Practical Implementation and Evaluation of Deep Reinforcement Learning Control for a Radiant Heating System

Conference Paper · November 2018

DOI: 10.1145/3276774.3276775

Zhiang Zhang

University of Nottingham Ningbo China

Khee Poh Lam

National University of Singapore

Practical Implementation and Evaluation of Deep Reinforcement Learning Control for a Radiant Heating System

Zhiang Zhang
Carnegie Mellon University
Pittsburgh, PA
zhiangz@andrew.cmu.edu

Khee Poh Lam
Carnegie Mellon University
Pittsburgh, PA
kplam@cmu.edu

ABSTRACT

Deep reinforcement learning (DRL) has become a popular optimal control method in recent years. This is mainly because DRL has the potential to solve the optimal control problems with complex process dynamics, such as the optimal control for heating, ventilation, and air-conditioning (HVAC) systems. However, DRL control for HVAC systems has not been well studied. There is limited research on the real-life implementation and evaluation of this method. This study implements and deploys a DRL control method for a radiant heating system in a real-life office building for energy efficiency. A physics-based model for the heating system is first created and then calibrated using the measured building operation data. After that, the model is used as a simulator to train the DRL agent. The trained agent is then deployed in the actual heating system, and a smartphone App is used to let the occupants submit their thermal preferences to the DRL agent. It is found the DRL control method can save 16.6% to 18.2% heating demand compared to the old rule-based control logic over the three-month deployment period. However, several limitations of this study are found, such as the low participation rate of the App-based thermal preference feedback system, inefficient DRL training, and the requirement for a large amount of building data.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; *Control methods*;

KEYWORDS

HVAC control, deep reinforcement learning, energy efficiency

ACM Reference Format:
Zhiang Zhang and Khee Poh Lam. 2018. Practical Implementation and Evaluation of Deep Reinforcement Learning Control for a Radiant Heating System. In *The 5th ACM International Conference on Systems for Built Environments (BuildSys '18), November 7–8, 2018, Shenzen, China.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3276774.3276775

1 INTRODUCTION

1.1 HVAC Optimal Control

The energy efficiency of heating, ventilation and air-conditioning (HVAC) systems can be significantly affected by their control strategies. Currently, in practice, rule-based control (RBC) is widely used to determine the supervisory level setpoints of HVAC systems, such as various temperature/flow rate setpoints. The "rules" in RBC are usually static and determined based on the experience of engineers and facility managers. Significant energy saving can be potentially achieved if optimal control strategies are used.

The complex dynamics of HVAC systems is one of the major difficulties for developing HVAC optimal control strategies. The most commonly-used method, model predictive control (MPC), requires low-order system dynamics and objective function. Consequently, developing the "model" for MPC is usually a complicated process. Linear models are commonly used to model building temperature response [19, 25, 35], and the control variables must be carefully selected to ensure there is a low-order relationship between the HVAC energy consumption and the state/control variables [8, 15, 30].

1.2 Model-free Reinforcement Learning

Model-free reinforcement learning (RL) is a trial-and-error learning method where the RL agent "learns" the optimal control policy by trying different control actions and observing the consequences. It is becoming popular for HVAC optimal control in recent years partially because of its model-free nature. However, pure model-free on-line learning for HVAC may take too long to converge and may lead to thermal comfort problems [9, 10, 21]. Therefore, most RL-based methods first use a HVAC simulator or historical data to train the RL agent offline and then use the trained RL agent for online control [13, 18, 20, 21, 24, 28, 36–38].

The offline training of reinforcement learning is usually based on a physics-based HVAC simulator [13, 20, 21, 24, 28, 36–38], i.e. a simulator that uses physical principles to predict thermal and energy performance of HVAC systems. This is partially because the RL agent needs to explore different control strategies in the training, and physics-based simulators can provide accurate predictions for historically-unseen control actions. Attempts have been made to use only historical data for reinforcement learning, such as Li et al. [18] propose an offline-trace method. However, the underlying assumption of Li et al.[18]'s method is that the optimal control solution resides in the distribution of the training data. Therefore, part of the study still uses a physics-based simulator to generate the training data under a random control policy, and the author's test based on the real-world data trace has not been evaluated through real-world deployments or calibrated computer simulations.

BuildSys '18, November 7–8, 2018, Shenzen, China

ACM ISBN 978-1-4503-5951-1/18/11…$15.00
https://doi.org/10.1145/3276774.3276775

Deep reinforcement learning (DRL), which uses a deep learning model as the function approximator for the RL agent, becomes popular after Mnih et al. [23] successfully demonstrated its ability to play the Atari video games at the human level. The use of deep learning models can potentially increase the representational capacity of the RL agent, and hence achieve "end-to-end" control, i.e., the DRL agent can use the high-dimensional raw sensor data to determine the optimal control actions. Deep reinforcement learning can also benefit HVAC optimal control, and recent studies show its feasibility and effectiveness through simulations [18, 24, 36, 37].

Even though reinforcement learning has gained popularity for HVAC optimal control in recent years, almost all the existing studies are based on simulations of simple and hypothetical building models (e.g. a single room with a "box" geometry and an independent air conditioner). Since real-world buildings usually have complex geometries and HVAC systems, the simulation results may not be convincing for real-world situations. To the best of our knowledge, the only reported study that deploys RL control in real-life HVAC systems was conducted by Liu and Henze [20]. However, the study uses an over-simplified RL method and the deployment lasted for only one week so the energy saving result may not be statistically solid. DRL control has been deployed in real-life for a domestic hot water heater [16], but it is beyond the scope of this paper.

1.3 Objectives

This study aims to fill the gap of real-life deployments of DRL-based HVAC optimal control. We implement a DRL control method for a radiant heating system and deploy it in a real-life office building for about 3 months. We also incorporate occupant thermal comfort into the DRL control method, which is not common in the existing building optimal control studies [26]. The implementation process is presented, and the energy efficiency performance is analyzed using both simulations and the real-life deployment results. The limitations found in the implementation and deployment are discussed.

2 DEEP REINFORCEMENT LEARNING ALGORITHM

2.1 Standard Reinforcement Learning Problem

In a standard RL problem as shown in Figure 1, a learning agent learns the control policy to maximize the accumulated returned reward from the environment by taking different control actions and observing the resulting environment transitions in a number of discrete steps [31]. The agent-environment interactions of one step can be expressed as a tuple $(S_t, A_t, S_{t+1}, R_{t+1})$, where S_t is the environment's state at time t, A_t is the control action performed by the agent at the time t, S_{t+1} is the resulting environment' state after the agent has taken the action, R_{t+1} is the reward received by the agent from the environment. Ultimately, the goal of reinforcement learning is to learn an optimal control policy $\pi : S_t \longrightarrow A_t$ that maximizes the accumulated future reward $\sum_t^{T_\infty} R_t$.

Three closely-related value functions are used to describe a control policy, including describe how good is a state (state-value function $v_\pi(s)$), how good is an action (action-value function $q_\pi(s,a)$), and how good is an action with respect to the state (advantage function $a_\pi(s,a)$), as shown below:

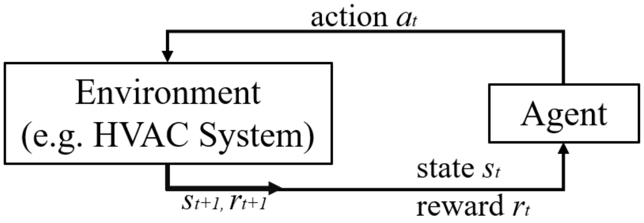

Figure 1: A Standard Reinforcement Learning Problem

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

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\Big], \tag{2}$$

$$a_\pi(s,a) \doteq q_\pi(s,a) - v_\pi(s), \tag{3}$$

where γ is the reward discount factor [31].

