Multi-Zone HVAC Control With Model-Based Deep Reinforcement Learning

Xianzhong Ding, Alberto Cerpa, *Member, IEEE*, and Wan Du, *Member, IEEE*

***Abstract*— The application of reinforcement learning in controlling Heating, Ventilation, and Air Conditioning (HVAC) systems has been extensively researched. Existing studies primarily focus on Model-Free Reinforcement Learning (MFRL), which involves trial-and-error interactions with real buildings to train the agent. However, MFRL encounters a significant challenge: it requires a large amount of training data to achieve satisfactory performance. While simulation models have been used to generate training data and expedite the training process, they necessitate high-fidelity building models that are difficult to calibrate. As a result, Model-Based Reinforcement Learning (MBRL) has been employed for HVAC control. Although MBRL demonstrates remarkable sample efficiency, it often falls short in terms of asymptotic control performance, particularly in achieving substantial energy savings while ensuring occupants' thermal comfort. In this study, we conduct experiments to analyze the limitations of current MBRL-based HVAC control methods, focusing on model uncertainty and controller effectiveness. Leveraging the insights gained from these experiments, we develop MB^{2}C, an innovative MBRL-based HVAC control system that combines high control performance with exceptional sample efficiency. MB^{2}C learns the dynamics of the building by employing an ensemble of environment-conditioned neural networks and utilizes a novel control method called Model Predictive Path Integral (MPPI) for HVAC control. MPPI generates candidate action sequences using an importance sampling weighted algorithm, which is well-suited for multi-zone buildings with high state and action dimensions. We evaluate MB^{2}C using EnergyPlus simulations in a five-zone office building, and the results demonstrate that MB^{2}C achieves 8.23% higher energy savings compared to the state-of-the-art MBRL solution while maintaining comparable thermal comfort. Moreover, MB^{2}C significantly reduces the required training data set by an order of magnitude (10.52×) while delivering performance on par with MFRL approaches.**

***Note to Practitioners*—Our research addresses a critical challenge in HVAC control, offering an innovative solution to enhance the data efficiency of HVAC systems while optimizing energy usage. Traditional approaches, such as Model-Free Reinforcement Learning, often require a large volume of real-world data. Our primary focus is improving the effectiveness of HVAC control, a vital aspect of building management that directly affects energy consumption and occupant well-being. We introduce MB^{2}C, a Model-Based Reinforcement Learning system designed to significantly improve energy savings while maintaining thermal comfort. MB^{2}C achieves remarkable results, offering exceptional sample efficiency and substantially reducing the required training data. Our research leverages an ensemble of environment-conditioned neural networks and employs Model Predictive Path Integral in HVAC control. While MB^{2}C presents notable benefits, it also has limitations. Further research and development are required to optimize its performance across different building environments and specific use cases. Future directions should focus on addressing the safety challenges associated with real-world deployment. Beyond HVAC control, the principles and methods explored in this research have potential applications in various automation domains, such as robotics, industrial automation, and manufacturing processes.**

***Index Terms*— HVAC control, model-based deep reinforcement learning, model predictive control, energy efficiency, optimal control.**

I. Introduction

PEOPLE spend the majority of their time indoors [2], making the optimization of indoor environmental quality through Heating, Ventilation, and Air Conditioning (HVAC) systems not only a matter of comfort but also of health and energy efficiency. HVAC control is the process of regulating these systems to achieve a balance between occupant comfort and energy usage. The primary control targets of HVAC systems include maintaining optimal indoor air quality, ensuring thermal comfort for building occupants and minimizing energy consumption and operational costs. In the United States, buildings account for approximately 40% of total energy consumption, with HVAC systems consuming about half of this energy [3], [4]. The evolution towards smart buildings, powered by the Internet of Things (IoT) [5], [6], has created opportunities for more sophisticated control strategies [7], [8]. These strategies aim to enhance energy efficiency and occupant comfort by using sensor data to intelligently manage HVAC operations [9] and analyze behavior [10].

