesian calibration and genetic algorithm optimization (IAT: indoor air temperature).

Objective	Metric	Bayesian calibration		Genetic algorithm	
		Calibration	Testing	Calibration	Testing
Average IAT (C)	5-min NMBE	0.52%	−1.01%	−1.66%	−2.86%
	5-min CVRMSE	4.82%	2.65	4.96%	3.94%
Heating Demand (kWh)	Hourly NMBE	0.43%	−1.66%	−0.24%	−0.68%
	Hourly CVRMSE	35.93%	59.91%	36.98%	61.56%
	Daily CVRMSE	10.46%	12.95	12.22%	13.45%

the calibrated EnergyPlus model can capture the overall trend of the heating demand, it fails to do so for the extremes (e.g. the sudden jumps and falls of the heating demand). As the time interval of the data increases, the simulated data can better match the observations. The phenomenon can be probably explained by the following two reasons. Firstly, the observed heating demand has over-estimated variability because it is calculated using a steady-state specific heat equation without considering the transient behaviors of the system; secondly, the current simulation engine of EnergyPlus cannot adequately model the Mullion system.

3.4.7. Genetic algorithm calibration results

The GA optimization converges after about 120 generations. Table 5 shows the calibration errors and testing errors under "Genetic Algorithm". It is found that the results are similar with the Bayesian calibration. The average indoor air temperature still has small errors. For the heating demand, the hourly CVRMSE has a relatively large error while the hourly NMBE and daily CVRMSE are small. The similar results between the two methods indicate that the large hourly CVRMSE of the heating demand is not caused by the inability of the calibration methods, but caused by the in-

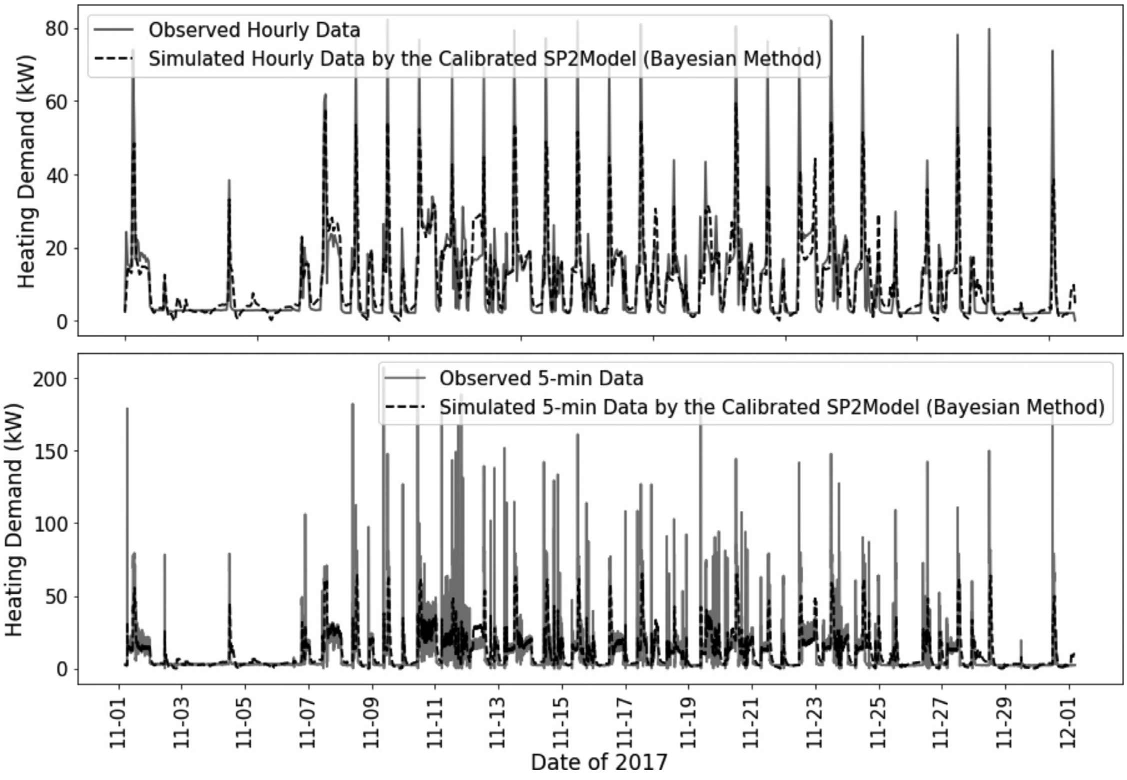

Fig. 14. Hourly and 5-min comparison between the simulated (after Bayesian calibration) and observed heating demand in the testing dataset.

Table 6
Items in the calibration datasets for *PIDModel*.

Item	Type
Outdoor air temperature (°C)	Inputs to *PIDModel*
Average indoor air temperature (T1) (°C)	
Average indoor air temperature setpoint (SP1) (°C)	
Occupancy flag	
Mullion supply water temperature (°C)	Output from *PIDModel*

ability of the BEM engine and the over-estimated variability in the observed data. The large variability in the observed data may be caused by the inappropriate heating demand calculation method (Eq. (21)) which does not consider the transient behavior of the system.

By comparing the calibration results between the Bayesian calibration and Genetic Algorithm (GA) optimization, it is found the GA method shows slightly worse accuracy for the most error metrics except the hourly NMBE of the heating demand. This may be because the GA method only minimizes the explicitly defined objective functions, which is the hourly NMBE of the heating demand and 5-min CVRMSE of the average indoor air temperature. It does not consider any other error metrics that are not in the objective functions. However, it is sometimes difficult to find the suitable objective functions that can fully capture the sub-hourly modeling accuracy of a BEM. In contrast, Bayesian calibration does not have explicit optimization objective functions but calculates the conditional distribution of calibration parameters given the observed building behaviors. However, the Bayesian calibration method of this study cannot be directly used for multi-objective calibration.

Based on the modeling errors, the calibrated EnergyPlus model using the Bayesian method is continuously used for the following sections as the calibrated *SP2Model*.

3.5. Model calibration of IW BEM:SP1Model

This is the second part of the second step of BEM-DRL control framework. This step aims to calibrate *SP1Model*. This model takes SP1 as the control input to predict the energy and comfort performance metrics.