The RL algorithm that uses the parameterized functional form to represent the value functions and the control policy is called function approximation, i.e. $v_\pi(s) \approx v_\pi(s, \boldsymbol{\theta})$, $q_\pi(s,a) \approx q_\pi(s,a,\boldsymbol{\theta})$, $\pi(s,a) \approx \pi(s,a,\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is weight vector. If a deep learning model is used as the function approximation, then it is called deep reinforcement learning.

2.2 Asynchronous Advantage Actor Critic

Asynchronous Advantage Actor Critic (A3C) [22], which belongs to policy gradient method [31], is the deep reinforcement learning algorithm used in this study. The goal of the policy gradient method is to learn the parameter $\boldsymbol{\theta}$ in $\pi_{\boldsymbol{\theta}}(s,a) = Pr(a|s, \boldsymbol{\theta})$ that maximizes average reward per time step $J(\boldsymbol{\theta})$:

$$J(\boldsymbol{\theta}) = \sum_s d_{\pi_{\boldsymbol{\theta}}}(s) \sum_a R_s^a \pi_{\boldsymbol{\theta}}(s,a), \tag{4}$$

where $d_{\pi_{\boldsymbol{\theta}}}(s)$ is the 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 reward of the agent at the state s taking the action a. Gradient descent can be used to maximize Equation (4). The gradient of $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is given by:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_s d_{\pi_{\boldsymbol{\theta}}}(s) \sum_a R_s^a \pi_{\boldsymbol{\theta}}(s,a) \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(s,a)}{\pi_{\boldsymbol{\theta}}(s,a)} \tag{5a}$$

$$= \sum_s d_{\pi_{\boldsymbol{\theta}}}(s) \sum_a R_s^a \pi_{\boldsymbol{\theta}}(s,a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a) \tag{5b}$$

$$= \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a) q_{\pi_{\boldsymbol{\theta}}}(s,a)] \tag{5c}$$

$$= \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\big[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a)\big(q_{\pi_{\boldsymbol{\theta}}}(s,a) - v_{\boldsymbol{\theta}_{\boldsymbol{v}}}(s)\big)\big], \tag{5d}$$

where Equation (5c) follows from the policy gradient theorem, (5d) is obtained by subtracting a zero-valued "baseline function" $\mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(s,a) v_{\boldsymbol{\theta}_{\boldsymbol{v}}}(s)]$ to reduce the variance of $q_{\pi_{\boldsymbol{\theta}}}$. The policy gradient in Equation (5d) is called advantage actor critic (A2C).

A3C has only one difference from A2C. Unlike A2C that has only one agent to interact with the environment, A3C fires more

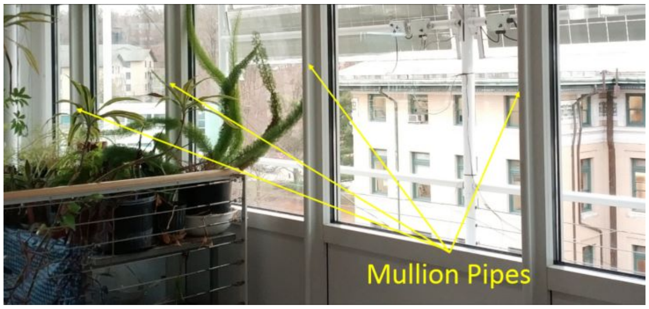

Figure 2: “Mullion" Radiant Heating System in IW

than one agent to interact with the copy of the same environment independently. The A3C agents use gradient descent to update the same state-value function ($v_{\theta_v}(s)$) and policy function ($\pi_\theta(s,a)$) asynchronously. The purpose of this method is to ensure the ($S_t, A_t, S_{t+1}, R_{t+1}$) tuples used to train the RL agent are roughly independent. Compared to the non-asynchronous methods, A3C significantly reduces memory usage and training time. Details of the algorithm can be found in [22].

3 CASE STUDY BUILDING

The case study building, the Intelligent Workplace (IW), is a one-level 600 m^2 office building that is located at Carnegie Mellon University of Pittsburgh, PA, USA. It has about 20 regular occupants and a 30-person conference room. IW uses a water-based radiant heating system and the hot water pipes are integrated with the window mullions (hereafter named “Mullion” system) as shown in Figure 2.

The existing control logic of the Mullion system in heating season is shown in Figure 3. The water flow rate is constant but the supply water temperature is variable for the different heating demands. The mullion supply water temperature setpoint (SP2 in Figure 3) is determined by a proportional-integral-derivative (PID) feedback controller (PID1 in Figure 3) based on the error between the IW average indoor air temperature (T1 in Figure 3) and its setpoint (SP1 in Figure 3). There is another PID feedback controller (PID2 in Figure 3) using the error between the Mullion supply water temperature (T2 in Figure 3) and its setpoint as the input to control the open state of the three-way valve. The three-way valve changes the mixture ratio between the hot water from campus and the recirculation water to change the Mullion supply water temperature.

Since the hot water for the Mullion system is from the district heating system of the campus, the energy metric for the Mullion system is the heating demand calculated by $Q_{Mull} = C_p\dot{m}(T2-T3)$ where C_p is the specific heat of capacity at constant pressure, $\dot{m}$ is the mass flow rate of hot water, T2 and T3 are the Mullion supply and return water temperature respectively.

In this study, an optimal control policy will be developed for the Mullion supply water temperature setpoint (SP2 in Figure 3) to replace the existing control logic.

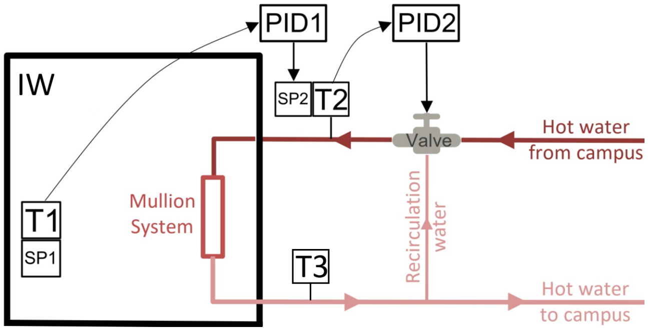

Figure 3: The Existing Heating Control Principle of the Mullion Radiant Heating System in IW

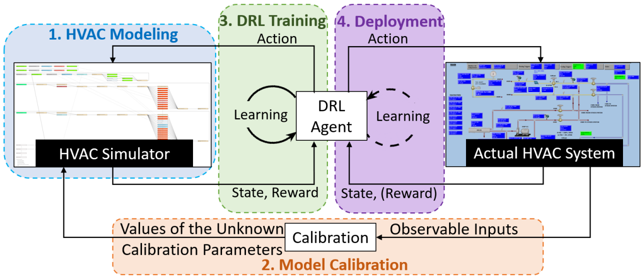

Figure 4: DRL Control Framework for HVAC Systems

4 CONTROL FRAMEWORK

The DRL control framework for the Mullion systems is shown in Figure 4, which includes four steps:

(1) HVAC modeling: The heating system of IW is firstly modeled and the model will be used as the simulator for the DRL training. A physics-based building energy simulation engine, EnergyPlus [34], is used in this step. A calibrated EnergyPlus model can accurately predict the thermal and energy behavior of the heating system for historically-unseen control actions due to its physics-based simulation nature. In addition, creating an EnergyPlus (or other similar simulation engines) model in building design phase is required for compliance for some major green building certifications and standards, such as LEED and ASHRAE. Using EnergyPlus for DRL control can potentially extend the lifecycle of EnergyPlus models from building design to building operation.
(2) Model calibration: The IW model needs to be calibrated using the observable inputs, such as measured weather conditions, indoor conditions, energy consumption, etc, to determine the values of the unknown calibration parameters.
(3) DRL training: The DRL agent is trained off-line using the calibrated IW model as the simulator to develop an optimal control policy. The calibrated EnergyPlus model is wrapped in OpenAI Gym interface [3] using the ExternalInterface function of EnergyPlus and BCVTB middleware [17]. The design of the state, reward and action of DRL is determined

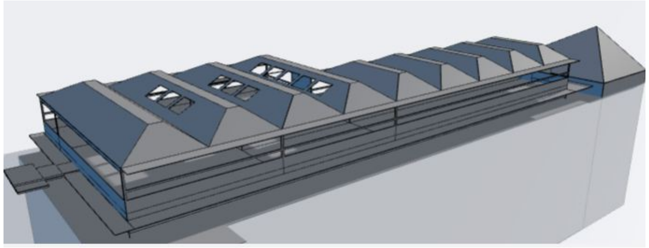

Figure 5: The Geometry Rendering of the IW EnergyPlus Model (rendered by BuildSimHub [4])

based on the building sensor data availability, control optimization objectives and HVAC system control capability.

(4) Deployment: The trained DRL agent is deployed to control the actual heating system in IW. In this step, the DRL agent can either be static (only state is needed for the agent) or continuously learn on-line to refine its control policy based on the actual feedbacks from the heating system (both state and reward are needed for the agent). In this study, the DRL agent is static without on-line learning.

5 IMPLEMENTATION OF THE FRAMEWORK

The framework explained in section 4 is implemented in IW with the goal to reduce the heating demand of the Mullion system while maintaining the acceptable indoor thermal comfort. This paper emphasizes on the third and fourth step of the framework. More details about the first and second step can be found in [39].

5.1 HVAC Modeling

The heating system of IW is modeled using EnergyPlus version 8.3 [34]. The geometry of the model is shown in Figure 5.

5.2 Model Calibration

The IW model is calibrated for the heating energy demand (Q_{Mull}) (kWh) and average indoor air temperature (IAT) (°C). Bayesian calibration using the method proposed by Chong et al.[7] is used for the calibration. More details can be found in [39].

After calibration, the modeling errors are shown in Table 1. The normalized mean bias error (NMBE) and the coefficient of variation of the root mean squared error (CVRMSE) suggested by ASHRAE Guideline 14 [1] are used as the error metrics. 5-minute interval data is used to calculate the error metrics for the indoor average air temperature. It is shown that less than 1% NMBE and less than 5% CVRMSE can be achieved. Hourly and daily data is used to calculate the error metrics for the heating energy demand. It is found that the CVRMSE for the hourly heating energy demand is relatively large, but the hourly NMBE and daily CVRMSE are still within the acceptable range. This means the IW model can only achieve accurate prediction for the aggregated heating demand.

5.3 DRL Training

This part trains an optimal control policy for the heating system based on the calibrated EnergyPlus simulator. The optimal control objective is to minimize the heating demand while maintaining an acceptable indoor thermal comfort level. We use Predicted Percentage of Dissatisfied (PPD) based on Fanger's model [12], rather than indoor air temperature, as the thermal comfort metric because the Mullion system can significantly change the mean radiant temperature of the room to affect the occupant's comfort feelings.

Table 1: IW Modeling Error after Calibration (for Jan 1st-Mar 31th of 2017)

Objective	Results
Average IAT (°C)	5-min NMBE/CVRMSE: 0.52%/4.82%
Q_{Mull} (kWh)	Hourly NMBE/CVRMSE: 0.43%/35.96% Daily CVRMSE: 10.46%

5.3.1 State Design.

The state is what the DRL agent observes for each control step. In this study, the state is a stack of the current and historical observations, as shown below:

$$S = \{ob_t, ob_{t-1}, ..., ob_{t-n}\}, \tag{6}$$

where t is the current control time step, n is the number of the historical control time steps to be considered, and each *ob* consists of the following 15 items: *day of the week, hour of the day, outdoor air temperature (°C), outdoor air relative humidity (%), wind speed (m/s), wind direction (degree from north), diffuse solar radiation (W/m^2), direct solar radiation (W/m^2), IW heating enable outdoor air temperature setpoint (°C)* [1]*, IW average PPD (%), IW Mullion system supply water temperature setpoint (°C), IW average indoor air temperature (°C), IW average indoor air temperature setpoint (°C), IW occupancy status flag*[2]*, IW average heating demand since last time step (kW).* Min-max normalization is used to normalize each item to 0-1.

The state design is determined based on the authors' judgment on the existing BAS data points that are related to the heating system operation. The only item that is not readily available in the BAS is *IW average PPD*, which will be calculated by the EnergyPlus simulator [3] in the training. In the deployment, an approximation method will be used (will be discussed in the later sections).

5.3.2 Action Design.

The action is how the DRL agent controls the environment. In this study, the action is the IW Mullion system supply water temperature setpoint (°C). The action space is discrete as $A = \{off, 20, 25, ..., 65\}$ where off is to turn off the heating.

5.3.3 Reward Design.

The reward design determines the control optimization objective. The reward function combining the Mullion system heating energy demand and the indoor thermal comfort is shown in the Equation (7),

$$R = -\begin{cases} \left[\tau * \left([PPD - 0.1]^{+} * \rho\right)^{2} + \beta * Q_{Mull}\right]_{0}^{1} |_{Occp=1} \\ \left[\tau * [Stpt_{low} - IAT]^{+} * \lambda + \beta * Q_{Mull}\right]_{0}^{1} |_{Occp=0}, \end{cases} \tag{7}$$

where Q_{Mull} is the Mullion system heating demand since last time step (kW), *Occp* is the occupancy status flag, and $\tau, \beta, \rho, \lambda, Stpt_{low}$ are the tunable hyperparameters: τ and β are the weights for the

[1] The heating is enabled if the outdoor air temperature is below this setpoint.

[2] The occupancy status flag is determined based on a fixed schedule: flag = 1 for 7:00 AM-7:00 PM of weekdays & 8:00 AM-6:00 PM of weekends; flag = 0 for all other time.

[3] It is calculated with the assumptions $Clo = 1.0$, $Met = 1.2$ and $v_{air} = 0.137m/s$.

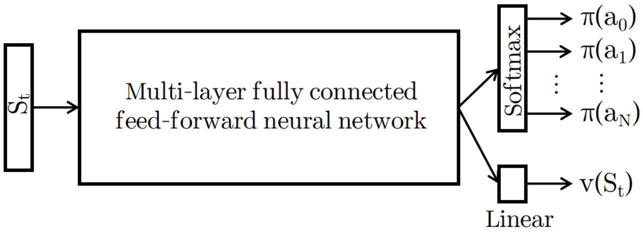

Figure 6: Policy and State-value Function Architecture

Mullion system heating demand and the indoor thermal comfort in the optimization; ρ is a scale factor to penalize large PPD value; λ is the penalty level for the indoor air temperature violation during the unoccupied hours, $Stpt_{low}$ is the indoor air temperature penalty threshold. All parameters are normalized between 0 and 1.