In HVAC systems, the widespread adoption of Rule-based Control (RBC) enables the adjustment of actuators such as temperature and fan speed [11]. A significant advantage of RBC is its simplicity, making it easy to comprehend. However, RBC relies on static thresholds and if-then rules, often derived from rule-of-thumb guidelines and the knowledge of

Manuscript received 25 March 2024; accepted 27 May 2024. This article was recommended for publication by Associate Editor G. Chen and Editor Z. Li upon evaluation of the reviewers' comments. This work was supported in part by NSF under Grant 2239458 and in part by University of California (UC) National Laboratory Fees Research Program under Grant 69763. An earlier version of this paper was presented at the Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, 2020 [DOI: 10.1145/3408308.3427986]. *(Corresponding author: Wan Du.)*

The authors are with the Department of Computer Science and Engineering, University of California at Merced, Merced, CA 95343 USA (e-mail: xding5@ucmerced.edu; acerpa@ucmerced.edu; wdu3@ucmerced.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2024.3410951.

Digital Object Identifier 10.1109/TASE.2024.3410951

engineers and facility managers. This approach faces two primary challenges. Firstly, RBC does not scale effectively as buildings become larger and more complex, necessitating the addition of numerous rules. Secondly, RBC struggles to handle incomplete or inaccurate information, which is a common occurrence in practical building scenarios. It is important to note that RBC does not offer a guaranteed optimal control solution.

To address these limitations, extensive research has been conducted on Model Predictive Control (MPC), which leverages an analytical building model [12], [13]. In this approach, an optimization problem is formulated with the building model and specific constraints, allowing for the simultaneous optimization of actions and building states through the use of analytic gradient computation. However, to achieve fast and scalable optimization, this methodology often requires convexification of the cost function and approximations of building dynamics [14]. Moreover, accurately capturing the complexities of thermal dynamics and various influential factors (e.g., building layouts, HVAC configurations, and occupancy patterns) in analytical energy models for heterogeneous buildings [15] is a challenging task. As a result, existing solutions often employ simplified models to meet the data requirements for parameter fitting and address computational complexity [12], [13]. For instance, Gnu-RL [16] adopts a differentiable MPC policy, employing a simplified linear model to represent the dynamics of a water-based radiant heating system.

The field of HVAC control has witnessed extensive research on Reinforcement Learning (RL) techniques [17], [18], [19], [20]. RL offers adaptability to various environments by learning control policies through direct interactions with the environment [21], [22], [23]. Currently, the prevalent approach relies on Model-Free Reinforcement Learning (MFRL) to obtain optimal HVAC control policies through trial-and-error interactions with real-world buildings. However, the convergence of MFRL requires a substantial number of interactions, with our experiments indicating a need for 500,000 timesteps (equivalent to 5200 days) to achieve desirable control performance. Although the use of simulated building models can accelerate the training process, it demands highly accurate calibration, posing challenges [17], [19]. In recent years, Model-Based Reinforcement Learning (MBRL) has been explored for HVAC control, with a focus on achieving data efficiency [24]. Initially, an HVAC system learns its dynamics using a neural network trained on historical HVAC data. Subsequently, an MPC controller, based on the learned building dynamics model, employs the Random Shooting (RS) method to determine optimal control actions [24]. In the case of single-zone HVAC systems, the MBRL-based approach achieves approximately 10 times faster training compared to MFRL, while maintaining comparable performance [24]. However, it is important to note that the aforementioned approach is not suitable for multi-zone HVAC systems, which are prevalent in commercial buildings [25]. Moreover, MBRL approaches often lag behind MFRL schemes in terms of control performance, particularly regarding achieving high energy savings while ensuring occupants' thermal comfort.

