OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control

Xianzhong Ding
University of California, Merced
xding5@ucmerced.edu

Wan Du
University of California, Merced
wdu3@ucmerced.edu

Alberto Cerpa
University of California, Merced
acerpa@ucmerced.edu

ABSTRACT

Recently, significant efforts have been done to improve quality of comfort for commercial buildings' users while also trying to reduce energy use and costs. Most of these efforts have concentrated in energy efficient control of the HVAC (Heating, Ventilation, and Air conditioning) system, which is usually the core system in charge of controlling buildings' conditioning and ventilation. However, in practice, HVAC systems alone cannot control every aspect of conditioning and comfort that affects buildings' occupants. Modern lighting, blind and window systems, usually considered as independent systems, when present, can significantly affect building energy use, and perhaps more importantly, user comfort in terms of thermal, air quality and illumination conditions. For example, it has been shown that a blind system can provide 12%∼35% reduction in cooling load in summer while also improving visual comfort.

In this paper, we take a holistic approach to deal with the trade-offs between energy use and comfort in commercial buildings. We developed a system called OCTOPUS, which employs a novel deep reinforcement learning (DRL) framework that uses a data-driven approach to find the optimal control sequences of *all* building's subsystems, including HVAC, lighting, blind and window systems. The DRL architecture includes a novel reward function that allows the framework to explore the trade-offs between energy use and users' comfort, while at the same time enable the solution of the high-dimensional control problem due to the interactions of four different building subsystems. In order to cope with OCTOPUS's data training requirements, we argue that calibrated simulations that match the target building operational points are the vehicle to generate enough data to be able to train our DRL framework to find the control solution for the target building. In our work, we trained OCTOPUS with 10-year weather data and a building model that is implemented in the EnergyPlus building simulator, which was calibrated using data from a real production building. Through extensive simulations we demonstrate that OCTOPUS can achieve 14.26% and 8.1% energy savings compared with the state-of-the art rule-based method in a LEED Gold Certified building and the latest DRL-based method available in the literature respectively, while maintaining human comfort within a desired range.

BuildSys '19, November 13–14, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7005-9/19/11…$15.00
https://doi.org/10.1145/3360322.3360857

CCS CONCEPTS

• **Computing methodologies** → Control methods.

KEYWORDS

HVAC control, deep reinforcement learning, energy efficiency

ACM Reference Format:
Xianzhong Ding, Wan Du, and Alberto Cerpa. 2019. OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control. In *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360322.3360857

1 INTRODUCTION

Energy saving in buildings is important to society, as buildings consume 32% energy and 51% electricity demand worldwide [21]. Rule-based control (RBC) is widely used to set the actuators (e.g., heating or cooling temperature, and fan speed) in the HVAC (heating, ventilation, and air-conditioning) system. The "rules" in RBC are usually set as some static thresholds or simple control loops based on the experience of engineers and facility managers. The thresholds and simple control rules may not be optimal and have to be adapted to new buildings at commissioning time. Many times these rules are updated in an ad-hoc manner, based on experience and feedback from occupants and/or trial and error performed by HVAC engineers during the operational use of the building. As a result, many model-based approaches have been developed to model the thermal dynamics of a building and execute a control algorithm on top of the model, such as Proportional Integral Derivative (PID) [24] and Model Predictive Control (MPC) [6]. However, the complexity of the thermal dynamics and the various influencing factors are hard to be precisely modeled, which is why the models tend to be simplified in order deal with the parameter-fitting data requirements and computational complexity when solving the optimization problem [6].

To tackle the limitations of the model-based methods, some model-free approaches have been proposed based on reinforcement learning (RL) for HVAC control, including Q-learning [20] and Deep Reinforcement Learning (DRL) [34]. With RL, an optimal control policy can be learned by the trial-and-error interaction between a control agent and a building, without explicitly modeling the system dynamics. By adopting a deep neural network as the control agent, DRL-based schemes can handle large state and action space in building control [25]. Some recent work [30, 34] has shown that DRL can provide real-time control for building energy efficiency. However, all existing methods only consider a single subsystem in buildings, e.g., the HVAC system [30] or the heating system [34], ignoring some other subsystems that can affect performance from the energy use and/or user comfort point of view.

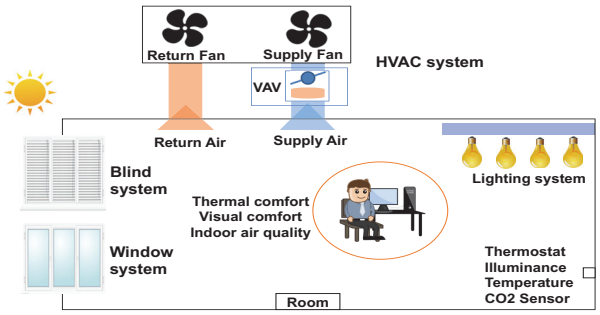

Figure 1: Four Subsystems in a Typical Building.

At present, more and more buildings are been equipped with automatically-adjustable windows and blinds. For example, motor-operated windows and blinds, like the intelligent products from GEZE [2], have been installed using an effective natural ventilation strategy [3]. In addition, researchers have studied the potential of energy saving by jointly controlling the HVAC system and another subsystem, like blind [27], lighting [8], and window [29]. For example, the energy consumed by HVAC can be reduced by 17%~47% if window-based natural ventilation is enabled [29].

In this work, we argue that a *holistic approach* that considers *all available subsystems* (HVAC, blinds, windows, lights) in buildings, which have complex and non-trivial interactions should be used in coordination to achieve a specific energy efficiency/comfort goal. Figure 1 shows a depiction of a modern building that includes multiple subsystems (e.g., HVAC, window, blind and lighting) that work together to guarantee human comfort goals, including thermal comfort, visual comfort, and indoor air quality goals. For example, indoor temperature can be influenced by three subsystems, like setting the HVAC temperature (adjusting the discharge temperature set points at the VAV level), and/or adjusting blind slats (allowing external sunlight to heat indoor air) and/or the window system (enabling exchange of indoor and outdoor air).

To achieve more efficient energy management in buildings, we propose to study the joint control problem of four subsystems of a building to meet three human comfort metrics as depicted in Figure 2. The energy consumption of a building is determined by four subsystems and their interaction. It is challenging to control four subsystems jointly, since they may have opposite outcomes on different human comfort metrics. For example, opening the window can improve indoor air quality and save the energy consumed by the HVAC system for ventilation, but it may also reduce (in winter) or increase (in summer) indoor temperature. To handle the temperature variation caused by the open window, the HVAC system may need to spend more energy rather than the energy saved by natural ventilation.