The design of the reward function is empirically determined for better training convergence speed and balance between the indoor thermal comfort and heating demand. Intuitively, $[PPD - 0.1]^+$ and $[Stpt_{low} - IAT]^+$ is to not penalize any PPD values less than 0.1 (0.1 is a recommended threshold value by ASHRAE [2]) or any indoor air temperature higher than $Stpt_{low}$; the square for the PPD term is to penalize large PPD values that are larger than 0.1.

5.3.4 Training setup. The neural network architecture for this study is shown in Figure 6. A shared multi-layer feed-forward neural network is used, and the output from the shared network is fed into a Softmax layer and a linear layer in parallel, where the Softmax layer outputs the control policy distribution (a distribution over all actions) and the linear layer outputs the state-value. The control action is sampled from the control policy distribution.

The shared network in Figure 6 has 4 hidden layers, and each layer has 512 ReLu units. RMSProp [33] is used for optimization with the learning rate 0.0001 and RMSProp decay factor 0.9. The gradients in the back-propagation are clipped with their L2 norm ≤ 5.0. 16 A3C agents are fired to interact with the environment in parallel and the total interactions times is 10M (~ 600K per agent). The history window n in the state function (Equation (6)) is 3.

The entropy of the policy, $H(\pi_\theta(s))$ is used to encourage random exploration of the agent[22]. A hyperparameter, κ, is used to control the exploration level (larger value encourages more exploration). In this study, κ is a piecewise constant with value 1, 0.1, 0.05 and 0.01 for interaction steps before 2M, 4M, 6M and 10M.

The calibrated IW model with TMY3 weather is used for the DRL training in this study. One training episode is three months from Jan 1st to Mar 31th (the heating season). Both the IW model simulation time step and control time step are 5 minutes. The $Stpt_{low}$ in Equation (7) is the actual indoor air temperature setpoint obtained from the IW building automation system (BAS).

5.4 Deployment

The deployment architecture is shown in Figure 7. As shown in the figure, the DRL agent reads the weather-related state from a database using the Web API, and reads other state and writes control actions from/to the BAS using BACnet protocol.

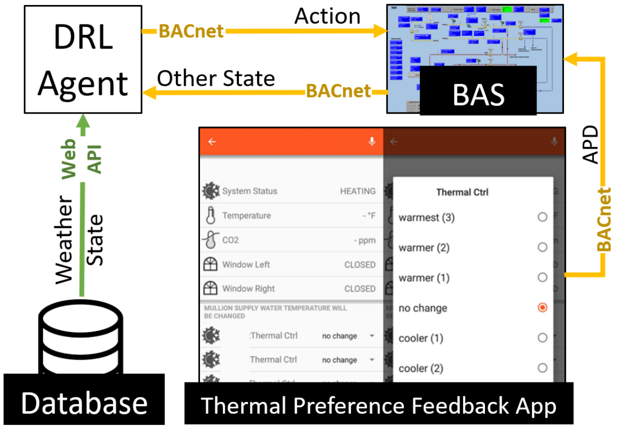

Figure 7: Deployment Architecture

As discussed in section 5.3.1, PPD is in the state as a thermal comfort metric in the training. However, it is difficult to calculate PPD in the deployment because of the lack of sensor to measure the mean radiant temperature. In addition, PPD is calculated using Fanger's model [12] and may not represent the actual thermal comfort preference of the occupants in IW. Therefore, openHAB [32] (a smart phone App) is used to collect the thermal preferences from the occupants to approximate the the Actual Percentage of Dissatisfied (APD) in the room to replace the PPD in the state (with the IRB consent). The user interface of the App is shown in Figure 7. Every IW regular occupant can use the App to submit their thermal preferences (how the occupant wants the thermal environment to change), which include 7 choices: "warmest", "warmer (2)", "warmer (1)", "no change", "cooler (1)", "cooler (2)", "coolest" (the number is used to differentiate different levels of warmness or coolness). Each of the 7 choices is mapped to an integer from -3 to 3 in the App's back-end, where "warmest" is mapped to -3 and "coolest" is mapped to 3. In addition, the choice that an occupant selects will retain in the system for 1 hour before the choice is automatically changed to "no change" if the occupant does not select a new preference. This assumes that the occupants feel satisfied with the thermal environment if they do not use the App to submit a new preference. The App is connected with the BAS using BACnet.

Algorithm 1 shows the method to approximate the APD using the feedbacks from the App, where $List_{thmPref}$ is a list of the thermal preferences from all the occupants collected by the App. One practical problem is that (will be discussed in more detail in the next section), the participation rate of the App is so low that most collected responses are "no change". Therefore, predicted mean vote (PMV)[4] is used as the substitute for the occupants' actual vote if the thermal preference choice in the App is "no change" (the corresponding value is zero) to avoid over-cooling of the space. This mechanism may overwrite the occupants' actual intention for "no change". But given the low participation rate of the App, this

[4] PMV is calculated using Fanger's model [12] with the measured average indoor air temperature, the measured indoor average air relative humidity, a constant Clo value 0.8, a constant MET value 1.2, a constant air speed 0.1 m/s, and the indoor average air temperature minus 1 °C as the mean radiant temperature.

was not an issue in our experiments. Note that the absolute value of the thermal preference and PMV is used in Algorithm 1. This is to approximate the thermal dissatisfaction level of the occupants.

Algorithm 1 APD Approximation

```
procedure GETAPD($List_{thmPref}$, PMV)
    APD = 0
    for thmPref in $List_{thmPref}$ do
        if thmPref == 0 then   APD += |PMV|
        else   APD += |thmPref|
    APD /= 3*NumberOfPeople
    return APD
```

Algorithm 2 Sequence of Operation for the DRL Agent

```
 1: procedure DOCTRL(ctrlTimeStep)
 2:     nextCtrlTime = getCurrentTime()
 3:     while True do
 4:         curTime = getCurrentTime()
 5:         if curTime ≥ nextCtrlTime then
 6:             nextCtrlTime = curTime + ctrlTimeStep
 7:             $List_{thmPref}$, PMV, otherState = readAllState()
 8:             APDRawAvg = getAverage(|$List_{thmPref}$|)
 9:                             ▷ Calculate the APD using Algorithm 1
10:             APD = getAPD($List_{thmPref}$, PMV)
11:             doControlAction(APD, otherState)
12:         else            ▷ when the occupants' preferences change
13:             $List_{thmPref}$ = getThmPrefState()
14:             if getAverage(|$List_{thmPref}$|) > APDRawAvg then
15:                 nextCtrlTime = getCurrentTime()
```

The DRL control signal is discrete with a fixed control time step 5 minutes. Therefore, the DRL agent reads the relevant state data and performs the control action at a fixed frequency. However, the occupants may use the App to submit their thermal preferences at any time and expect the instant change of the system behavior. Therefore, to quickly respond to the occupants' changing thermal preferences, the DRL agent starts a new control time step when it detects the thermal dissatisfaction level increases, as shown in Algorithm 2: Sequence of Operation of the DRL Agent (from line 13 to line 15). *ctrlTimeStep* is the control time step duration.