This paper presents MB^{2}C, an innovative HVAC control approach based on Model-Based Reinforcement Learning, to overcome the limitations observed in existing methods. MB^{2}C aims to achieve the combined benefits of MBRL's data efficiency and Model-Free Reinforcement Learning's control performance. The core objective of MB^{2}C is to optimize energy savings while meeting occupants' thermal comfort requirements. Energy consumption and thermal comfort in a building's HVAC system are influenced by various factors, including the current state of all zones, outdoor weather conditions, and control actions (e.g., temperature setpoints). In multi-zone buildings, control actions are represented as a vector, denoted as A_s, encompassing the control actions for each thermal zone. MB^{2}C identifies the best A_s from all possible action combinations, A_{all}, for each control cycle. The selected A_s ensures that thermal comfort remains within an acceptable range throughout the control interval, while minimizing energy consumption. MB^{2}C comprises two main components: (a) a building dynamics model and (b) an HVAC control algorithm.

Our approach changes the way we model building dynamics by using a group of neural networks, each carefully designed to adapt to different environmental situations. At the heart of our system is a neural network model built to understand both the current state of the building and future actions, making it possible to predict future states accurately. The strength of this model is greatly increased by our new weighted ensemble learning algorithm, which combines the outputs of different building dynamics models. This algorithm uses a dynamic weighting process, carefully adjusting how much each model affects the overall prediction based on how accurate it is. This method helps us deal with the challenges of model uncertainties, ensuring predictions that are both reliable and strong.

Moreover, we introduce an environment-conditioned neural network architecture, a strategic innovation that categorizes state variables into those that are modifiable by control actions, such as the temperatures within different zones, and those that are inherently influenced by external environmental factors, such as the outdoor temperature. This critical difference underscores the understanding that environmental conditions are beyond the scope of direct control actions. By integrating this architecture, our model achieves an unprecedented level of detail in depicting the complex dynamics between the building's operational state, the executed control actions, and the surrounding environmental conditions. This integration significantly boosts the accuracy and efficacy of our predictive models and HVAC control strategies, paving the way for optimized energy efficiency and improved occupant comfort.

The benefits of our proposed approach extend far beyond enhanced predictive accuracy. By differentiating between controllable and uncontrollable variables, our model enables more strategic, informed decisions in HVAC control, leading to significant improvements in energy conservation and occupant satisfaction. The adaptive learning capability introduced by our dynamic weighting mechanism allows for continuous refinement and improvement of the system, ensuring it remains

effective under evolving conditions and insights. Additionally, the environment-conditioned architecture ensures that our model is finely attuned to real-world conditions, facilitating a more intelligent, responsive approach to building environment management.

To tackle the challenge of control optimization within our proposed building dynamics model, we utilize a highly adaptable strategy known as the shooting method. This technique involves generating stochastic action trajectories over a defined future timeline [26]. Each action trajectory represents a planned sequence of actions for forthcoming time-steps. During the evaluation phase, we analyze every sequence over a set number of time-steps, denoted as H, yet we strategically execute only the initial action at the next time-step. This selective execution ensures that our model remains both proactive and responsive to immediate operational demands. In some MBRL-based HVAC control solutions [24], the RS method has traditionally been employed, where potential actions are randomly selected according to a uniform distribution. Despite its simplicity, RS often falls short in identifying the most effective action trajectory, mainly because the randomness does not guarantee coverage of the optimal path. To overcome this critical drawback, we have integrated the MPPI control method into our framework. Renowned for its efficacy in robotics [27], MPPI excels by computing an optimal control action through a noise-weighted average of sampled action trajectories, a process that meticulously fine-tunes both the initial control input and the variance of the sampling distribution to pinpoint the optimal action.

By tailoring the MPPI method specifically for HVAC control within an MBRL setup, and by optimizing parameter settings, we significantly improve the precision and efficiency of our control strategies. This customization allows us to leverage MPPI's strengths—such as its ability to navigate complex, dynamic environments and its robustness against uncertainty—in the context of building climate control. The result is a control optimization solution that not only meets but exceeds traditional performance benchmarks, ensuring energy-efficient operation while maintaining optimal environmental conditions.