3.5.1. Calibration objective

SP1Model is consisted of *PIDModel* and *EplusModel*, as shown in Eq. (24). Since *EplusModel* has already been calibrated in the previous section, only *PIDModel* is left for calibration. The calibration objective of *PIDModel* is to minimize the gap between the modeled and observed Mullion supply water temperature.

After *PIDModel* is calibrated, *SP1Model* can be formed. *SP1Model* has the same calibration objective as *SP2Model*, which is to minimize the gap between the simulated and observed Mullion heating demand and average indoor air temperature at sub-hourly resolution. Still, PPD is not calibrated because the data is not available.

3.5.2. Calibration dataset for PIDModel

As a data-driven model, *PIDModel* requires a training dataset to train the model, a validation dataset for model tuning, and a testing dataset to evaluate the model. The training dataset is the one-month observed data in Dec 2017, the validation dataset is the one-month observed data in Jan 2018, and the testing dataset is the one-month observed data in No. 2017. The time resolution of all datasets is 5 minute. Table 6 shows the items contained in the datasets.

3.5.3. PIDModel results

To control the over-fitting of *PIDModel*, we use the validation accuracy to select the minimum number of samples at the leaf nodes of the Random Forest model. The model with the highest validation accuracy is used for the prediction on the testing dataset.

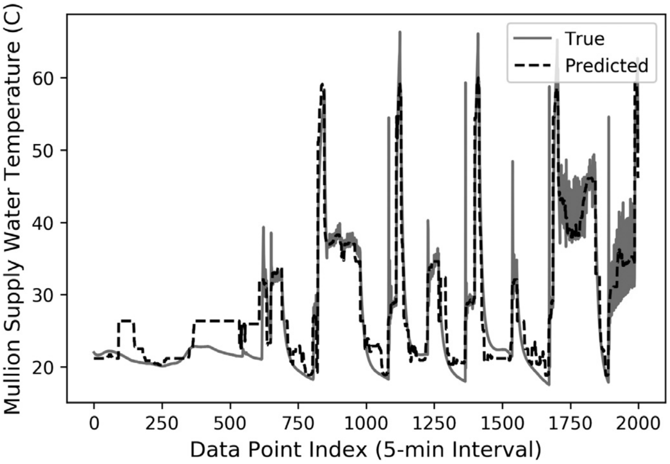

Fig. 15. Selected comparison between the true and predicted mullion supply water temperature of *PIDModel* in the testing dataset.

Table 7
Modeling errors on the testing dataset of *SP1Model* and the best calibrated *SP2Model* (IAT: indoor air temperature).

Objective	Metric	*SP1Model*	*SP2Model*
Average IAT (C)	5-min NMBE	−0.51%	0.52%
	5-min CVRMSE	2.16%	4.82%
Heating Demand (kWh)	Hourly NMBE	−0.17%	0.43%
	Hourly CVRMSE	81.08%	35.93%
	Daily CVRMSE	18.43%	10.46%

The testing accuracy of *PIDModel* is 0.88 ($R^2 = 1$ means the perfect accuracy). Fig. 15 shows the comparison between the true value and the predicted value for the selected data points. It is shown that the model can capture the major trend of the Mullion supply water temperature, but fails to do so for the fluctuations. In addition, the controller model does poor prediction when the Mullion supply water temperature remains low for a long period (as shown in the first half of Fig. 15). This is because this pattern usually occurs for the warms days in winter, which is not common in the training dataset.

3.5.4. SP1Model results

After *PIDModel* is created, it is integrated with the calibrated *EplusModel* to create *SP1Model* (as shown in Eq. (24)).

The model errors are summarized in Table 7. It can be seen that, compared to the best calibrated *SP2Model, SP1Model* has lower errors for the average indoor air temperature but has obviously higher errors for CVRMSE of the heating demand. The lower errors of the indoor air temperature are attributed to the inclusion of *PIDModel*, which makes the indoor air temperature closer to its setpoint. The *SP1Model*'s deteriorated errors in the heating demand is caused by the accumulative effect of the errors from *PIDModel* and *EplusModel*. Firstly, the heating demand prediction of *SP1Model* is influenced by the small errors in the average indoor air temperature prediction of *EplusModel*, as shown in Fig. 16. It can be seen in the figure that the higher-than-measured prediction of the indoor air temperature in *EplusModel* causes the lower-than-measured prediction of the heating demand in *SP1Model*, and vice-versa. This is because, when *EplusModel* drives the indoor air temperature high, *PIDModel* provides lower Mullion supply water temperature to meet the indoor air temperature setpoint, and hence consumes less heating energy. Secondly, as previously discussed,