6 RESULTS

6.1 DRL Training and Simulated Control Performance

The DRL training is performed on a desktop computer with only a 6-core CPU. The training usually takes less than 10 hours to finish. The trained control policy is tested on the calibrated IW model with the actual weather data in 2017 (in contrast to the training that uses TMY3 weather data). The test episode length is 3 months from Jan 1st to Mar 31th. We compare the trained control policy with the existing rule-based control (RBC) of IW to evaluate both energy and thermal comfort performance.

In the training, it is found that the slow thermal response of IW is a major problem for the DRL training convergence. This is also known as "delayed reward problem", which means the control actions taken by the DRL agent cannot take effect immediately in terms of the environment observations. For example, in a cold winter morning, a 1°C increase in the IW average indoor air temperature takes more than 1 hour even though the supply water temperature of the Mullion system has been set to the maximum. The delayed reward problem may cause the DRL agent to be stuck in some local optimal areas during the training.

The hyperparameters in the reward function (Equation 7) are tuned to reduce the effect of the delayed reward problem and balance between the heating energy consumption and thermal comfort. Control action repeat (i.e. repeat the same action for multiple control steps) is also added to mitigate the delayed reward problem.

Table 2 shows the control performance results of the trained DRL agents in the selected experiments of the hyperparameter tuning. The total heating demand (kWh) and the mean and standard deviation of the PPD in the simulation period are used as the evaluation metrics. The control performance of the trained DRL agent is compared with the current RBC strategy of IW. It is interesting to find that the control performance results of the different hyperparameters are not intuitive. For example, we would expect the bigger β (the weight for the heating demand in the reward) and smaller ρ (the penalty scaling factor for the thermal comfort violation in the reward) lead to lower heating demand and worse indoor thermal comfort. However, cases 4, 5 and 6 in Table 2 do not respect that. Such counter-intuitive results are possibly caused by the delayed reward problem that the DRL agents are stuck in some local optimal areas during the training. Out of the six experiments in Table 2, case 6 saves 15.0% of the heating demand with only slightly worse indoor thermal comfort quality in the testing model, which comparably achieves the best balance between the indoor thermal comfort and heating demand. Therefore, the trained agent of case 6 is used for the real-life control deployment.

6.2 Actual Control Performance

The trained DRL agent of case 6 in Table 2 was deployed in the actual system from Feb 6th to Apr 24th of 2018. This section analyzes the energy consumption data and the thermal preferences feedback data during the deployment period.

6.2.1 Direct Energy Comparison with the Historical Data.

Direct comparison with the historical data is the simplest and the most common way in practice to analyze the energy efficiency performance of a new control strategy. However, the result from this approach may not be valid because the distributions of the energy-influencing factors may be changed over the time.

Table 3 shows the comparisons of the heating demand and the major energy influencing factors in two different years. It can be found that the weather and indoor conditions in 2018 are significantly different from that in 2017. This makes the direct comparison of the heating demand meaningless. For example, Table 3 shows the heating demand in Feb 2018 (when the DRL control was deployed) is significantly lower than that in Feb 2017. However, it may be caused

[5] Data from Feb 6th-28th and Apr 1st-24th of 2018 is used for Feb and Apr of 2018.

Table 2: Simulated Control Performance of the Trained DRL Agents for Jan 1st-Mar 31st of 2017 (B: the baseline rule-based control)

#	Hyperparameters		Control Performance		
	Action Repeat	$\tau, \beta, \rho^\dagger$ in Eq. (7)	Total Heating Demand (kWh)	PPD Mean (%)	PPD Std (%)
B	N/A	N/A	43709	9.46	5.59
1	1	1.0, 1.5, 20	47522	8.23	2.46
2	1	1.0, 2.5, 20	39484	11.11	4.53
3	1	1.0, 2.5, 10	37238	14.2	8.65
4	3	1.0, 1.5, 20	38550	10.63	3.34
5	3	1.0, 2.5, 20	39109	10.44	3.75
6	3	1.0, 2.5, 10	37131	11.71	3.76

Note: $\dagger\rho$ is determined by a function $\rho = \frac{1}{PPD_{thres}-10\%}$. $\rho = 20$ and $\rho = 10$ correspond to $PPD_{thres} = 15\%$ and $PPD_{thres} = 20\%$ respectively, meaning the reward function (Equation (7)) returns the minimum value if the PPD exceeds the PPD_{thres}.

by the higher outdoor air temperature, lower wind speed, and lower indoor air temperature in Feb 2018. Another example is that, the heating demand in Apr 2018 is significantly higher than that in Apr 2017, but it may be caused by the significantly colder weather of Apr 2018. Therefore, the heating demand must be normalized by the influencing factors for the meaningful comparisons.

6.2.2 *Normalized Energy Efficiency Analysis.*

A data-driven approach is proposed in this section to conduct the normalized energy efficiency analysis. The approach is inspired by the Weather Normalized Energy method in ENERGY STAR [11], but it is extended to consider multiple variables and perform stochastic analysis. This approach firstly fits a Gaussian process (GP) model using the historical data, and then uses the GP model to "predict" the heating demand for the deployment period if the old rule-based control was still used. After that, the "predicted" heating demand is compared with the observed heating demand to determine the energy efficiency performance of the DRL control method. Gaussian process is chosen because it can give the distribution of its prediction, so the confidence interval of the prediction can be determined.

Model and Dataset Description. The GP model is used to predict the daily total heating demand (kWh) of IW under the old rule-based control. This model will be named "baseline" model in the following sections. The model can be written in the form $GP(\mathbf{x}_i) = \mathcal{N}(\mu_i, \sigma_i)$ where $\mathbf{x}$ is the input, μ and σ are the mean and standard deviation of the predicted daily total heating demand.

The input to the model will be selected from the following 5 features, including the daily average outdoor air temperature (OAT), the daily average global solar radiation (GSR), the daily average wind speed (WS), the daily average indoor air temperature (IAT), and weekday/weekend day type (DAY). Rational Quadratic function is the covariance function for the Gaussian process, that is:

$$Cov(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha l^2}\right)^{-\alpha}, \tag{8}$$

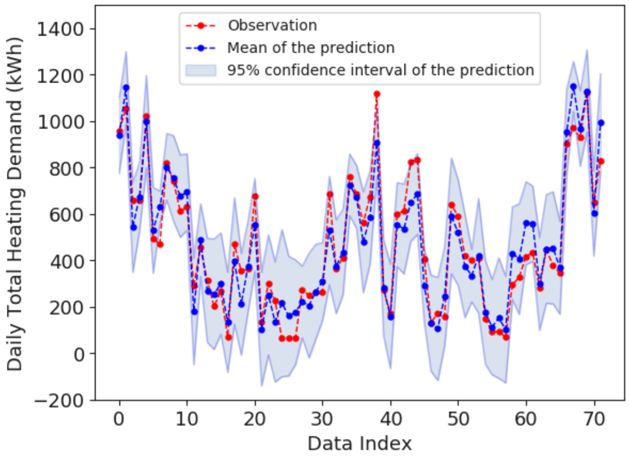

Figure 8: Prediction of the Gaussian Process Model on the Testing Dataset ($R^2 = 0.91$ by the mean of the prediction)

where $\mathbf{x}_i, \mathbf{x}_j$ are two independent inputs, and α, l are the hyperparameters that will be optimized during the training.