This paper presents a customized DRL-based control system, named OCTOPUS, which controls four subsystems of a building to meet three human comfort requirements with the best energy efficiency. It leverages all the advantages of DRL-based control, including fast adaptation to new buildings, real-time actuation and being able to handle a large state space. However, to control four subsystems jointly in a unified framework, we need to tackle three main challenges:

High-Dimension Control Actions. With a uniform DRL framework, OCTOPUS needs to decide a control action for four subsystems

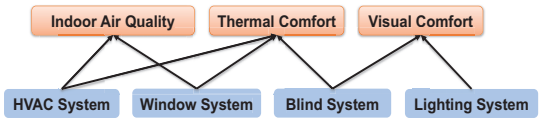

Figure 2: Relationship between Four Subsystems and Three Human Comfort Metrics

jointly and periodically, including the heating/cooling air temperature of the HVAC system, the brightness level of electric lights, the blind slat range and the open proportion of the window. Each subsystem adds one dimension in the action space. The goal of OCTOPUS is to select the best action combination A_s from the set of all possible combinations A_{all} that meet the requirement of human comfort with the lowest energy consumption. Since each subsystem can set its actuator to a large number of discrete values, e.g., we have 66 possible values to set the zone temperature by the HVAC system, the set of all possible action combinations A_{all} is extremely large, i.e., 2,371,842 actions in our case. To solve this problem, we leverage a novel neural architecture featuring a shared representation followed by four network branches, one for each action dimension. In addition, from the shared representation, a state value is obtained that links the joint interrelations in the action space, and it is added to the output of the four previous branches. This approach achieves a linear increase in the number of network outputs by allowing independence for each action dimension.

Reward Function. To explore the potential energy saving energy across four subsystems while considering three human comforts, we formulate this problem into an optimization problem. We define a reward function in our DRL framework to solve the optimization problem. The novel reward function jointly combines energy consumption, thermal comfort, visual comfort, and indoor air quality, offering better control and more flexibility to meet the unique requirement of specific users.

Data Training Requirements. While model-free approaches in general, and RL techniques in particular, are very powerful, their main weakness is the amount of data required to train them properly. The amount of training data should be in proportion to the action space, which in our case it is very large. This issue is very important since we cannot expect building stakeholders to have years of building data readily available so we can use OCTOPUS. Instead, we use a calibrated building simulator combined with weather data that is readily available, in order to generate as much training data as we needed. We trained our OCTOPUS system with 10-year of weather data of two areas; one is Merced, CA, and the other one in Chicago, IL, due to their distinct weather characteristics. The critical point is that this method allows to train OCTOPUS for any building under any weather profile, as long as there is a repository of weather data for the location, and a few months of building data to perform the calibration of the simulator.

We would like to highlight the main contributions of the paper:

- To the best of our knowledge, this is the first work that leverages DRL to balance the tradeoff between energy use and human comfort in a holistic manner.
- OCTOPUS adopts a special reward function and a new DRL architecture to tackle the challenges imposed by the combined joint control of four subsystems with a very large action space.
- We tackle the issue of data training requirement by adopting a simulation strategy for data generation, and spending effort in

calibrating the simulations to make them as close as possible to the target building. This allows our system to generate as much data as needed within a finite amount of time.

2 RELATED WORK

Conventional control of the HVAC system. Model predictive control (MPC) models have been developed for HVAC control. For example, complex models are commonly used to model building temperature response [6]. However, MPC control only works well for low-order system dynamics, and its control variables must be carefully set for different buildings [10].

Conventional control of multiple subsystems. Kolokotsa et al. [19] develop an energy efficient fuzzy controller based on a genetic algorithm to control four subsystems (HVAC, lighting, window, and blind) and meet the occupant requirements of human comfort. However, the genetic algorithm requires a few minutes to hours to generate one control action. It is not practical to be used in real building control.

RL-based control of the HVAC system. Li et al. [20] adopt Q learning for HVAC control. Dalamagkidis et al. [9] design a Linear Reinforcement Learning Controller (LRLC) using linear function approximation of the state-action value function to meet the thermal comfort with minimal energy consumption. However, the tabular Q learning approaches are not suitable for problems with a large state space, like the state of four subsystems.

DRL-based control of the HVAC system. Wei et al. [30] develop a data-driven deep reinforcement learning approach to intelligently learn an effective strategy for HVAC control. Zhang et al. [33, 34] implement and deploy a DRL-based control method for radiant heating systems in a real-life office building. Although the above works can improve the performance of HVAC control, they require discretization of the state-action space and are only focused on HVAC subsystem.

3 MOTIVATION

In this section, we perform a set of preliminary simulations in EnergyPlus [22] in order to understand the relationships between the different subsystems and their impact on human comfort in a building as described in Figure 2. This is also used to gain trust that the simulator is being run correctly, with intuitive results that can be understood. Our goal is to study the effect of different subsystems to three human comfort metrics. A single-floor office building of 100 m^2 at Merced, California is modeled. The building is equipped with a north-facing single-panel window of 2 m^2 and an interior blind. The simulations are conducted with weather data for the month of October. This is a shoulder season, with outdoor temperatures being a bit cold, but mostly sunny days, i.e. high solar gain.

Figure 3 shows the effect of three subsystems on thermal comfort. Predictive Mean Vote (PMV) is used to evaluate thermal comfort. A PMV value that is close to zero represents the best thermal comfort, with higher positive values meaning people are hot, and lower negative values meaning people are cold. A detailed description of PMV values and ranges will be provided in Section 4.4.2. The baseline case (green-solid) in Figure 3 shows the case when all three subsystems are closed. This case acts like a "fishtank" model, where the only effect in the room is due to the solar gain during the day, with no other interactions through any system but the window in the room. When only the blind is open (blue-dashed), the PMV value can be affected from 1.45 to 1.75, showing an increase in the temperature due to the increase of solar gain. This is more prominent in the middle of the day, when the sun is at its apex. When the window is open (red-dashed-dot), the PMV value is lowered due to the temperature effect, colder outside air enters the room, producing a colder, more comfortable temperature. The HVAC system (black-dot) can maintain the PMV value to an acceptable range (between -0.5 and +0.5) by forcing air to be at the correct temperature through the room vents. From the results of Figure 3, we can conclude that all these three subsystems have an obvious impact on thermal comfort. Figure 4 shows the illuminance measured at a place close to the window from 5 am to 7 pm when the blind is open (green-solid) and the room has natural light. Illuminance values from 500-1000 lux or higher are acceptable in most environments. We clearly see that with the blind open, the values are within this range for most of the day. Figure 5 shows the indoor temperature when the blind is open (red-dashed) or closed (blue-solid). The outdoor temperature (green-dash-dot) is lower than the indoor temperature, due to the "fish tank" effect and the lack of window open or an HVAC system on during the day. Combining the results from Figures 4 and 5 we see that the blind system can save the energy consumed by the lighting system by reducing the need of artificial light, but it may also increase the energy used by the HVAC system in order to maintain the load. However, for lower outdoor temperatures in winter, the sunlight through the blind can increase the indoor temperature and save the energy of the HVAC system.

The simulations are conducted to show some examples of the non-trivial interactions between subsystems and human comfort. It is challenging to quantify the complex relationships among different subsystems and the three human comfort metrics and serves as motivation for our work.

4 DESIGN OF OCTOPUS

In this section, we describe in detail the design of OCTOPUS, including a system overview, DRL-based building control, branching dueling Q-Network, and reward function calculation.