We implement MB^{2}C using TensorFlow, a Python-based open-source machine learning library. The building dynamics model is constructed using a 3-layer neural network, and the control algorithm is based on MPPI. To assess the performance of MB^{2}C, we conduct experiments on a building consisting of five thermal zones. These experiments involve extensive simulations using EnergyPlus. Through these simulations, we evaluate MB^{2}C and compare its performance against benchmark methods. The results demonstrate that MB^{2}C surpasses the latest model-based DRL method by achieving an 8.23% reduction in total energy consumption for the building, all while maintaining optimal thermal comfort. Moreover, when compared to the model-free DRL approach, MB^{2}C significantly reduces the training convergence time, achieving an improvement of 10.52×, which is more than an order of magnitude.

We summarize the main contributions of this paper as follows:

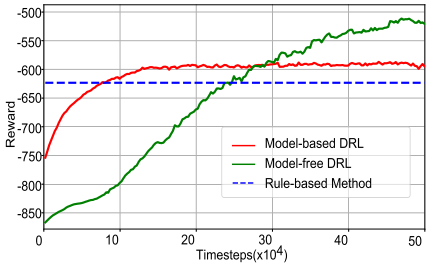


Fig. 1. Convergence time and the achieved reward.

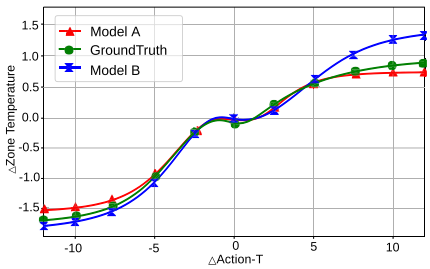


Fig. 2. Uncertainty of the building dynamics model.

- We provide a thorough examination of the existing limitations found in both MFRL and MBRL strategies, highlighting areas where improvements are necessary for effective HVAC control in multi-zone environments.
- We introduce MB^{2}C, a novel HVAC control system rooted in MBRL principles, specifically designed for multi-zone buildings. MB^{2}C stands out by delivering superior control performance paired with exceptional data efficiency, setting a new benchmark in the field.
- Through rigorous experimentation, we validate the superiority of MB^{2}C over existing methods. Our results demonstrate MB^{2}C's significant advancements, including its ability to cut down total energy consumption by 8.23% compared to the latest model-based DRL methods, while also ensuring optimal thermal comfort. Additionally, MB^{2}C markedly reduces the training convergence time, achieving a more than tenfold improvement over model-free DRL approaches.

This journal article substantially builds on our prior conference paper[1], presenting extensive revisions and new content. Notable improvements include refined discussions in the introduction and motivation sections and a deeper dive into the environment-conditioned neural network architecture and weighted ensemble learning. We have enriched the paper with detailed figures and tables such as the HVAC system layout and DRL component summaries, enhancing understanding of our methodologies and findings. Further, we delve into the dynamics of energy efficiency, neural network architectures, ensemble model impacts, and execution overhead analysis, providing a more comprehensive view of our MB^{2}C system's capabilities. This expansion not only demonstrates the depth of our enhancements but also solidifies our contributions to advancing multi-zone HVAC control using model-based deep reinforcement learning.

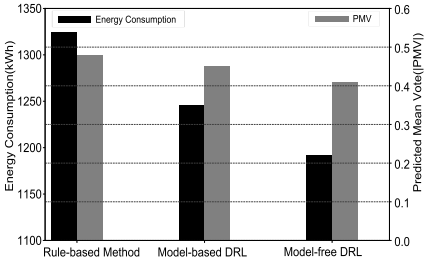


Fig. 3. Random shooting in the model-based DRL method.