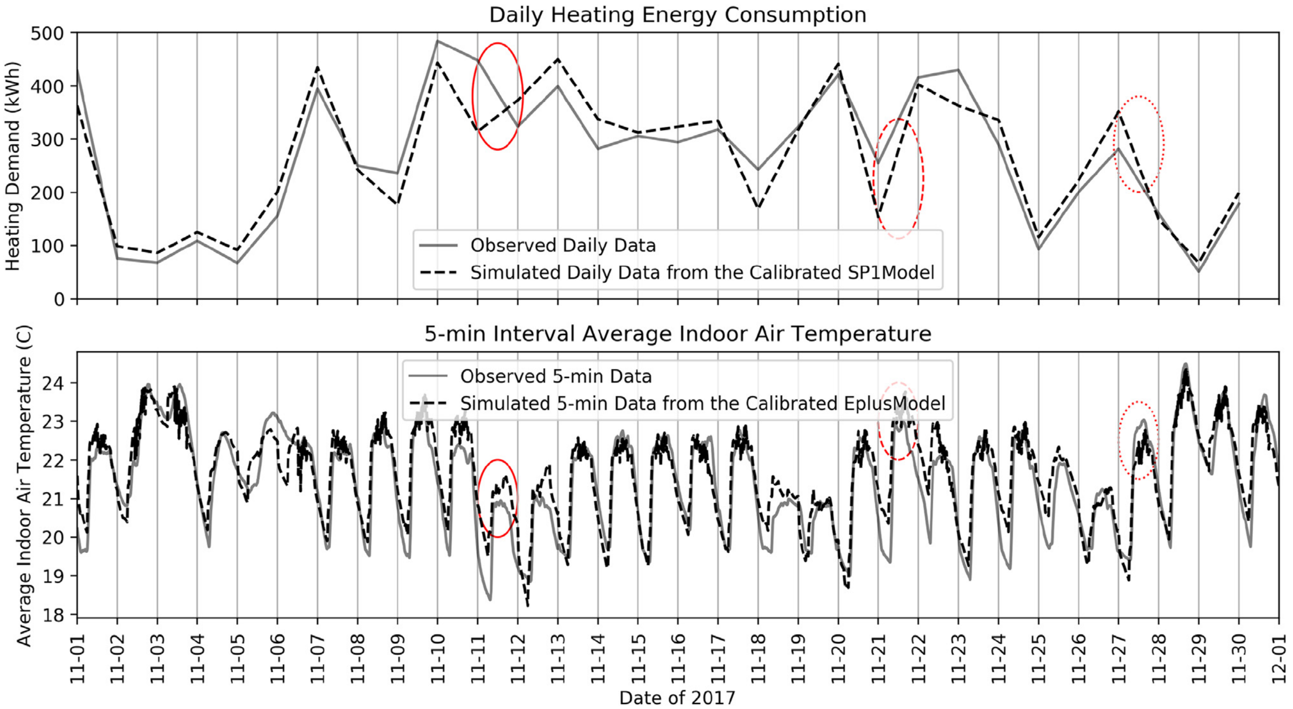

Fig. 16. Daily heating demand comparison of *SP1Model* vs. 5-min average indoor air temperature comparison of *EplusModel* (Red circles show some examples of the relationships between the heating demand prediction error of *SP1Model* and the temperature prediction error of *EplusModel*).

PIDModel cannot capture the fluctuations of the actual Mullion supply water temperature and cannot predict well for the warm days in winter. These errors introduce the additional inaccuracy for the heating demand prediction in *SP1Model*.

3.6. Model calibration of IW BEM:Model selection

Two BEMs, including *SP1Model* and *SP2Model*, are calibrated in the previous section. *SP1Model* achieves the lower errors for the average indoor air temperature but much higher errors for the heating demand. *SP2Model* achieves a better balance for the accuracy of the two performance metrics. Therefore, *SP2Model* is selected for the following section for the DRL training and control deployment. The control point is SP2: Mullion supply water temperature setpoint.

3.7. DRL Training based on the IW BEM (SP2model)

This is the third step of BEM-DRL control framework. The RL agent is trained using the calibrated IW BEM (*SP2Model*) based on the method described in Section 2.3.

As stated in Section 3.2, the optimal control objective is to reduce the heating demand consumption and maintain an acceptable indoor thermal comfort level. In the training, we use PPD as the metric for indoor thermal comfort.

3.7.1. State design

The state follows the structure defined in Eq. (17) and each *ob* consists of the 15 items [46] shown in Table 8.

All items in the state must be between 0 and 1 for the optimization purpose of the deep neural network. Therefore, min-max normalization is used to normalize the data, as shown in Eq. (29). $ob_{\min}$ and $ob_{\max}$ are determined based on the item's physical limits or its expected bounds in the operation. The min-max bounds of the state items are also shown Table 8.

$$ob_{norm} = \frac{ob - ob_{min}}{ob_{max} - ob_{min}} \tag{29}$$

Note that all the state items can be easily accessed through the BAS, except *IW average PPD* which will be replaced by the occupants' real-time thermal comfort feedback during the deployment.

3.7.2. Action design

The action is the Mullion system supply water temperature setpoint (°C). The discrete action space includes the following actions: $\{\text{turn-off heating}, 20^\circ\text{C}, 25^\circ\text{C}, \ldots, 65^\circ\text{C}\}$ [46].

3.7.3. Reward design

A reward function is defined in Eq. (30), that is [46]:

$$R = -\begin{cases} [\tau * ([PPD - 0.1]^+ * \rho)^2 + \beta * Q_{Mull}]_0^1|_{Occp=1} \\ [\tau * [SP_{thres} - T1]^+ * \lambda + \beta * Q_{Mull}]_0^1|_{Occp=0}, \end{cases} \tag{30}$$

where Q_{Mull} is the average Mullion system heating demand since the last control time step (kW), $T1$ is the average indoor air temperature, *Occp* is the occupancy mode flag, and τ, β, ρ, λ, SP_{thres} are the tunable hyperparameters to control the relative weight between the heating demand and indoor thermal comfort. More discussions on the reward design can be found in [46].

3.7.4. Training setup

The DRL training setup is summarized in Table 9. The table summarizes the hyperparameter values resulting from a hyperparameter tuning experiment [50]. The DRL agent is trained using the calibrated IW EnergyPlus model (*SP2Model*) with Pittsburgh TMY3 weather data, and one simulation episode lasts from Jan 1st to Mar 31th. We use TMY3 weather data because that is the only available weather data representing the typical weather conditions. We will then use the actual weather data of 2017 to evaluate the adaptability of the trained RL agent.

3.7.5. Simulation performance results

The performance of DRL training is usually assessed by the training evaluation reward, i.e., the cumulative reward of one simulation episode on the training simulator. This method is valid for

Table 8
State design for the DRL training of IW.

No.	Item	$ob_{\min}$	$ob_{\max}$
1	Day of the week	0	6
2	Hour of the day	0	23
3	Outdoor air temperature (°C)	−13	26
4	Outdoor air relative humidity (%)	0	100
5	Wind speed (m/s)	0	11
6	Wind direction (degree from north)	0	360
7	Diffuse solar radiation (W/m^2)	0	378
8	Direct solar radiation (W/m^2)	0	1000
9	IW steam heat exchanger enable setpoint (°C)*	−30	30
10	IW average PPD	0	100
11	Mullion system supply water temperature setpoint (°C)	20	65
12	IW average indoor air temperature (IAT, °C)	18	25
13	IAT setpoint (°C)	18	25
14	IW occupancy mode flag◇	0	1
15	IW average heating demand since last observation (kW)	0	85

Note: *The outdoor air temperature setpoint below which the IW steam heat exchanger will be enabled. ◇The scheduled occupancy mode flag (the flag is 1 for the period 7:00 a.m.–7:00 p.m. of weekdays and 8:00 a.m.–6:00 p.m. of weekends).

Table 9
DRL training setup for IW.