4.1 OCTOPUS Overview

The design goal of OCTOPUS is to meet the requirement of human comfort by energy efficient control of four subsystems in a building. Our goal is to minimize the energy E consumed by all subsystems in the building, including the energy used in heating/cooling coils to heat and cool the air, the electricity used in the water pumps and flow fans in the HVAC system, electricity used by the lights, and the electricity used by the motors to adjust the blinds and windows. The value of E is constantly being affected by the vector A_s, which is an action combination for four subsystems, which belongs to the vector A_{all} that is all the possible action combinations.

In addition to the minimization of energy, we would like to maintain the human comfort metrics within a particular range. This can be expressed as $P_{min} \leq PMV \leq P_{max}$, $V_{min} \leq V \leq V_{max}$, and $I_{min} \leq I \leq I_{max}$. PMV is a parameter that measures thermal comfort; V measures visual comfort; and I measures indoor air quality. The consumed energy E and the human comfort metrics

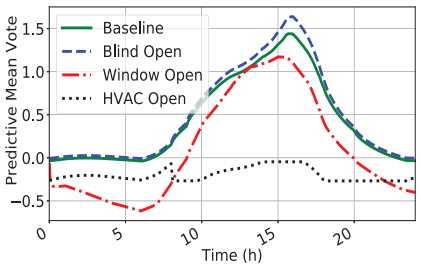

Figure 3: Thermal Comfort, PMV

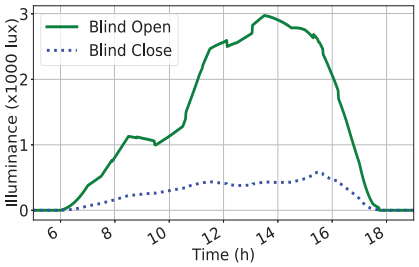

Figure 4: Visual Comfort, Illuminance

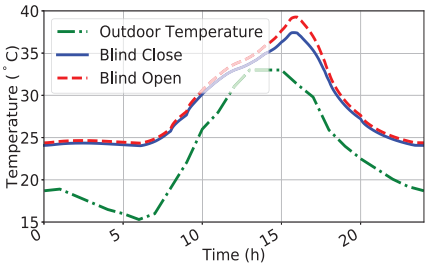

Figure 5: Temperature Effect

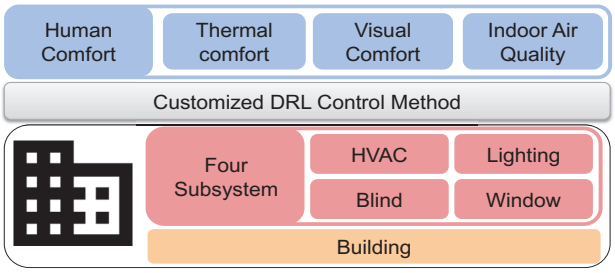

Figure 6: OCTOPUS architecture

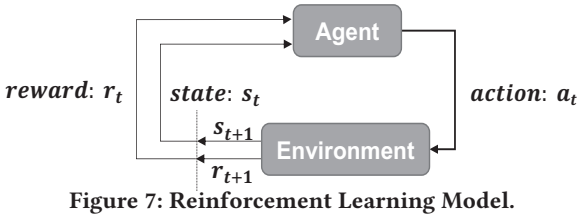

Figure 7: Reinforcement Learning Model.

(PMV, V, and I) are determined by the current state of all four subsystems, the outdoor weather and the action we are about to take. They can be measured in real buildings or calculated in a building simulator, like EnergyPlus, after the action is executed.

The achieved human comfort results should fall into an acceptable range to meet the requirements of users. We use $[P_{min}, P_{max}]$, $[V_{min}, V_{max}]$, $[I_{min}, I_{max}]$ to present the accepted range for thermal comfort, visual comfort and indoor air quality. They can be set by individual users according to their preference, or by facility managers based on building standards. The details on calculation of the above parameters (E, PMV, V and I), the definition of an action (A_s) and the settings of the human comfort ranges (e.g., $[P_{min}, P_{max}]$) will be introduced in Section 4.4.

Our goal is to find the best A_s from A_{all} for each action interval (15 mins in our implementation). The best A_s should maintain the three human comfort metrics in their acceptable ranges for the entire control interval with the lowest energy consumption (E). To achieve this goal, we implement a DRL-based control system for buildings. Figure 6 shows the overview of OCTOPUS as a building control system. It consists of three layers, i.e., building layer, control layer, and user demand layer. The building layer is composed of the real building or a building simulation model, and the sensor data management components. It provides sensor data to the control layer and executes the control actions generated by the latter. The user demand layer quantifies the user requirement of three human comfort metrics. The range of each human comfort metric is then passed to the control layer, which searches for the optimal control to meet the human comfort ranges with minimal energy consumption.

4.2 DRL-based Building Control

4.2.1 Basics for DRL and DQN. In a standard RL framework, as shown in Figure 7, an agent learns an optimal control policy by trying different control actions to the environment. In our case, the environment is a building simulation model due to the extensive data requirements to train the system. With DRL, the agent is implemented as a deep neural network (DNN). 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 time t, S_{t+1} is the resulting environment' s state after the agent has taken the action, R_{t+1} is the reward received by the agent from the environment. The goal of DNN agent training is to learn an optimal control policy to maximize the accumulated returned reward by taking different control actions.

4.2.2 State in OCTOPUS. The state is what the DRL agent takes as input for each control step. In this study, the state is a stack of the current and historical observations, as shown below:

$$S = \{ob_t, ob_{t-1}, ..., ob_{t-n}\}, \quad (1)$$

where t is the current time step, n is the number of the historical time steps to be considered, and each ob consists of the following 15 items: outdoor air temperature (°C), outdoor air relative humidity (%), indoor air temperature(°C), indoor air relative humidity (%), diffuse solar radiation (W/m^2), direct solar radiation (W/m^2), solar incident angle (°), wind speed (m/s), wind direction (degree from north), average PMV (%), heating setpoint of the HVAC system (°C), cooling setpoint of the HVAC system (°C), the dimming level of lights (%), the window open percentage (%), and the blind open angle (°). All the values we can be calculated by the EnergyPlus simulation model. Min-max normalization is used to convert each item to a value within 0-1.