The dataset used to train and test the Gaussian process model is from the heating seasons of 2016-2018[6]. Then the dataset is randomly divided into two parts where 80% of the data is used for training and the other 20% is used for testing.

Feature Selection. An exhausting search is performed to find the optimal subset from the 5 candidate features, including OAT, GSR, WS, IAT and DAY. The average R^2 of the ten-fold cross validation (CV) on the training dataset is used as the score to rank the different choices. It is found the choice {OAT, GSR, WS, IAT} and the choice {OAT, GSR, IAT} are ranked 1st and 2nd with very similar R^2. To reduce the model complexity, the second choice with the features OAT, GSR and IAT are used as the inputs to the model.

Model Training and Testing. scikit-learn [27] is used to fit a GP model, and L-BFGS-B [5] is used to optimize the hyper-parameters in the covariance function (Equation (8)). The fitted model is then tested using the testing dataset. The testing R^2 is 0.91 by the mean of the GP model prediction. Figure 8 shows the mean and the 95% confidence interval of the prediction of the GP model on the testing dataset. It can be seen that most observations are inside the 95% confidence interval of the prediction.

Energy Efficiency Analysis. The GP model is used to predict the "baseline" daily total heating demand for the periods that the DRL control is deployed (78 days in total). Figure 9 (left) shows the daily comparison between the observed value and the predicted baseline value. Compared to the median (50th percentile), 25th percentile, 10th percentile and 5th percentile of the predicted baseline, the DRL control can achieve lower energy consumption in 88%, 54%, 27% and 17% of the deployment days. Compared to the mean of the predicted baseline, the total heating demand saving of the DRL control over the deployment period is 23.0%.

[6] The data entries with the daily total heating demand larger than 40 kWh are considered as the heating season data. The total number of entries in the dataset is 357.

Table 3: Comparison of the Daily Average Heating Demand and Daily Average Weather & Indoor Conditions in 2017 and 2018 (RBC: the old rule-based control is used, DRL: the proposed deep reinforcement learning control is used)[5]

		Daily Average Heating Demand (kWh)			**Daily Average Weather & Indoor Conditions**											
					OAT (°C)			GSR (W/m^2)			WS (m/s)			IAT (°C)		
		Feb	Mar	Apr	Feb	Mar	Apr	Feb	Mar	Apr	Feb	Mar	Apr	Feb	Mar	Apr
2017	Mean	436.8	354.7	46.9	4.6	4.9	15.3	92.9	134.0	191.3	1.5	1.5	1.2	22.2	21.5	24.0
(RBC)	Std	301.7	276.4	92.8	7.2	7.4	4.8	45.3	66.3	73.5	0.9	1.1	0.7	1.3	1.6	1.8
2018	Mean	313.5	376.6	217.5	6.1	2.6	8.5	71.5	135.8	178.2	0.8	1.3	1.4	21.6	21.0	22.4
(DRL)	Std	205.4	155.7	141.3	6.5	4.0	6.7	56.1	81.3	80.4	0.5	0.9	1.0	1.3	0.7	2.1

OAT: outdoor air temperature, GSR: global solar radiation, WS: wind speed, IAT: indoor air temperature.

However, Figure 9 (left) shows that most daily heating demand observations are actually within the 95% confidence interval of predicted baseline. Therefore, it is difficult to conclude the energy saving potential of the DRL method. To make further statistical quantification, we use the GP model (the model generates the distribution of the baseline daily heating demand) to sample the distribution of the total heating demand for the 78 DRL deployment days. This means a *sampleNumber* * 78 matrix is generated, where each row is a sample of the 78 days and each col is the total heating demand for each day. The 78-day total heating demand is then calculated for the generated samples. The sampled cumulative distribution (sampled for 5000 times) for the 78-day total heating demand is shown in Figure 9 (right). In this figure, the blue dots show the corresponding points at about 5% and 10% cumulative percentile. It can be found that the DRL control reduces the 78-day total heating demand by 16.6% and 18.2% respectively compared with the predicted baselines at 5% and 10% cumulative percentile.

6.2.3 Thermal Preferences Feedback. This study uses a smart phone App to collect the thermal preferences from the occupants. The biggest problem with this strategy is the low participation rate. The number of times that each occupant uses the App to submit the preferences other than “no change” is shown in Figure 10. It can be seen that there are only 85 total submissions by all the occupants in the 78-day long deployment, and most submissions are for the warmer preferences (“warmer(1)”, “warmer(2)” and “warmest”). There is one person who uses the App much more frequently than others, whose submissions account for more than 40% of the total submissions. Most other occupants submit their thermal preferences less than 5 times. This means for the most of the time, the selected thermal preference on the App stays at “no change”, which is the default selection. However, when the selection on the App is “no change”, there is no way to differentiate whether the occupant really does not need any change on the thermal environment, or the occupant wants the change to the thermal environment but does not use the App to submit. This means the data collected by the App cannot represent the actual thermal comfort level of the occupants over the deployment period.

Figure 11 shows the change of operation of the heating system corresponding to the total vote value for the warmer environment preferences (i.e. the sum of all “warmer(1)”, “warmer(2)” and “warmest” votes which correspond to the value -1, -2 and -3; smaller value means more occupants want warmer environment). Even though the feedback from the App is used to determine the Mullion supply water temperature setpoint, the Mullion return water temperature (M-RWT) is used to represent the Mullion system response. This is because, it may take a while for the hot water to fully charge all the Mullion water pipes so M-RWT can more closely represent the operating status of the heating system. Moreover, most IW occupants sit near the Mullion system, so M-RWT can significantly affect the feeling of warmness of the occupants. It can be found in the figure that the data trend does not follow the expectation of common sense that M-RWT should increase with the smaller total vote value. This is because the DRL agent not only considers the thermal preference feedback, but also considers a number of other factors including outdoor weather conditions, heating demand, etc to make a control decision. Therefore, after the occupants submit a thermal preference for the warmer environment, the operation of the heating system may not change as expected. This may also discourage the occupants from using the App.

7 DISCUSSION

Practical and theoretical limitations are found in the DRL control framework through the real-life implementation experiment.

Firstly, the physics-based modeling of the HVAC system is a tedious process and the quality of the model depends on the experience of the modeler. Building drawings and specifications of HVAC components must be collected, and the correct information is usually hard to get especially for the old buildings. However, the model built in building design phase can be potentially reused for DRL control [40], given EnergyPlus modeling is required in some major green building design standards and certifications.

Secondly, model calibration requires historical data with sub-hourly resolution, and the multi-objective Bayesian calibration method for HVAC physics-based models has not been adequately studied. In addition, even though the calibrated IW model of this study does not achieve high accuracy for the hourly heating demand prediction, the DRL agent trained on this model can still achieve obvious energy saving. The effect of the simulator accuracy on the DRL control should be further studied.