II. Motivation

To evaluate the performance of the state-of-the-art MBRL method[24], we conduct a series of simulations in EnergyPlus for a multi-zone building with five zones. All system settings remain consistent with those in [24], with the exception of the state and action dimensions, which increase due to the additional zones. Our approach utilizes a deterministic neural network to model the building dynamics, and we employ a random shooting method to determine the optimal heating and cooling setpoints. In addition, we implement Proximal Policy Optimization (PPO) [28], a simple MFRL-based method, for comparison purposes and to ensure the proper functioning of the simulator, yielding intuitive and comprehensible results.

Our primary objective is to examine the impact of a multi-zone building on existing model-based and model-free DRL control methods. To assess the level of thermal comfort achieved, we measured the Predicted Mean Vote (PMV) [29], which should be maintained within the range of -0.7 to 0.7. The simulations are conducted using weather data for the month of January, considering a 463 m^2 building located in Fresno, CA. The building features windows on all four facades, while the south- and north-facing glass doors are shaded by overhangs. In the context of our five-zone building, the state dimension encompasses 37 variables, including indoor air temperature, humidity, PMV, energy consumption for each zone, and relevant outdoor environmental parameters. Furthermore, the action dimension comprises 10 variables, representing cooling and heating setpoints for each zone.

A. Experiment Results

Figure 1 illustrates the energy-saving performance of model-based and model-free DRL control methods based on 50×10^4 time-steps of training data. The reward represents the energy-saving performance while maintaining reasonable thermal comfort, as defined in Section III-B4. We assess the accumulated reward every 2976 time-steps (equivalent to one month). The reward for the rule-based method remains constant since it is unaffected by changing weather data and building environment, resulting in a linear line.

From Figure 1, it is evident that both the model-based DRL and PPO methods require 7.5×10^4 and 23.75×10^4 time-steps, respectively, to surpass the performance of the rule-based method. In terms of convergence time, the model-based method requires 11.5×10^4 time-steps, while the PPO method requires 50×10^4 time-steps. The model-based method proves to be 4.38 times more data-efficient than the PPO method. However, in the long run, the model-free method eventually outperforms the model-based method. The model-free method follows a trial-and-error approach, and its performance improves with additional training data. In contrast, the model-based method in this case fails to achieve the same level of performance as the model-free method even with increased training data. The efficiency of the model-based method [24] diminishes as the state and action dimension grows, as observed in our 5-zone building with 47 dimensions.

The challenges associated with the high state and action space lie in the incapacity of the current model to accurately capture the building dynamics, leading to sub-optimal selection of heating and cooling setpoints by the controller. Another contributing factor is the diminishing effectiveness of the random shooting method as the action space expands. To gain insights into these issues, we delve into the details of the existing model-based method, examining its components from two perspectives: the uncertainty of the building dynamics model and the efficacy of the control method.

B. Challenge 1 - Model Uncertainty

Neural network models can exhibit epistemic uncertainty due to limited data, which hinders their ability to uniquely capture the underlying system [30], [31], [32], [33]. In MBRL-based HVAC control systems, the building dynamics model predicts the next state of the building based on the current state (e.g., zone temperature) and a control action (e.g., temperature set-points for actuators). Even a small bias in the building dynamics model can have a significant impact on the controller's decision [31], [32]. To investigate this uncertainty in existing building dynamics models, we conducted an experiment using 8000 historical data points for training and 2000 data points for testing.

Figure 2 illustrates the predicted zone temperature as a function of the performed action. The x-axis represents the temperature differential between the supply temperature (action) and the zone temperature at time t, while the y-axis displays the temperature differential between the zone temperature after and before actuation. The figure presents the predicted temperatures from two neural network models and the ground truth. Both models have the same architecture and are trained with identical training data but start with different initialization states. In the middle region of Figure 2, where we have sufficient data since most actions in the historical data do not induce sharp state changes, both models accurately predict the next state. However, when actions aim to cause significant state changes, we lack sufficient training data, leading to divergence in the performance of the two models.