4.2.3 Action in OCTOPUS. The action is how the DRL agent controls the environment. Given the state, we want the agent to find the most suitable action combinations among HVAC, lighting, blind and window system to balance energy consumption and three human comfort metrics. There are four action dimensions when considering these four subsystems, represented as

$$A_t = \{H_t, L_t, B_t, W_t\}, \quad (2)$$

where A_t is the action combination of four subsystems at time t. H_t is the temperature set-point of the HVAC system, which can be set to 66 values. L_t is the dimming level of electric lights. B_t is the blind slat angle. The range of blind slat can be adjusted from $0\,^\circ \sim 180\,^\circ$. W_t is the open percentage of the window. Each of the

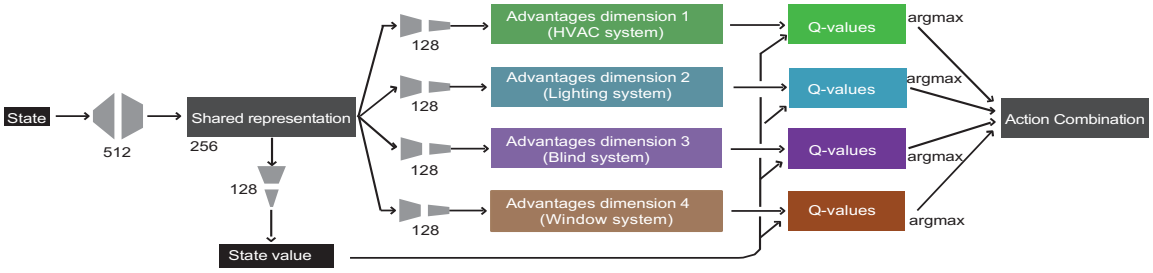

Figure 8: The Specific Action Branching Network Implemented for the Proposed BDQ Agent

above three actuation parameters can be set to 33 values in our current implementation to achieve a proper balance between control granularity and calculation complexity. According to Equation 2, the total number of possible actions in the action space is 2,371,842 ($66 \times 33 \times 33 \times 33$). Existing DRL architectures, like Deep Q-Network (DQN) in [30] and Asynchronous Advantage Actor-Critic (A3C) in [34], cannot work efficiently in our problem, because the large number of actions requires to be explicitly represented in the agent DNN network and it will significantly increase the number of DNN parameters to be learned and consequently the training time [28]. To solve this problem, we leverage a novel neural architecture featuring a shared representation followed by four network branches, one for each action dimension.

4.2.4 Reward Function in OCTOPUS. Reward illustrates the immediate evaluation of the control effects for each action under a certain state. Both human comfort and energy consumption should be incorporated. To define the reward function, a common approach is to use the Lagrangian Multiplier function [17] to first convert the constrained formulation into an unconstrained one:

$$R = -[\rho_1 Norm(E) + \rho_2 Norm(T_c) + \rho_3 Norm(V_c) + \rho_4 Norm(I_c)], \quad (3)$$

where ρ_1, ρ_2, ρ_3 and ρ_4 are the Lagrangian multipliers. E is energy consumption, Tc is thermal comfort, Vc is visual comfort and Ic is Indoor air quality. $Norm(x)$ is a normalization process, i.e., $Norm(x) = (x - x_{min})/(x_{max} - x_{min})$ to transform energy and three human comfort to the same scale. This reward function merges the objective (e.g. energy consumption) and constraint satisfaction (e.g. human comfort). The reward consists of four parts, namely, the penalty for the energy consumption of the HVAC and lighting system, the penalty for the occupants' thermal discomfort, the penalty for the occupants' visual discomfort and the penalty for the occupants' indoor air condition discomfort. Specifically, the reward should be less, if more energy is consumed by the HVAC system or the occupants feel uncomfortable about the building thermal, visual and indoor air condition. The details about how to define and formulate energy consumption E, thermal comfort Tc, visual comfort Vc and indoor air condition Ic are explained in Section 4.4.

4.3 Branching Dueling Q-Network

To solve the high-dimensional action problem described in Section 4.2.3, OCTOPUS adopts a Branching Dueling Q-Network (BDQ), which is a branching variant of the dueling Double Deep Q-Network (DDQN). BDQ is a new neural architecture featuring a shared decision module followed by several network branches, one for each action dimension. BDQ can scale robustly to environments

Algorithm 1: The Training Process of Our BDQ-Based Agent

Input: The range of human comfort metrics and maximum acceptable energy consumption
Output: A trained DRL agent
Initialize BDQ's prediction Q with random weights θ;
Initialize BDQ's target Q^- with weight $\theta^- = \theta$;
for *episode =0,1,...,M* **do**
  Obtain the initial state S_t and A_t randomly;
  for *control time step t = 0,1,...,T* **do**
    Update H_t, L_t, B_t, W_t by the control action, A_t;
    Calculate reward R_{t+1} by Equation 3;
    Obtain current state observation S_{t+1};
    Store $(S_t, A_t, S_{t+1}, R_{t+1})$ in reply memory Λ;
    Draw mini-batch sample transitions from Λ;
    Calculate the target vector and update weights in neural network Q ;
    Update target network $Q_d^-(s, a_d)$ using Equation 5;
    Perform greedy descent iteratively to tune BDQ by Equation 6.

with high dimensional action spaces and even outperform the Deep Deterministic Policy Gradient (DDPG) algorithm in the most challenging task [26]. In our current implementation, we use a simulated building model developed in EnergyPlus as the environment for training and validation. Our BDQ-based agent interacts with the EnergyPlus model. At each control step, it processes the state (building and weather parameters) and generates a combined action set for four subsystems.

Figure 8 demonstrates the action branching network of BDQ agent. When a state is inputted, the shared decision module computes a latent representation that is then used for the calculation of the state value and the output of the network (Advantages dimension in Figure 8) for each dimension branch. The state value and the factorized advantages are then combined, via a special aggregation layer, to output the Q-values for each action dimension. These Q-values are then queried for the generation of a joint-action tuple. The weights of the fully connected neural layers are denoted by the gray trapezoids and the size of each layer (i.e. number of units) is depicted in the figure.

Training Process: The training process of the BDQ-based control agent is outlined in Algorithm 1. At the beginning, we first initialize a neural network Q with random weight θ. Another neural network Q^- with the same architecture is also created. The outer

"for" loop controls the number of training episodes, and the inner "for" loop performs control at each control time step within one training episode. During the training process, the recent transition tuples (S_t, A_t, S_{t+1}, R_{t+1}) are stored in the replay memory Λ from which a mini-batch of samples will be generated for neural network training. The variable A_t stores the control action in the last step, and S_t and S_{t+1} represent the building state in the previous and current control time steps, respectively. At the beginning of each time slot t, we first update four actions and obtain the current state S_{t+1}. In line 7, the immediate reward R_{t+1} is calculated by Equation 3. A training mini-batch can be built by randomly drawing some transition tuples from the memory.

We calculate the target vector and update the weights of the neural network Q by using an Adam optimizer for every control step t. Formally, for an action dimension $d \in 1, ...N$ with n discrete actions, a branch's Q-value at state $s \in S$ and with action $a_d \in A_d$ is expressed in terms of the common state value $V(s)$ (the result of the shared representation layer in Figure 8) and the corresponding (state-dependent) action advantage $A_d(s, a_d)$ of each branch (the result of the each advantage dimension in Figure 8) by:

$$Q_d(s, a_d) = V(s) + (A_d(s, a_d) - \frac{1}{n} \sum_{a'_d \in A_d} A_d(s, a'_d)). \quad (4)$$

The target network Q^- will be updated with the latest weights of the network Q every c control time steps. c is set to 50 in our current implementation. Q^- is used for inferring the target value for the next c control steps. We use y_d to represent the maximum accumulative reward we can obtain in the next c steps. y_d can be calculated by temporal-difference (TD) targets in a recursive fashion:

$$y_d = R + \gamma \frac{1}{N} \sum_d Q_d^-(s', \arg \max_{a'_d \subseteq A_d} Q_d(s', a'_d)), \quad (5)$$

where Q_d^- denoting the branch d of the target network Q^-; R is the reward function result; and γ is discount factor.