Item	Value	Item	Value
Neural network width◇	512	A3C local agent number	16
Neural network depth◇	4	Reward discount factor	0.99
Nonlinear activation◇	ReLu	Entropy weight (κ)*	1.0,0.1,0.01,0.001
Optimizer	RMSProp [66]	$\boldsymbol{\kappa}$ decay steps*	[2M, 4M, 6M]
RMSProp decay rate	0.9	Value loss weight	0.5
RMSProp momentum	0.0	RL total interaction times	10 millions
RMSProp epsilon	1e^{-10}	Learning batch size†	5
Learning rate	0.0001	State history window	3
Gradient clip method	By the L2-norm	$\boldsymbol{\tau}$ in the reward	1.0
Gradient clip threshold	5.0	$\boldsymbol{\beta}$ in the reward	2.5
Simulation time step+	5 min	$\boldsymbol{\rho}$ in the reward	10
Control time step+	15 min	$\boldsymbol{\lambda}$ in the reward	5.0

Note: *κ is a staircase decayed constant, and the decay happens at the 2M, 4M and 6M interaction times, e.g., from step 0 to 2M (non-inclusive), $\kappa = 1.0$, from step 2M to 4M, $\kappa = 0.1$, etc; ◇The neural network is the shared fully connected feed-forward neural network in Fig. 3. †This means each local RL agent performs one learning update after 5 interactions with the environment (simulator). +The control time step is sparser than the simulation time step because the slow-response Mullion system needs long time to respond to a new control action.

the classical DRL problems with a single and clear reward metric, e.g. Atari video games which have a single score for the gameplay. However, in this study, the reward is a combination of the heating demand and thermal comfort performance. Therefore, the cumulative reward value has no clear meaning and cannot directly reflect the energy performance and thermal comfort performance of the RL agent.

To better assess the DRL training on the energy and thermal comfort performance, Fig. 17 shows the history of the DRL training evaluation results during the learning, including the cumulative reward, heating demand saving and mean PPD improvement in the training simulator of one simulation episode. It can be found in the figure that:

- The training evaluation reward firstly increases with some fluctuations, and then drops. This means that the RL agent is stuck in some local sub-optimal areas after 7.25M interaction times.
- Similar evaluation reward may result in dramatically different energy and thermal comfort performance. For example, the rewards at the interaction times 5.75M and 6.0M are similar (-2893 at 5.75M and -2751 at 6.0M), but 5.75M has the reduced heating demand with the worse PPD and 6.0M has the increased heating demand with the better PPD. This indicates that this RL environment consists of multiple local sub-optimal areas, which is one of the reasons that make the learning fluctuated and stuck. In addition, this phenomenon indicates that training evaluation reward alone is not enough to fully assess the performance of the DRL training.
- A large improvement in the PPD and a small increase in the heating demand can bring high evaluation reward, such as the results at the interaction times 6.5M, 6.75M and 7.25M. This indicates that the reward function design and the selection of the hyperparameters cannot completely fulfill the control goal of energy efficiency. The current design is more favorable for thermal comfort enhancement.
- Subjective decision is necessary to select a result that achieves obvious heating demand reduction without over deteriorating the thermal comfort. In this study, the RL agent trained at the interaction times 3.25M is selected. It can achieve 7.06% heating demand saving but the mean PPD is increased by 9.64% in the training simulator.

The selected RL agent is then tested in the same IW EnergyPlus model but with the actual weather data of 2017. One simulation episode is Jan 1st-Mar 31th (of 2017) for the testing. The results are shown in Table 10. In the testing simulator, the selected RL agent saves 15% heating demand but the mean PPD is increased (with decreased standard deviation) compared to the baseline control. This result is similar to that in the training simulator, which means the RL agent is robust to the weather change.

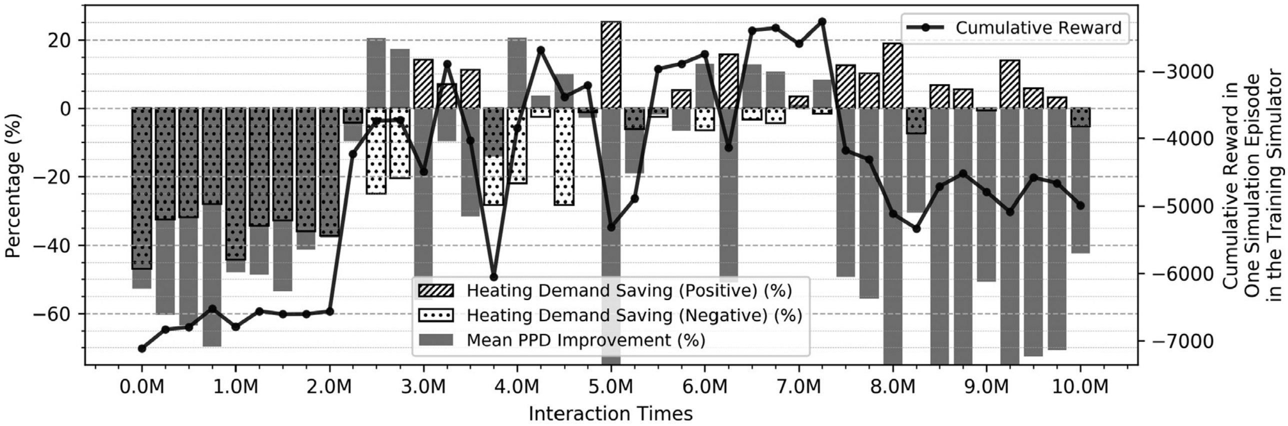

Fig. 17. IW DRL training evaluation results during the learning process (*Note:* 1. the left axis is truncated at −75 for the better presentation; 2. the heating demand saving and mean PPD improvement is derived by comparing with the existing rule-based control, the baseline values are 45,302 kWh and 10.48%).

Table 10
Simulated performance of the selected RL agent in the testing simulator.

	Total heating Demand (kWh)	PPD Mean(%)	PPD Std(%)
Baseline*	43,709	9.46	5.59
DRL control	37,131	11.71	3.76

Note: *The baseline control strategy is the rule-based control strategy used in IW as described in Section 3.1.1.

Table 11
Comparison between the observed total heating demand of DRL control and the GP baseline total heating demand at the 4.99th percentile.

DRL Observed	GP Baseline	Save
24,103 kWh	28,940 kWh	16.7%

3.8. Deployment of the DRL control

This is the fourth step in BEM-DRL control framework. The trained RL agent is deployed in the IW system to control the Mullion supply water temperature setpoint. In this case study, the RL agent is statically deployed without learning going on.

The state design for the deployment is the same as the training, except the PPD is replaced by the real-time thermal comfort feedback from the IW occupants. Details of the implementation setup and the thermal preference feedback system can be found in [46].