C. Challenge 2 - Controller Effectiveness

The RS algorithm generates N independent random action sequences $\{a_t, \dots a_{t+H-1}\}$, where each sequence $A_i = \{a_0^i \dots a_{H-1}^i\}$ for $i = 1 \dots N$ has a length of H actions. Given a reward function $r(s,a)$ that defines the task and the future state predictions $\hat{s}_{t+1} = s_t + f_\theta(\hat{s}_t, a_t)$ from the

learned dynamics model f_θ, the optimal action sequence A_{i*} is selected based on the highest predicted reward: $i^* = \arg\max_i R_i = \arg\max_i \sum_{t'=t}^{t+H-1} r(\hat{s}_{t'}, \hat{a}_{t'})$. This approach has demonstrated success in controlling single-zone buildings using learned models. However, when applied to a five-zone building, it exhibits several drawbacks. Firstly, it scales poorly with the dimensions of the planning horizon and action space. Secondly, it often falls short in achieving high task performance since a randomly sampled sequence of actions does not directly result in meaningful behavior. When considering a five-zone building, the RS approach encounters challenges due to its limitations. It struggles to handle the increased complexity associated with larger planning horizons and action spaces, hampering its scalability. Moreover, the randomness inherent in selecting action sequences can lead to suboptimal performance as these random actions may not align with desired behavior or achieve the desired outcomes.

In Figure 3, we examine the energy consumption and thermal comfort provided by three HVAC control methods: a rule-based method, a model-based method, and a model-free method. To eliminate the influence of model uncertainty on the model-based method, we utilize the ground-truth states of the building as the outputs of the building dynamics model, resulting in perfect future state predictions. From the findings in Figure 3, we observe that all three methods successfully meet the required thermal comfort level, as indicated by the equivalent PMV values (0.48, 0.45, 0.41). However, the model-based method exhibits a 4.70% higher energy consumption compared to the model-free method. This disparity can be attributed to the RS control approach utilized, where the perfect building dynamics model employed in the model-based method contributes to this outcome during the experiment. Also, it is reasonable to say that model-free might work better in the end, although it needs more iterations to converge, potentially capturing more straightforward strategies through direct environmental interactions.

D. Summary

Building upon the aforementioned observations, our primary objective is to address the limitations associated with model uncertainty and controller effectiveness. We aim to develop a methodology that combines the exceptional performance of model-free methods with the sample and data efficiency typically exhibited by model-based approaches. By achieving this, we strive to strike a balance where the resulting method excels in both performance and efficiency.

III. Design of MB^{2}C

In this section, we present the design of MB^{2}C, which encompasses various aspects of multi-zone building control. Specifically, we outline the model-based DRL approach employed, the architecture and training specifics of the building dynamics model, the methodology for online control action planning, and the in-situ update process of the building dynamics model.

A. MB^{2}C Overview

Figure 4 illustrates the overall structure of MB^{2}C, a model-based DRL control approach [32] designed for multi-zone building HVAC systems. The framework comprises two essential components: a building dynamics model and a MPPI-based controller. The building dynamics model is constructed using an Ensemble of Environment-conditioned Neural Networks (ENN). It takes into account the current state of the building HVAC system and a specific control action as inputs, and generates the predicted next state of the building HVAC system. By leveraging historical data, we train the building dynamics model through a supervised learning process. Equipped with the trained model, our MPPI-based controller assesses different control actions and determines the optimal action for the subsequent time step. This approach ensures compliance with thermal comfort requirements while minimizing energy consumption.