Finally, at the end of the inner "for" loop, we calculate the following loss function every c control steps:

$$L = \mathbb{E}_{(s,a,r,s')} \sim D \left[\sum_d (y_d - Q_d(s, a_d))^2 \right], \quad (6)$$

where D denotes a (prioritized) experience replay buffer and a denotes the joint-action tuple $(a_1, a_2, ..., a_N)$. The loss function L should decrease as more training episodes are performed.

4.4 Reward Calculation

This section describes how we calculate the reward function in Equation 3, including energy cost E, thermal comfort T, visual comfort V and indoor air condition I.

4.4.1 Energy Consumption. The energy consumption of a building includes heating coil power P_h and cooling coil power P_c and fan power P_f from the HVAC system and electric light power P_l from the lighting system. We calculate the reward function for energy consumption E during a time slot as

$$E = (P_h + P_c + P_f + P_l) \quad (7)$$

The heating and cooling coil are used to cool or heat the air and the fan is used to distribute the heating air or cooling air to the zone. The electric lights are used for normal work in the zone. They are calculated by EnergyPlus simulator in our training and evaluation. In our current implementation, we ignore the power consumed by the water pumps and the motors to adjust blinds and windows, because it is relatively small compared with the power consumption of the HAVC system or the lighting systems, and can be safely ignored (less than 1% total).

4.4.2 Human Comfort. We define and explain the measurement of the three human comfort metrics.

Thermal Comfort: It is determined by the index PMV (Predictive Mean Vote) that is calculated by Fanger's equation [13]. PMV predicts the mean thermal sensation vote on a standard scale for a large group of persons. The American Society of Heating Refrigerating and Air Conditioning Engineers (ASHRAE) developed the thermal comfort index by using coding -3 for cold, -2 for cool, -1 for slightly cool, 0 for natural, +1 for slightly warm, +2 for warm, and +3 for hot. PMV has been adopted by the ISO 7730 standard [12]. The ISO recommends maintaining PMV at level 0 with a tolerance of 0.5 as the best thermal comfort. We calculate the reward function for thermal comfort T_c during a time slot as

$$T_c = \begin{cases} 0, & PMV \leqslant P \\ |PMV - P|, & PMV > |P| \end{cases} \quad (8)$$

The occupants can feel comfort when PMV value is within an acceptable range. We denote the range as $[-P, P]$, where P is the threshold for PMV value. If the PMV value lies within $[-P, P]$, it will not incur a penalty. Otherwise, it will incur a penalty for the occupants' dissatisfaction with the building thermal condition.

Visual Comfort: The research on visual comfort is dominated by studies analyzing the presence of an adequate amount of light where discomfort can be caused by either too low or too high level of light as glare. In this paper, the major glare metric is illuminance range [23]. The illuminance source includes daylight and electrical light. Thus, the main subsystems that can have an impact on visual comfort are blind system and lighting system. We calculate the reward function for visual comfort V_c during a time slot as

$$V_c = \begin{cases} -F - M_L, & F < M_L \\ 0, & M_L \leqslant F \leqslant M_H \\ F - M_H, & F > M_H \end{cases} \quad (9)$$

The occupants can feel comfort when illuminance value F is within an acceptable range. We denote the range as $[M_L, M_H]$, where M is the threshold for illuminance value. If the illuminance value lies within $[M_L, M_H]$, it will not incur a penalty. Otherwise, it will incur the penalty for the occupants' dissatisfaction with the building illuminance condition.

Indoor Air Quality: Carbon dioxide (CO_2) concentration in a building is used as a proxy for air quality [11]. The carbon dioxide concentration comes from building's users. There are various other sources of pollution (NOx, Total Volatile Organic Compounds (TVOC), respirable particles, etc.). Ventilation is an important means for controlling indoor air quality (IAQ) in buildings [4]. Ventilation in this work mainly comes from the HVAC system and the window system. We calculate the reward function for indoor air condition I_c during a time slot as

$$I_c = \begin{cases} -C - A_L, & C < A_L \\ 0, & A_L \leqslant C \leqslant A_H \\ C - A_H, & C > A_H \end{cases} \quad (10)$$

The occupants can feel comfort when carbon dioxide concentration value C is within an acceptable range. We denote the range as $[A_L, A_H]$, where A is the threshold for dioxide concentration value. If the dioxide concentration value lies within $[A_L, A_H]$, it will not incur a penalty. Otherwise, it will incur a penalty for the occupants' dissatisfaction with the building indoor air quality.

5 IMPLEMENTATION OF OCTOPUS

In this section, we illustrate in detail the implementation of OCTOPUS including platform setup, HVAC modeling and calibration, and OCTOPUS training.

5.1 Platform setup

Our building model is rendered using SketchUp [1]. It replicates a LEED Gold Certified Building in our University Campus. Using OpenStudio, the HVAC, lighting, blind and window system are installed in the building/zones. The control scheme - OCTOPUS is implemented using Tensorflow, which is an open-source machine learning library for Python. Using the Building Control Virtual Test Bed (BCVTB), a Ptolemy II platform that enables co-simulation across different models [31], we implement the control of each zone temperature set points, blinds, lighting and window schedule during each action time in EnergyPlus for our Building alongside weather data. OCTOPUS is modeled using EnergyPlus version 8.6 [22]. We train OCTOPUS based on 10-year weather data from two different cities, Merced, CA and Chicago, IL due to their distinct weather characteristics. The weather data for Merced has intensive solar radiation and large variance in temperature, while Chicago is classified as hot-summer humid continental with four distinct seasons. To train our model, we define an "episode" as one inner for loop of Algorithm 1.

5.2 Rule Based Method

We implement a rule-based method based on our current campus building control policy. This policy was first set up at commissioning time by a mechanical engineering company, and then it was further optimized by two experienced HVAC engineers when going over the LEED certification process.

First, we assign different zone temperature setpoints. Each zone has a separate heating and cooling setpoint. The heating setpoint is set to 70 °F, and the cooling setpoint to 74 °F during the warm-up stage. The cooling setpoint is limited between 72°F and 80°F, and the heating setpoint is limited between 65°F and 72°F. Second, we set control restrictions and actuator limits and control inputs are subject to the following constraints: the heating setpoint should not exceed the cooling setpoint minus 1 °F. The adjustment will move both the existing heating and cooling setpoints upwards or downwards by the same amount unless the limit has been reached. Third, for the control Loops: two separate control loops operate to maintain space temperature at setpoint, the Cooling Loop and the Heating Loop. Both loops are continuously active.