3.8.1. Normalized energy saving performance evaluation results

The real-life deployment lasts for 78 days, from Feb 6th to Apr 24th of 2018. We follow the normalized energy saving performance evaluation method described in Section 2.5 to quantify the energy saving performance of the RL agent compared to the old rule-based control strategy. This section focuses on analyzing the results, and the details about the implementation can be found in [46].

The GP baseline model is built based on the data of 357 days in the rule-based control period. The baseline model has the inputs (**x** in Eq. (19)) including outdoor air temperature, global solar radiation, and indoor air temperature, which are selected by the 10-fold cross validation. The GP baseline model is used to generate 10,000 samples of the daily heating demand over the DRL control deployment period. Fig. 18 shows the 50 random examples of the 10,000 samples. These samples represent the possible daily heating demand if the old rule-based control had still been used over the DRL control deployment period.

The total heating demand of the 10,000 samples is calculated based on Eq. (20), so a sampling distribution is generated for the baseline total heating demand over the DRL control deployment period. The baseline samples of the total heating demand are shown in Fig. 19. It can be found that the samples are generally shaped like a normal distribution but noises are existed. Kernel Density Estimation (KDE) is then applied with Gaussian kernel to generate the approximated probability density function (PDF). The PDF represents the distribution of the baseline total heating demand. To make the energy comparison statistically solid, we select the baseline heating demand at around the 5th percentile of the PDF, which is 28,940 kWh at the 4.99th percentile as shown in Fig. 19. This indicates the baseline heating demand is higher than this value with more than 95% probability. By comparing this value with the observed total heating demand, it is concluded that the DRL control achieves 16.7% heating demand reduction compared to the rule-based control, as summarized in Table 11.

4. Discussion

Whole building energy model (BEM) is a detailed physics-based model to predict a building's thermal and energy behaviors. Compared to data-driven models, BEM has better generality because it is based on fundamental physical principles. For example, for a new unseen control strategy, a BEM can accurately predict its energy performance while a data-driven model may not. This is because the data-driven model is developed using a training dataset that has no information about the unseen control strategy.

BEM has been widely used for building design decision support, but it also has strong potential for HVAC optimal control. However, BEM-based control is facing a number of theoretical and practical challenges. This is primarily because BEM is a high-order simulation program with relatively slow computational speed. Therefore, this study proposes a reinforcement learning-based framework, where BEM is used as an environment simulator to train a reinforcement learning agent offline. In this way, BEM can be directly used for developing optimal control strategies, and its slow computational speed is no longer an issue. This study has successfully implemented this framework in a real-life heating system, and showed significant energy saving compared to the rule-based control.

This paper presents the implementation process of the control framework in detail. Building energy modeling is the first step of this control framework. However, the case study building has a unique Mullion heating system with two PID controllers ruling its operation, which cannot be directly modeled in the BEM tool En-

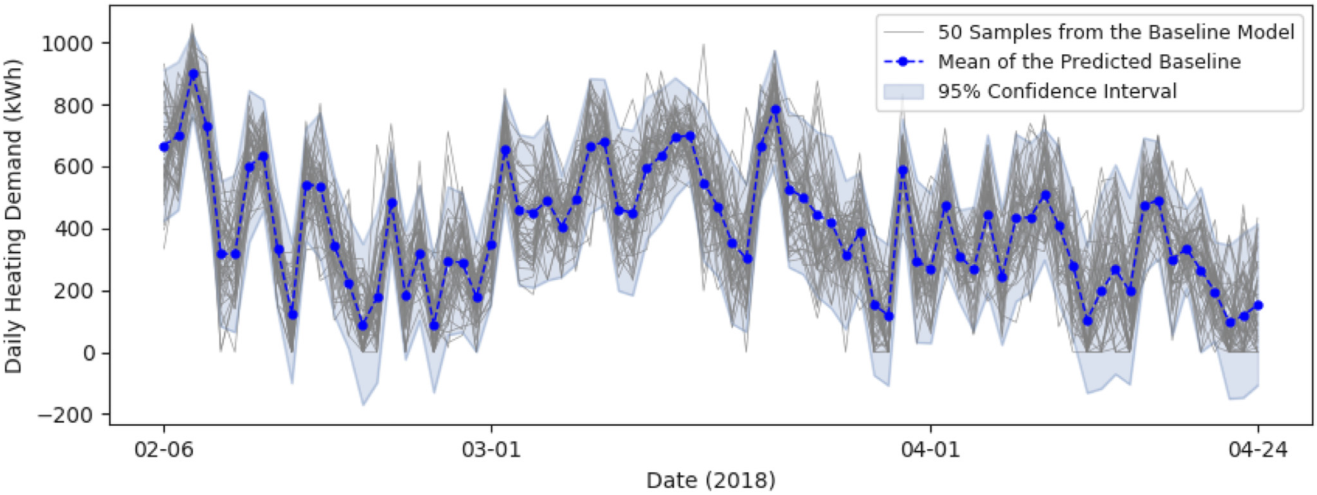

Fig. 18. Baseline daily heating demand samples generated from the GP model (50 out of 10,000 samples are shown).

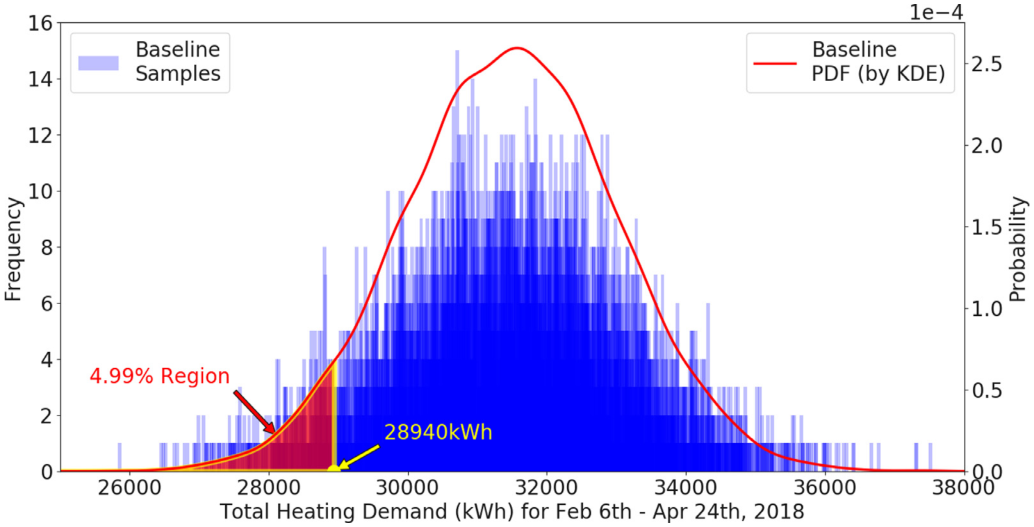

Fig. 19. Baseline total heating demand samples and the estimated probability density function (PDF) generated by Kernel density estimation (KDE) (The shaded area shows the lowest 4.99% of the distribution).

ergyPlus. We propose several workarounds to model the system, but the workarounds make the modeling process complicated and make the model calibration difficult.