Thirdly, the delayed thermal response of IW causes the convergence problem in the DRL training, so systematic intuitions for the effects of the hyperparameters (e.g. the hyperparameters in the

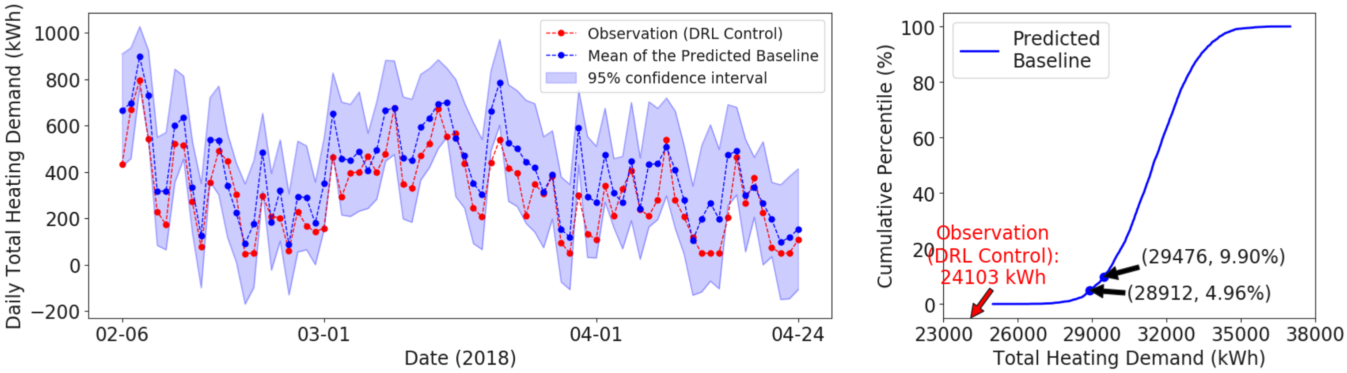

Figure 9: Left: Comparison between the Observed Heating Demand and the Predicted Baseline; Right: Sampled Cumulative Distribution of the Predicted Baseline for the Total Heating Demand over Feb 6th-Apr 24th, 2018

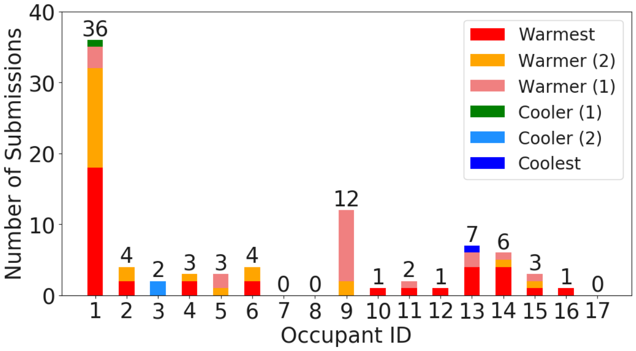

Figure 10: Per-person Usage of the Thermal Preference Feedback App to Submit Other-than ”no change“ Preferences

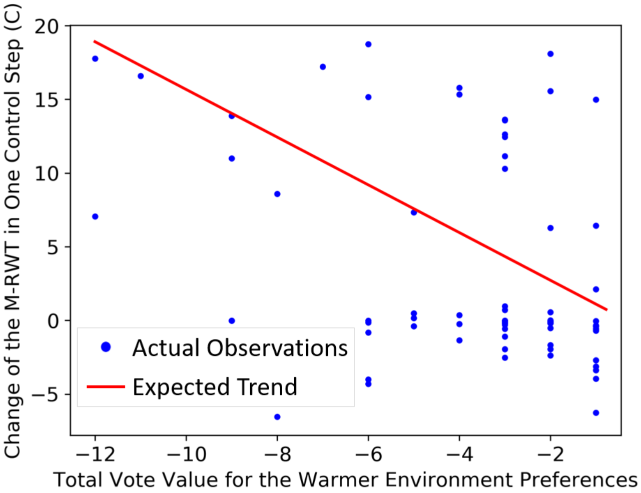

Figure 11: Mullion System Response versus Total Vote Value for the Warmer Environment Preferences

reward function) cannot be obtained. Future work on the design of deep reinforcement learning for HVAC control, such as the deep neural network architecture, state space design, reward function design, etc., is necessary. The fundamental question “Does deep neural network really outperforms other simple function approximations, such as multi-layer perceptron?” must be answered.

Fourthly, the smart phone App based thermal preference feedback system is not effective in practice. First of all, it has very low participation rate so the calculated PMV has to be used as a practical alternative to avoid over-cooling of the space. In addition, the operation of the heating system cannot instantly respond to the occupants' thermal preferences, which may discourage the occupants from using the App and may affect the occupants' psychological feeling of comfort. Moreover, even though the App's UI provides both “warm” and “cool” preference selections, its back-end just uses the absolute value so the “cool” and“warm” selections cannot be differentiated. The design must be improved in the future.

8 CONCLUSION AND FUTURE WORK

This study implements a deep reinforcement learning based optimal control method for a radiant heating system in an office building. A calibrated EnergyPlus model for the heating system is used as the simulator, and A3C is used to train the DRL agent. The trained agent is then deployed in the actual heating system, and is tested for about three months in the heating season. A smart phone App is used to let the occupants submit their thermal preferences, and this information is used by the DRL agent to calculate the control decision. By using a Gaussian process model to generate the baseline heating demand under the old rule-based control logic, it is found there is more than 95% possibility that the DRL control method saves 16.6% heating demand in the deployment period. However, the App-based thermal preference feedback system has very low participation rate, and submitting a thermal preference in the App may not necessarily change the operation of the heating system.

The future work should firstly focus on the delayed reward problem caused by the slow thermal response of the radiant system. The DRL control method should also be compared with other optimal control methods, such as MPC, to further evaluate its effectiveness. Multi-objective Bayesian calibration as well as the effect of the simulator accuracy on the DRL training should also be further studied to facilitate the model calibration process. Moreover, a more effective and non-intrusive way (e.g. through smartwear [6] and computer vision [14, 29]) to collect thermal preference information from occupants should be developed.

ACKNOWLEDGMENTS

The first author would like to thank the China Scholarship Council for the financial support of the Ph.D. fellowship at Carnegie Mellon University. We appreciate the occupants in IW for their cooperation 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. Last but not the least, we would like to thank the reviewers and our shepherd Dr. Bharathan Balaji for the insightful comments.

REFERENCES

[1] American Society of Heating, Refrigerating and Air-Conditioning Engineers 2002. *Guideline 14, Measurement of Energy and Demand Savings.* Guideline.

[2] American Society of Heating, Refrigerating and Air-Conditioning Engineers 2017. *Standard 55, Thermal Environmental Conditions for Human Occupancy.* Standard.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *CoRR* abs/1606.01540 (2016). arXiv:1606.01540

[4] BuildSimHub, Inc. 2018. BuildSimHub. Retrieved January 18, 2018 from https://www.buildsim.io/

[5] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. Sci. Comput.* 16, 5 (Sept. 1995), 1190–1208. https://doi.org/10.1137/0916069

[6] Joon-Ho Choi and Vivian Loftness. 2012. Investigation of human body skin temperatures as a bio-signal to indicate overall thermal sensations. *Building and Environment* 58 (2012), 258 – 269. https://doi.org/10.1016/j.buildenv.2012.07.003

[7] Adrian Chong, Khee Poh Lam, Matteo Pozzi, and Junjing Yang. 2017. Bayesian calibration of building energy models with large datasets. *Energy & Buildings* 154 (2017), 343–355. https://doi.org/10.1016/j.enbuild.2017.08.069

[8] Roel De Coninck and Lieve Helsen. 2016. Practical implementation and evaluation of model predictive control for an office building in Brussels. *Energy and Buildings* 111 (2016), 290–298. https://doi.org/10.1016/j.enbuild.2015.11.014