Table 1: Model Calibration Parameters

Parameter	Range	Adoption
Infiltration Rate	$0.01\ m^3 \sim 0.5\ m^3$	$0.05\ m^3$
Window Type/Area	Single Pane/$1m^2 \sim 4m^2$	$2m^2$
Window Thickness	$3mm \sim 6mm$	$3mm$
Fan Efficiency	$0.5 \sim 0.8$	0.7
Blind Type/Thickness	Interior Blind/$1mm \sim 6mm$	$1mm$

Table 2: Modeling Error after Calibration

	MBE	CVRMSE
February (hourly temperature)	-1.48%	5.32%
March (hourly temperature)	-0.26%	4.95%
April (hourly temperature)	1.20%	5.06%
May (hourly temperature)	0.48%	4.38%
February - May(monthly energy)	-3.83%	12.33%

5.3 HVAC Modeling and Calibration

The purpose of the calibration is to ensure the energy model can generate energy use results close to the measured values in the target building using actual inputs, including weather, occupancy schedule, and the HVAC system parameters and controls.

The first step of the calibration is to collect the real weather data from a public weather station for the period to be tested. We use a Dark Sky's API, a public weather website, to collect real weather data for three months. The second step is to replace the default occupancy schedules in the simulator with the actual occupancy schedules collected from the real target building using ThermoSense [7]. This system was installed in the target building on our campus and allows the collection of fine grain occupancy data at the zone level in the building, allowing the evaluation using accurate occupancy patterns. We used the hourly occupancy data from 3 months as the occupancy schedule in our simulated building by EnergyPlus. The third step is to calibrate certain system and control parameters to match those in the target building we want to replicate. This involves multiple issues, including (a) the selection of the parameters to be calibrated, (b) the range of those parameters, and (c) the step used within the range. In our work, we use an N-factorial design with 5 parameters and ranges to be tested based on operational experience. We tested different combinations of HVAC system parameters (Infiltration rate) and control (mass flow rate, heating, and cooling setpoints) and found the combination that minimized the calibrated error (see below). The selected calibration parameters are listed in the Table 1 with their calibration ranges and value selected. The final step is to compare the calibrated error between the calibrated model and the actual measured zone temperature and energy consumption stored in the operational building database. The whole calibration process of modeling our building takes nearly one month.

ASHRAE Guideline 14-2002 [16] defines the evaluation criteria to calibrate BEM models. According to the Guideline, monthly and hourly data can be used for calibration. Mean Bias Error (MBE) and Coefficient of Variation of the Root Mean Squared Error (CVRMSE) are used as evaluation indices. The guideline states that the model should have an MBE of 5% and a CVRMSE of 15% relative to monthly calibration data. If hourly calibration data are used, these requirements should be 10% and 30%, respectively. In our case, hourly data is used to calculate the error metrics for the average zone temperature. We choose monthly data to calculate energy error

Table 3: Human Comfort Statistical Results for Rule Based, DDQN-HVAC and OCTOPUS Schemes

Location	Method	Metric	PMV		Illuminance (lux)		CO_2 Concentration (ppm)		Energy Consumption (kWh)	
			January	July	January	July	January	July	January	July
Merced	Rule Based Method	Mean	0.03	-0.25	576.78	646.45	623.61	668.03	1990.99	3583.03
		Std	0.11	0.13	152.54	157.11	120.64	181.22		
		Violation rate	0	2%	0.94%	0	0.3%	3.629%		
	DDQN-HVAC [34]	Mean	-0.19	0.28	576.78	646.45	625.62	648.01	1859.10	3335.58
		Std	0.21	0.11	152.54	157.11	122.62	120.57		
		Violation rate	2.99%	4.4%	0.94%	0	0	0.2%		
	OCTOPUS	Mean	-0.31	0.27	587.12	569.88	594.77	612.33	1756.24	2941.46
		Std	0.2	0.10	382.27	75.83	111.59	110.35		
		Violation rate	5.7%	2.5%	0.26%	0.2%	1.31%	0.33%		
Chicago	Rule Based Method	Mean	-0.28	-0.15	583.27	637.07	610.26	638.33	3848.61	3309.56
		Std	0.11	0.02	163.96	151.37	63.94	151.37		
		Violation rate	3.09%	0	1.1%	0	0	0		
	DDQN-HVAC [34]	Mean	-0.32	0.24	583.27	637.07	612.74	649.32	3605.21	3078.67
		Std	0.08	0.07	163.96	151.37	65.09	90.16		
		Violation rate	3.7%	2.9%	1.1 %	0	0	0		
	OCTOPUS	Mean	-0.4	0.29	598.34	544.09	640.31	633.71	3496.54	2722.03
		Std	0.1	0.11	259.88	55.37	99.85	111.04		
		Violation rate	4.2%	1.47%	1.6 %	0	1%	1.31%		

metrics because energy data can only be obtained monthly. The calibration results for zone temperature and energy consumption are shown in Table 2. It is shown that less than 2% NMBE and less than 6% CVRMSE for the zone temperature can be achieved with the optimal parameter setting. We found that both the CVRMSE for the monthly heating and cooling energy demand is relatively large, but the NMBE and CVRMSE are still within the acceptable range. This means the model can achieve accurate calculation for the monthly energy.

5.4 OCTOPUS Training

10-year weather data for training from the two locations tested (Merced, CA and Chicago, IL) is randomly divided, with eight years used for training and the remaining two years used for testing. In our implementation of OCTOPUS, we use the Adam optimizer [18] for gradient-based optimization with a learning rate of 10^{-4}. We train the agent with a minibatch size of 64 and a discount factor γ = 0.99. The target network is updated every 10^3 time steps. We use the rectified non-linearity (or ReLU) [15] for all hidden layers and linear activation on the output layers. The network has two hidden layers with 512 and 256 units in the shared network module and one hidden layer per branch with 128 units. The weights are initialized using the Xavier initialization [14] and the biases were initialized to zero. We used the prioritized replay with a buffer size of 10^6. To explore actions well in our building environment, we sample actions with a Gaussian noise throughout the training. The duration of each time (action) slot is 15 minutes. We achieved convergence of our reward function after 1000 episodes as explained in Section 6.5.

6 EVALUATION

In this section, we compare the performance of OCTOPUS with the rule-based method and the latest DRL-based method.

6.1 Experiment Setting

The implementation of the rule-based HVAC control has been introduced in Section 5.2. The rule-based method only controls the HVAC system. For the conventional DRL-based method, we implement the dueling DQN architecture used in [34], which controls the water-based heating system. We name that work as DDQN-HVAC in our comparison. Since these two benchmarks do not control the light system, for a fair comparison, we initialize the lights on in all experiments. OCTOPUS may dim the lights if the blind is open during the day. In addition, the two benchmarks always leave the blind and window system closed.

The three human comfort metrics are measured by PMV, Illuminance, and carbon dioxide concentration. We set the acceptable range of three human comfort metrics according to building standards and previous experiences in related work. The comfort range of PMV is set to -0.5 to 0.5 [5]. The comfort range of illuminance is set to 500-1000 lux [23]. The comfort range of carbon dioxide concentration is set to 400-1000 ppm [4].