Model calibration is the most challenging work in this study. Firstly, model calibration needs high-quality building operation data. Secondly, the BEMs must be calibrated for more than one output at hourly and sub-hourly resolution. Two calibration methods, including Bayesian calibration and Genetic Algorithm optimization, are compared in this study. Both methods achieve a similar level of modeling accuracy, but the Bayesian calibration requires extensive parameter tuning to adopt a single-objective calibration method for the multi-objective calibration problem. Genetic Algorithm optimization can natively support multi-objective calibration, but suitable optimization error functions must be found to represent the sub-hourly modeling errors.

Since the PID controllers cannot be directly modeled in EnergyPlus, we also develop a data-driven PID model and integrate it with the EnergyPlus model of the Mullion system. However, the integrated model has excessive modeling errors for the heating demand prediction due to the accumulative effects of the errors from the two sub-models. The failure of this model rules out the possibility to use the indoor air temperature setpoint as the optimal control point. Therefore, the applicability of this control framework is subjective to the quality of the calibration.

After the calibration, the calibrated BEM still has relatively large modeling error for the heating demand prediction, which may be attributed to the workaround measures to model the unique Mullion system. However, the thresholds for the modeling errors are unknown. In the case study, the RL agent trained in this “inaccurate” BEM still improves the energy efficiency of the target system in the real-life operation. Future study is necessary in this aspect.

The RL agent is then trained by A3C algorithm with a simple reward function which returns a low value if the heating demand is large or the PPD is high. However, we found that this reward design makes the RL agent fluctuate between different local sub-optimal solutions, and the solutions with similar evaluation reward values have dramatically different energy and comfort performance. Therefore, we have to manually select one training result that can achieve obvious energy saving without much sacrifice of the thermal comfort level. Further study is necessary for multi-objective reinforcement learning.

The selected trained RL agent is deployed in real-life and the deployment lasts for about three months. A Gaussian process-based normalized energy saving evaluation method is proposed to use the historical data to create a normalized baseline for the energy comparison. However, this method requires a large amount of historical data to create a solid baseline model, which may not be feasible for poorly-managed buildings.

5. Conclusion and future work

This study proposes a whole building energy model-based deep reinforcement learning control framework. The control framework has four steps: building energy modeling, model calibration, deep reinforcement learning training, and control deployment. The control framework is successfully implemented in an existing novel radiant heating system of an office building. A 78-day deployment test shows that, compared to the old rule-based control, the new control strategy can save 16.7% heating demand with more than 95% probability. Limitations of the control framework are also discussed through the case study, including the inability of the BEM engine, quality of calibration, inappropriate reward design and data dependency of the normalized energy saving evaluation method.

The future work should focus on the following areas:

- Modeling of novel HVAC systems and PID controllers. Current BEM tools cannot directly model novel systems and PID controllers. The problem may be solved by the future extension of EnergyPlus, Spawn of EnergyPlus (SOEP) [67], which integrates a more advanced and flexible modeling language Modelica into EnergyPlus modeling workflow.
- Multi-objective Bayesian calibration with sub-hourly data. The Bayesian calibration method in this study is not suitable for the BEM calibration with multiple outputs, which makes the implementation process complicated. Therefore, future work is necessary to reduce the calibration difficulty.
- The effects of an inaccurate BEM on reinforcement learning control performance. Since BEM calibration is challenging, there is no need to develop an “overly accurate” BEM if it does not bring obvious benefits for the control performance. The future work in this aspect could establish a practical guideline for the level of accuracy that a calibrated BEM should achieve.
- The design of the DRL training, such as the reward function, neural network architecture, multi-objective algorithm, etc. Theses parameters can significantly affect the results of reinforcement learning. A systematic study in this aspect will help future users to select the right design.
- Real-life implementation on different types of HVAC systems. Since different HVAC systems may have dramatically different characteristics, it is necessary to further evaluate the robustness of the control framework via more implementation experiments.
- Adaptability of reinforcement learning. HVAC systems have dynamic characteristics, such as decreasing efficiency due to equipment aging, changing operational schedules, etc. Therefore, it is necessary to study how reinforcement learning can adapt to changing system characteristics.

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.

Acknowledgments

The first author is supported by the China Scholarship Council-Carnegie Mellon University joint Ph.D. fellowship. The building energy model management and 3D visualization are supported by BuildSimHub, Inc. We appreciate the occupants in IW for their cooperation and understanding with the control test. We also appreciate Prof. Mario Berges and Mr. Henning Lange from Carnegie Mellon University for their suggestions on the normalized energy efficiency analysis section. Finally, we would like to thank the Global Communication Center of Carnegie Mellon University for the language support.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.enbuild.2019.07.029.

References

[1] A. Mahdavi, Simulation-based control of building systems operation, Build. Environ. 36 (6) (2001) 789–796 Building and Environmental Performance Simulation: Current State and Future Issues. doi:10.1016/S0360-1323(00)00065-2.

[2] J. Zhao, Design-Build-Operate Energy Information Modeling (DBO-EIM) for Occupant-oriented Predictive Building Control Ph.D. thesis, Carnegie Mellon University, 2015.

[3] E. O'Dwyer, L. De Tommasi, K. Kouramas, M. Cychowski, G. Lightbody, Prioritised objectives for model predictive control of building heating systems, Control Eng. Pract. 63 (March) (2017) 57–68, doi:10.1016/j.conengprac.2017.03.018.

[4] S. Fielsch, T. Grunert, M. Stursberg, A. Kummert, Model predictive control for hydronic heating systems in residential buildings, IFAC-PapersOnLine 50 (1) (2017) 4216–4221, doi:10.1016/j.ifacol.2017.08.817.