[9] Giuseppe Tommaso Costanzo, Sandro Iacovella, Frederik Ruelens, Tim Leurs, and Bert J. Claessens. 2016. Experimental analysis of data-driven control for a building heating system. *Sustainable Energy, Grids and Networks* 6 (2016), 81–90. https://doi.org/10.1016/j.segan.2016.02.002 arXiv:1507.03638

[10] Konstantinos Dalamagkidis, Denia Kolokotsa, Konstantinos Kalaitzakis, and George S. Stavrakakis. 2007. Reinforcement learning for energy conservation and comfort in buildings. *Building and Environment* 42, 7 (2007), 2686–2698. https://doi.org/10.1016/j.buildenv.2006.07.010

[11] ENERGY STAR 2017. *Portfolio Manager Technical Reference: Climate and Weather.* Technical Reference. Retrieved August 27, 2018 from https://www.energystar.gov/buildings/tools-and-resources/portfolio-manager-technical-reference-climate-and-weather

[12] Povl Ole Fanger. 1970. *Thermal comfort: analysis and applications in environmental engineering.* Danish Technical Press, Copenhagen, Denmark.

[13] Pedro Fazenda, Kalyan Veeramachaneni, Pedro Lima, and Una-May O'Reilly. 2014. Using Reinforcement Learning to Optimize Occupant Comfort and Energy Usage in HVAC Systems. *Journal of Ambient Intelligence and Smart Environment* 6, 6 (2014), 675–690. https://doi.org/10.3233/AIS-140288

[14] Peter Xiang Gao and S. Keshav. 2013. Optimal Personal Comfort Management Using SPOT+. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys'13).* ACM, New York, NY, USA, Article 22, 8 pages. https://doi.org/10.1145/2528282.2528297

[15] Hao Huang, Lei Chen, and Eric Hu. 2015. A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study. *Building and Environment* 89 (2015), 203 – 216. https://doi.org/10.1016/j.buildenv.2015.01.037

[16] Hussain Kazmi, Fahad Mehmood, Stefan Lodeweyckx, and Johan Driesen. 2018. Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based optimal control of hot water systems. *Energy* 144 (2018), 159 – 168. https://doi.org/10.1016/j.energy.2017.12.019

[17] Lawrence Berkeley National Laboratory. 2016. Building Controls Virtual Test Bed. Retrieved September 25, 2018 from https://simulationresearch.lbl.gov/bcvtb

[18] Yuanlong Li, Yonggang Wen, Kyle Guan, and Dacheng Tao. 2017. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning. *ArXiv e-prints* (Sept. 2017). arXiv:cs.AI/1709.05077

[19] Wei Liang, Rebecca Quinte, Xiaobao Jia, and Jian-Qiao Sun. 2015. MPC control for improving energy efficiency of a building air handler for multi-zone VAVs. *Building and Environment* 92 (2015), 256–268. https://doi.org/10.1016/j.buildenv.2015.04.033

[20] Simeng Liu and Gregor P. Henze. 2006. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis. *Energy and Buildings* 38, 2 (2006), 148–161. https://doi.org/10.1016/j.enbuild.2005.06.001

[21] Simeng Liu and Gregor P. Henze. 2007. Evaluation of Reinforcement Learning for Optimal Control of Building Active and Passive Thermal Storage Inventory. *Journal of Solar Energy Engineering* 129, 2 (2007), 215. https://doi.org/10.1115/1.2710491

[22] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. *ArXiv e-prints* (Feb. 2016). arXiv:1602.01783

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *ArXiv e-prints* (Dec. 2013). arXiv:1312.5602

[24] Adam Nagy, Hussain Kazmi, Farah Cheaib, and Johan Driesen. 2018. Deep Reinforcement Learning for Optimal Control of Space Heating. *ArXiv e-prints* (May 2018). arXiv:stat.AP/1805.03777

[25] Edward O'Dwyer, Luciano De Tommasi, Konstantinos Kouramas, Marcin Cychowski, and Gordon Lightbody. 2017. Prioritised objectives for model predictive control of building heating systems. *Control Engineering Practice* 63, March (2017), 57–68. https://doi.org/10.1016/j.conengprac.2017.03.018

[26] June Young Park and Zoltan Nagy. 2018. Comprehensive analysis of the relationship between thermal comfort and building control research - A data-driven literature review. *Renewable and Sustainable Energy Reviews* 82 (2018), 2664 – 2679. https://doi.org/10.1016/j.rser.2017.09.102

[27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830. http://dl.acm.org/citation.cfm?id=1953048.2078195

[28] Kuo Shiuan Peng and Clayton T. Morrison. 2016. Model Predictive Prior Reinforcement Learning for a Heat Pump Thermostat. In *IEEE International Conference on Automatic Computing: Feedback Computing*, Vol. 16.

[29] Juhi Ranjan and James Scott. 2016. ThermalSense: Determining Dynamic Thermal Comfort Preferences Using Thermographic Imaging. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16).* ACM, New York, NY, USA, 1212–1222. https://doi.org/10.1145/2971648.2971659

[30] Meysam Razmara, Mehdi Maasoumy, Mahdi Shahbakhti, and Rush D. Robinett. 2015. Optimal exergy control of building HVAC system. *Applied Energy* 156 (2015), 555–565. https://doi.org/10.1016/j.apenergy.2015.07.051

[31] Richard S. Sutton and Andrew G. Barto. 2017. *Reinforcement Learning: An Introduction* (second edi ed.). MIT Press, Cambridge, MA, USA.

[32] The openHAB Foundation. 2018. openHAB. Retrieved May 22, 2018 from https://www.openhab.org/

[33] Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude. *COURSERA: Neural Networks for Machine Learning* 4 (2012), 26–31.

[34] U.S. Department of Energy. 2015. EnergyPlus 8.3.0. Retrieved January 18, 2018 from https://energyplus.net/

[35] Zdenek Vana, Jiri Cigler, Jan Siroky, Eva Zacekova, and Lukas Ferkl. 2014. Model-based energy efficient control applied to an office building. *Journal of Process Control* 24, 6 (2014), 790 – 797. https://doi.org/10.1016/j.jprocont.2014.01.016

[36] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. 2017. A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems. *Processes* 5, 46 (2017). https://doi.org/10.3390/pr5030046

[37] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep Reinforcement Learning for Building HVAC Control. In *Proceedings of the 54th Annual Design Automation Conference 2017 (DAC '17).* ACM, New York, NY, USA, Article 22, 6 pages. https://doi.org/10.1145/3061639.3062224

[38] Lei Yang, Zoltan Nagy, Philippe Goffin, and Arno Schlueter. 2015. Reinforcement learning for optimal control of low exergy buildings. *Applied Energy* 156 (2015), 577–586. https://doi.org/10.1016/j.apenergy.2015.07.050

[39] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, Siliang Lu, and Khee Poh Lam. 2018. A Deep Reinforcement Learning Approach to Using Whole Building Energy Model for HVAC Optimal Control. In *2018 Building Performance Analysis Conference and SimBuild.* Chicago, IL, USA.

[40] Jie Zhao, Khee Poh Lam, B. Erik Ydstie, and Omer T. Karaguzel. 2015. EnergyPlus model-based predictive control within design-build-operate energy information modelling infrastructure. *Journal of Building Performance Simulation* 8, 3 (2015), 121–134. https://doi.org/10.1080/19401493.2014.891656