We use three control methods to control the building we modeled in Section 5 for two months (January and July) and at two places with distinct weather patterns. Table 3 shows the human comfort results of three control methods and their energy consumption. The violation rate is calculated as the time when the value of a human comfort metric falls beyond its acceptable range divided by the total simulated time. Other quality of service metrics, including the amount by the which the violation occurred, or combination of amount and time will be explored in future work.

6.2 Human Comfort

From the results in Table 3, we see that all three methods can maintain the PMV value in the desired range for most of the time since the violation rate is low. The average PMV violation rate of OCTOPUS and DDQN-HVAC is higher than the rule-based method by 2.19% and 2.22% respectively. The reason for this is that the DRL-based methods try to save more energy by setting the PMV to a value close to the boundary of the acceptable range. It can be observed in Table 3 that the average PMV value of OCTOPUS and DDQN-HVAC (-0.36 and -0.26) is closer to the range boundary (-0.5), compared with the rule-based method (-0.13).

For both visual comfort and indoor air quality, the three control methods provide a very small violation rate. For illuminance, the mean illuminance value of OCTOPUS and DDQN-HAVC is 590.69 lux and 610.89 lux respectively. OCTOPUS saves energy by utilizing

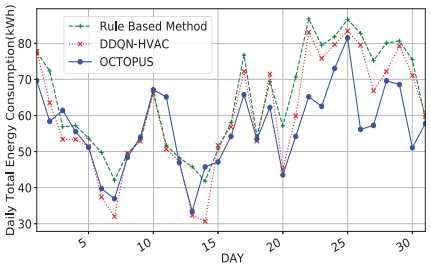

Figure 9: Daily Energy Consumption of Control Methods.

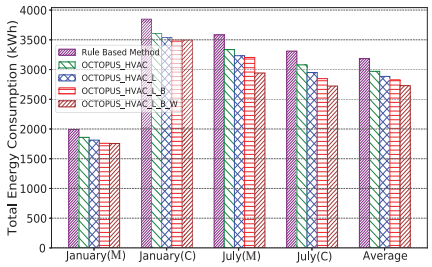

Figure 10: Performance Contribution of Each Subsystem.

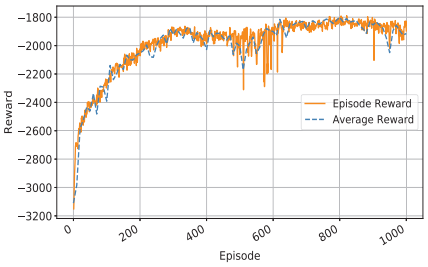

Figure 11: The Convergence of OCTOPUS.

natural light as much as possible. For indoor air quality, the average of CO_2 concentration of OCTOPUS, DDQN-HVAC, and rule-based method is 620.28 ppm, 633.92 ppm, and 635.06 ppm. OCTOPUS adjusts both window system and HVAC system to maintain the CO_2 concentration level within the desired range. DDQN-HVAC and the rule-based method only use the HVAC system.

6.3 Energy Efficiency

The results in Table 3 reveal that OCTOPUS save 14.26% and 8.1% energy on average, compared with the rule-based control method and DDQN-HVAC. In both cities, OCTOPUS achieves similar performance gain. OCTOPUS reduces the energy consumption of HVAC by using the other subsystems. Figure 9 shows a daily energy consumption of three control methods in January at Merced. In most days, OCTOPUS consumes less energy than the other two methods; however, OCTOPUS is not always the best although we see clear gains towards the second half of the month due to a change in weather temperature. The average range of outdoor temperature changes from 2°C∼ 13°C in the first half of the month to -1°C ∼ 18°C in the second half of the month. OCTOPUS could use external air with the window open for more natural ventilation.

In Table 3, compared to the rule-based method and DDQN-HVAC, OCTOPUS saves more energy in July (17.6% and 11.7%) than in January (10.05% and 3.9%). In July, the outdoor air temperature range at Merced and Chicago is 15°C ∼ 42°C and 15°C ∼ 40°C respectively. The window can be opened when the temperature is within the acceptable range, in order to save the energy consumed by the HVAC system. However, in January, due to the cold weather at both places, the windows stay closed most of the time and cannot make much contribution to energy savings.

6.4 Performance Decomposition

We implement four versions of OCTOPUS to study the energy saving contribution of each subsystem, i.e., OCTOPUS just with the HVAC system (OCTOPUS _HVAC), OCTOPUS with HVAC and lighting (OCTOPUS _HVAC_L), OCTOPUS with HVAC, lighting and blind (OCTOPUS _HVAC_L_B) and OCTOPUS with all four subsystems (OCTOPUS _HVAC_L_B_W). Figure 10 depicts the energy consumption of these four versions in two different months and at two different places (Merced and Chicago). Compared with the rule-based method, OCTOPUS _HVAC can save 6.16% more energy by only considering HVAC. When the lighting system is added in OCTOPUS _HVAC_L, 2.73% more energy can be saved. If the blind system is further added in OCTOPUS _HVAC_L_B 1.93% more energy can be saved. Finally, when the window system is added in OCTOPUS _HVAC_L_B_W, 3.44% more energy can be saved. Four subsystems make different contributions to energy saving in January and July. In January, four subsystems (i.e., HVAC, lighting, blind and window) make 6.16%, 2.73%, 1.93% and 0% contribution of energy savings respectively. In July, the contribution of these subsystems changes to 5.9 %, 3.31 %, 1.99%, and 6.4% respectively. The most obvious difference between these two months is made by the window system (6.4%). The reason for this has been explained above. In January, the windows are closed almost all the time. In July, the cold outdoor air is used to cool down the building instead of using the HVAC system.

6.5 Convergence of OCTOPUS training

Figure 11 shows that the accumulated reward of OCTOPUS in each episode during a training process. We calculate the reward function every control time step (15 minutes), and thus one episode (one month) contains 2880 time steps. The accumulated reward of one episode (episode reward in Figure 11) is the sum of the rewards of 2880 time steps. From the results in Figure 11, we see that the episode reward increases and tends to be stable as the number of training episodes increases. When the episode reward does not change much, it means that we cannot do further to improve the learned control policy and thus the training process converges. As indicated in Figure 11, the training reward fluctuates between two adjacent episodes, because the number of time steps is large in one episode, i.e., 2880. The rewards calculated at some of these 2880 time steps may vary dynamically because we randomly choose some time steps by an exploration rate (determined by a Gaussian distribution with a standard deviation of 0.2). At these time steps, we do not use the action generated by the agent, but randomly choose an action to avoid local minimum convergence. If we smooth the episode reward using a sliding window of 10 episodes, the average reward in Figure 11 is more stable during the training.