[5] Z. Vana, J. Cigler, J. Siroky, E. Zacekova, L. Ferkl, Model-based energy efficient control applied to an office building, J. Process Control 24 (6) (2014) 790–797 Energy Efficient Buildings Special Issue. doi:10.1016/j.jprocont.2014.01.016.

[6] R.D. Coninck, L. Helsen, Practical implementation and evaluation of model predictive control for an office building in brussels, Energy Build. 111 (2016) 290–298, doi:10.1016/j.enbuild.2015.11.014.

[7] H. Huang, L. Chen, E. Hu, A new model predictive control scheme for energy and cost savings in commercial buildings: an airport terminal building case study, Build. Environ. 89 (2015) 203–216, doi:10.1016/j.buildenv.2015.01.037.

[8] S. Li, J. Joe, J. Hu, P. Karava, System identification and model-predictive control of office buildings with integrated photovoltaic-thermal collectors, radiant floor heating and active thermal storage, Solar Energy 113 (2015) 139–157, doi:10.1016/j.solener.2014.11.024.

[9] D. Lindelf, H. Afshari, M. Alisafaee, J. Biswas, M. Caban, X. Mocellin, J. Viaene, Field tests of an adaptive, model-predictive heating controller for residential buildings, Energy Build. 99 (2015) 292–302, doi:10.1016/j.enbuild.2015.04.029.

[10] M. Killian, M. Kozek, Implementation of cooperative fuzzy model predictive control for an energy-efficient office building, Energy Build. 158 (2018) 1404–1416, doi:10.1016/j.enbuild.2017.11.021.

[11] W. Liang, R. Quinte, X. Jia, J.-Q. Sun, MPC Control for improving energy efficiency of a building air handler for multi-zone VAVs, Build. Environ. 92 (2015) 256–268, doi:10.1016/j.buildenv.2015.04.033.

[12] X. Chen, Q. Wang, J. Srebric, Occupant feedback based model predictive control for thermal comfort and energy optimization: a chamber experimental evaluation, Appl. Energy 164 (2016) 341–351, doi:10.1016/j.apenergy.2015.11.065.

[13] Y. Ma, J. Matuško, F. Borrelli, Stochastic model predictive control for building HVAC systems : complexity and conservatism, IEEE Trans. Control Syst. Technol. (2014) 1–16.

[14] M. Razmara, M. Maasoumy, M. Shahbakhti, R.D. Robinett, Optimal exergy control of building HVAC system, Appl. Energy 156 (2015) 555–565, doi:10.1016/j.apenergy.2015.07.051.

[15] J. Shi, N. Yu, W. Yao, Energy efficient building HVAC control algorithm with real-time occupancy prediction, Energy Procedia 111 (September 2016) (2017) 267–276, doi:10.1016/j.egypro.2017.03.028.

[16] J. Ma, S.J. Qin, T. Salsbury, Application of economic mpc to the energy and demand minimization of a commercial building, J. Process Control 24 (8) (2014) 1282–1291 Economic nonlinear model predictive control. doi:10.1016/j.jprocont.2014.06.011.

[17] N. Yu, S. Salakij, R. Chavez, S. Paolucci, M. Sen, P. Antsaklis, Model-based predictive control for building energy management: part II experimental validations, Energy Build. 146 (2017) 19–26, doi:10.1016/j.enbuild.2017.04.027.

[18] 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, doi:10.1016/j.enbuild.2013.12.037.

[19] J. Zhao, K.P. Lam, B.E. Ydstie, O.T. Karaguzel, Energyplus model-based predictive control within design build operate energy information modelling infrastructure, J. Build. Perform. Simul. 8 (3) (2015) 121–134, doi:10.1080/19401493.2014.891656.

[20] Z. Zhang, K.P. Lam, an implementation framework of model predictive control for HVAC systems: a case study of energyplus model-based predictive control, in: ASHRAE 2017 Annual Conference, 2017. Long Island, CA, USA.

[21] F. Ascione, N. Bianco, C. De Stasio, G.M. Mauro, G.P. Vanoli, Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort, Energy Build. 111 (2016) 131–144, doi:10.1016/j.enbuild.2015.11.033.

[22] P. May-Ostendorp, G.P. Henze, C.D. Corbin, B. Rajagopalan, C. Felsmann, Model-predictive control of mixed-mode buildings with rule extraction, Build. Environ. 46 (2) (2011) 428–437, doi:10.1016/j.buildenv.2010.08.004.

[23] M. Aftab, C. Chen, C.K. Chau, T. Rahwan, Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system, Energy Build. 154 (2017) 141–156, doi:10.1016/j.enbuild.2017.07.077.

[24] M. Miezis, D. Jaunzems, N. Stancioff, Predictive control of a building heating system, Energy Procedia 113 (2017) 501–508, doi:10.1016/j.egypro.2017.04.051.

[25] Y. Kwak, J. Huh, C. Jang, Development of a model predictive control framework through real-time building energy management system data, Appl. Energy 155 (2015) 1–13, doi:10.1016/j.apenergy.2015.05.096.

[26] M.C. Mozer, L. Vidmar, R.H. Dodier, The neurothermostat: Predictive optimal control of residential heating systems, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems 9, MIT Press, 1997, pp. 953–959. http://papers.nips.cc/paper/1299-the-neurothermostat-predictive-optimal-control-of-residential-pdf.

[27] S. Liu, G.P. Henze, Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory, J. Sol. Energy Eng. 129 (2) (2007) 215, doi:10.1115/1.2710491.

[28] K. Dalamagkidis, D. Kolokotsa, K. Kalaitzakis, G.S. Stavrakakis, Reinforcement learning for energy conservation and comfort in buildings, Build. Environ. 42 (7) (2007) 2686–2698, doi:10.1016/j.buildenv.2006.07.010.

[29] G.T. Costanzo, S. Iacovella, F. Ruelens, T. Leurs, B.J. Claessens, Experimental analysis of data-driven control for a building heating system, Sustain. Energy, Grids Netw. 6 (2016) 81–90 arXiv:1507.03638, doi:10.1016/j.segan.2016.02.002.

[30] P. Fazenda, K. Veeramachaneni, P. Lima, U.-M. O'Reilly, Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems, J. Ambient Intell. Smart Environ. 6 (6) (2014) 675–690, doi:10.3233/AIS-140288.

[31] A. Nagy, H. Kazmi, F. Cheaib, J. Driesen, Deep Reinforcement Learning for Optimal Control of Space Heating, 2018 arXiv:1805.03777.

[32] J.R. Vzquez-Canteli, S. Ulyanin, J. Kmpf, Z. Nagy, Fusing tensorflow with building energy simulation for intelligent energy management in smart cities, Sustain. Cities Soc. 45 (2019) 243–257, doi:10.1016/j.scs.2018.11.021.

[33] Z. Yu, A. Dexter, Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning, Control Eng. Pract. 18 (5) (2010) 532–539, doi:10.1016/j.conengprac.2010.01.018.

[34] D. Urieli, P. Stone, A Learning Agent for Heat-Pump Thermostat Control, vol. 2, 2013, pp. 1093–1100. Cited By 20. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84899440153&partnerID=40&md5=42dc765943588e8b391dce1c4a79b654.

[35] F. Ruelens, S. Iacovella, B.J. Claessens, R. Belmans, Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning, Energies 8 (8) (2015) 8300–8318, doi:10.3390/en8088300.

[36] D. Li, D. Zhao, Y. Zhu, Z. Xia, Thermal comfort control based on mec algorithm for hvac systems, in: 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–6, doi:10.1109/IJCNN.2015.7280436.

[37] Y. Chen, L.K. Norford, H.W. Samuelson, A. Malkawi, Optimal control of hvac and window systems for natural ventilation through reinforcement learning, Energy Build. 169 (2018) 195–205, doi:10.1016/j.enbuild.2018.03.051.

[38] S. Liu, G.P. Henze, Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: part 2: results and analysis, Energy Build. 38 (2) (2006) 148–161, doi:10.1016/j.enbuild.2005.06.001.

[39] L. Yang, Z. Nagy, P. Goffin, A. Schlueter, Reinforcement learning for optimal control of low exergy buildings, Appl. Energy 156 (2015) 577–586, doi:10.1016/j.apenergy.2015.07.050.

[40] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building hvac control, in: 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC), 2017, pp. 1–6, doi:10.1145/3061639.3062224.

[41] Y. Wang, K. Velswamy, B. Huang, A long-Short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems, Processes 5 (46) (2017), doi:10.3390/pr5030046.

[42] K.S. Peng, C.T. Morrison, Model predictive prior reinforcement learning for a heat pump thermostat, in: IEEE International Conference on Automatic Computing: Feedback Computing, 16, 2016.

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

[44] Y. Li, Y. Wen, K. Guan, D. Tao, Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning, 2017 arXiv:1709.05077v1.

[45] J. Vazquez-Canteli, J. Kampf, Z. Nagy, Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted q-iteration, Energy Procedia 122 (2017) 415–420 CISBAT 2017 International Conference Future Buildings & Districts Energy Efficiency from Nano to Urban Scale. doi:10.1016/j.egypro.2017.07.429.

[46] Z. Zhang, K.P. Lam, Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system, in: Proceedings of the 5th Conference on Systems for Built Environments, in: BuildSys '18, ACM, New York, NY, USA, 2018, pp. 148–157, doi:10.1145/3276774.3276775.

[47] U.S. Department of Energy, EnergyPlus 8.3.0, 2015. Accessed on Jan 18, 2018. https://energyplus.net/.

[48] A. Chong, K. Menberg, Guidelines for the bayesian calibration of building energy models, Energy Build. 174 (2018) 527–547, doi:10.1016/j.enbuild.2018.06.028.

[49] A. Chong, K.P. Lam, M. Pozzi, J. Yang, Bayesian calibration of building energy models with large datasets, Energy Build. 154 (2017) 343–355, doi:10.1016/j.enbuild.2017.08.069.

[50] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, K.P. Lam, A deep reinforcement learning approach to using whole building energy model for HVAC optimal control, in: 2018 Building Performance Analysis Conference and SimBuild, 2018. Chicago, IL, USA.

[51] C.a.C. Coello, G.B. Lamont, D.a.V. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition, second ed., Springer, 2007, doi:10.1007/978-0-387-36797-2.

[52] Guideline 14-2014, Measurement of Energy, Demand, and Water Savings, Standard, American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia, 2014.

[53] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, second ed., MIT Press, Cambridge, MA, USA, 2017.

[54] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.P. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: 33rd International Conference on Machine Learning, vol. 48, 2016. New York, NY, USA . http://arxiv.org/abs/1602.01783.

[55] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, 2016 arXiv:1606.01540, CoRR.

[56] Lawrence Berkeley National Laboratory, Building Controls Virtual Test Bed, 2016. https://simulationresearch.lbl.gov/bcvtb.

[57] A. Chong, W. Xu, S. Chao, N.-T. Ngo, Continuous-time Bayesian calibration of energy models using bim and energy data, Energy Build. 194 (2019) 177–190, doi:10.1016/j.enbuild.2019.04.017.

[58] Portfolio Manager Technical Reference: Climate and Weather, Technical Reference, 2017. Accessed on Aug 27, 2018. https://www.energystar.gov/buildings/tools-and-resources/portfolio-manager-technical-reference-climate-and-weather.

[59] J. Kissock, T. Reddy, D. Claridge, Ambient-temperature regression analysis for estimating retrofit savings in commercial buildings, J. Solar Energy Eng., Trans. ASME 120 (3) (1998) 168–176.

[60] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, USA, 2006.

[61] X. Gong, D.E. Claridge, Study of mullion heating and cooling system of intelligent workplace at carnegie mellon university, 2006. https://www.researchgate.net/publication/237211966_Study_of_Mullion_Heating_and_Cooling_System_of_Intelligent_Workplace_at_Carnegie_Mellon_University.

[62] P.O. Fanger, Thermal Comfort: Analysis and Applications in Environmental Engineering, Danish Technical Press, Copenhagen, Denmark, 1970.

[63] BuildSimHubInc., Buildsimhub, 2018. Accessed on Jan 18, 2018. https://www.buildsim.io/.

[64] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, doi:10.1023/A:1010933404324.

[65] M.D. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics 33 (2) (1991) 161–174, doi:10.1080/00401706.1991.10484804.

[66] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude, COURSERA 4 (2012) 26–31.

[67] Spawn-of-EnergyPlus (SOEP), 2017, Accessed on Sept 9, 2018. https://www.energy.gov/eere/buildings/downloads/spawn-energyplus-soep.