7 DISCUSSION

Deploying in a Real Building. Although we have developed a calibrated simulation model of a real building on our campus for training and evaluation, we have not deployed OCTOPUS in the building, because we do not have access to automatic blind and window system at the moment. We are seeking financial support to work with our facility team for a possible upgrade. OCTOPUS is designed for real deployment in buildings. For a new building, we need to build an EnergyPlus model for it and calibrate the model using real building operation data. After training the OCTOPUS control agent using the calibrated simulation model and real weather data,

we can deploy the trained agent in the building for real-time control. For a certain action interval (e.g., every 10 mins), the OCTOPUS control agent takes the state of the building as input and generates the control actions of four subsystems. OCTOPUS can provide real-time control, as one inference only takes 22 ms. We plan to deploy OCTOPUS in a real building in our future work.

Scalability of OCTOPUS. OCTOPUS can work in a one-zone building with one HVAC system, lighting zone, blind and window. However, a realistic building (or even a small home) is usually equipped with many lighting zones, blinds and windows which may take different actions in one subsystem. OCTOPUS may solve this scalability problem by increasing the number of BDQ branches, i.e., each branch corresponds to one subsystem in each zone of a building. We will tackle this scalability problem in our future work.

Building Model Calibration. A critical component of our architecture is the use of a calibrated building model that is close to the target building, allowing us to generate sufficient data for our training needs. However, getting a calibrated model "right" is a tedious process of trial-and-error over a large number of parameters. Out of the thousands of parameters available in EnergyPlus, we use our experience and consulted experts to determine both the most important parameters and a sensible range of values to explore (it took us four weeks to get it "right"). However, there is no magic bullet, and this may become a problem, especially for unusual building architectures or specialized HVAC systems that may not be trivial to replicate in a simulation environment.

Accepting Users' Feedback. Some existing work [32] allows users to send their feedback to the control server. The feedback can represent a user's personalized preference on different human comfort metrics and will be considered in the control decision process. OCTOPUS can easily accept users' feedback to train a better agent model by making a small modification, i.e., changing the calculated comfort values in the reward function by the users' feedback. This can be used for the initial training or for updated training (once deployed). For example, the OCTOPUS control agent can be trained incrementally with a certain time interval (e.g., one month). The newly-trained agent will be used for real-time.

8 CONCLUSIONS

This paper proposes OCTOPUS, a DRL-based control system for buildings that holistically controls many subsystems in modern buildings (e.g., HVAC, light, blind, window) and manages the trade-offs between energy use and human comfort. As part of our architecture, we develop a system that addresses the issues of large action state, a novel reward function based on energy and comfort, and data requirements for training using existing historical weather data together with a calibrated simulator for the target building. We compare our results with both the state-of-art rule-based control scheme obtained from a LEED Gold certified building, a DRL scheme used for optimized heating in the literature, and show that we can get 14.26% and 8.1% energy savings while maintaining (and sometime even improving) human comfort values for temperature, air quality and lighting.

REFERENCES

[1] 2018. sketchup. https://www.sketchup.com
[2] 2019. https://www.geze.com/en/discover/topics/natural-ventilation/
[3] 2019. https://www.buildings.com/article-details/articleid/12969/title/operable-windows-for-operating-efficiency
[4] ANSI/ASHRAE Standard 62.1. 2016. Ventilation for Acceptable Indoor Air Quality.
[5] Refrigerating American Society of Heating and Air-Conditioning Engineers. Standard 55. 2017. Thermal Environmental Conditions for Human Occupancy.
[6] Alex Beltran and Alberto E Cerpa. 2014. Optimal HVAC building control with occupancy prediction. In *ACM BuildSys*.
[7] Alex Beltran, Varick L Erickson, and Alberto E Cerpa. 2013. Thermosense: Occupancy thermal based sensing for hvac control. In *ACM BuildSys Workshop*.
[8] Zhijin Cheng, Qianchuan Zhao, Fulin Wang, Yi Jiang, Li Xia, and Jinlei Ding. 2016. Satisfaction based Q-learning for integrated lighting and blind control. *Energy and Buildings* (2016).
[9] Konstantinos Dalamagkidis, Denia Kolokotsa, Konstantinos Kalaitzakis, and George S Stavrakakis. 2007. Reinforcement learning for energy conservation and comfort in buildings. *Building and environment* (2007).
[10] Roel De Coninck and Lieve Helsen. 2016. Practical implementation and evaluation of model predictive control for an office building in Brussels. *Energy and Buildings* (2016).
[11] Steven J Emmerich and Andrew K Persily. 2001. *State-of-the-art review of CO2 demand controlled ventilation technology and application.* Citeseer.
[12] PO Fanger. 1984. Moderate thermal environments Determination of the PMV and PPD indices and specification of the conditions for thermal comfort. *ISO 7730* (1984).
[13] Poul O Fanger et al. 1970. Thermal comfort. Analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.* (1970).
[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*.
[15] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *AISTATS*.
[16] ASHRAE Guideline. 2002. Guideline 14-2002, Measurement of Energy and Demand Savings. *American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia* (2002).
[17] Kazufumi Ito and Karl Kunisch. 2008. *Lagrange multiplier approach to variational problems and applications.* Siam.
[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[19] D Kolokotsa, GS Stavrakakis, K Kalaitzakis, and D Agoris. 2002. Genetic algorithms optimized fuzzy controller for the indoor environmental management in buildings implemented using PLC and local operating networks. *Engineering Applications of Artificial Intelligence* (2002).
[20] Bocheng Li and Li Xia. 2015. A multi-grid reinforcement learning method for energy conservation and comfort of HVAC in buildings. In *IEEE CASE*.
[21] Oswaldo Lucon, Diana Ürge-Vorsatz, A Zain Ahmed, Hashem Akbari, Paolo Bertoldi, Luisa F Cabeza, Nicholas Eyre, Ashok Gadgil, LD Harvey, Yi Jiang, et al. 2014. Buildings. (2014).
[22] U.S. Department of Energy. 2016. EnergyPlus 8.6.0. https://energyplus.net/
[23] David Christopher Pritchard. 2014. *Lighting.* Routledge.
[24] Wai Wai Shein, Yasuo Tan, and Azman Osman Lim. 2012. PID controller for temperature control with multiple actuators in cyber-physical home system. In *IEEE NBiS*.
[25] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. 2019. DeepAPP: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction. In *ACM SenSys*.
[26] Arash Tavakoli, Fabio Pardo, and Petar Kormushev. 2018. Action branching architectures for deep reinforcement learning. In *AAAI*.
[27] Athanassios Tzempelikos and Andreas K Athienitis. 2007. The impact of shading design and control on building cooling and lighting demand. *Solar energy* (2007).
[28] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *AAAI 2016*.
[29] Liping Wang and Steve Greenberg. 2015. Window operation and impacts on building energy consumption. *Energy and Buildings* 92 (2015), 313–321.
[30] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *ACM DAC*.
[31] Michael Wetter. 2011. Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation* (2011).
[32] Daniel A Winkler, Alex Beltran, Niloufar P Esfahani, Paul P Maglio, and Alberto E Cerpa. 2016. FORCES: feedback and control for occupants to refine comfort and energy savings. In *ACM UbiComp*.
[33] 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*.
[34] Zhiang Zhang and Khee Poh Lam. 2018. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *ACM BuildSys*.