During deployment in a building, MB^{2}C carries out the optimal control action by adjusting the relevant actuators within each control cycle. Simultaneously, we collect building data traces, which consist of the next HVAC state determined by the current HVAC state and the executed control action. By utilizing these newly acquired building traces, we can periodically update the building dynamics model in-situ to enhance its accuracy. This update is performed at regular intervals, such as every week, using a sliding window of 2 months, as the seasonal characteristics of the data change throughout the year. An iterative training process for updating the model takes approximately 25.32 minutes to complete on a laptop equipped with an Intel 4-core i7-6700 CPU and Nvidia GTX 960M GPU. Importantly, this training process can be conducted in parallel while the current model continues to operate in the building, ensuring that the overhead of the iterative training does not interfere with the real-time usage of MB^{2}C in practical building applications.

B. Model-Based Deep Reinforcement Learning for Multi-Zone Building Control

Our work involves an extension of the current MBRL-based method to address multi-zone building HVAC control, incorporating the design of crucial components specific to this domain.

1) Preliminaries for DRL: Reinforcement learning aims to develop a policy that maximizes the cumulative rewards over time. At each time step t, the controller exists in a state $s_t \in S$, performs an action $a_t \in A$, receives a reward $r_t = r(s_t, a_t)$, and transitions to the next state s_{t+1} based on an unknown dynamics function $f : S \times A \to S$. The primary objective at each time step is to select the action that maximizes the discounted sum of future rewards, represented by $\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})$, where $\gamma \in [0, 1]$ is a discount factor that prioritizes immediate rewards. Notably, it is essential to have knowledge of the underlying reward function $r(s_t, a_t)$ used for planning actions under the learned model in order to perform this policy extraction.

In model-based reinforcement learning, we utilize a dynamics model to predict future states, which guides the

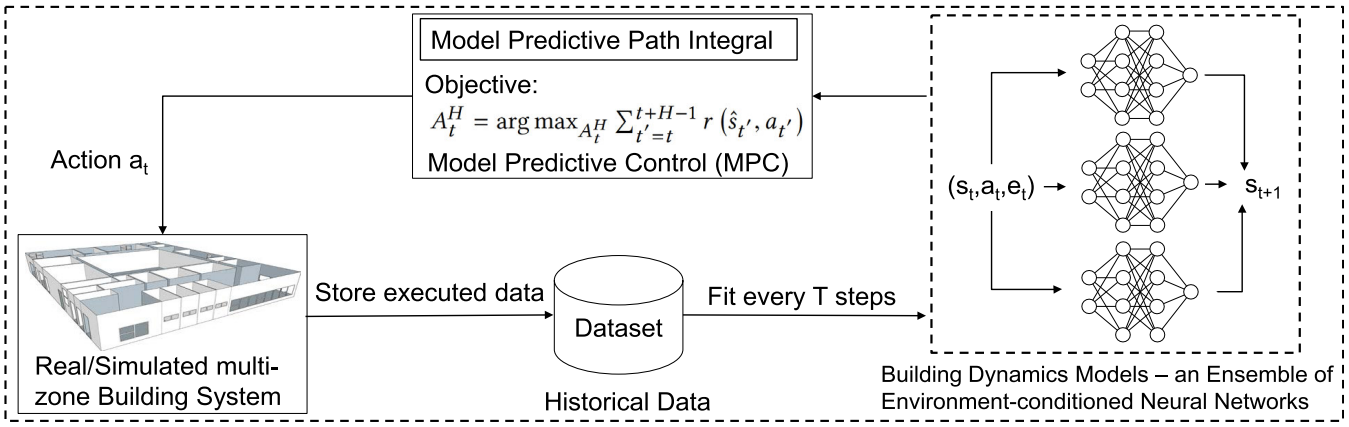


Fig. 4. Overall of the proposed building energy control framework.

selection of actions. The learned discrete-time dynamics function $f_\theta(s_t, a_t)$, with θ as the parameter, takes the current state s_t and action a_t and estimates the next state at time $t+\Delta t$. By solving the optimization problem:

$$(a_t, \ldots a_{t+H-1}) = \arg\max_{a_t, \ldots a_{t+H-1}} \sum_{t'=t}^{t+H-1} \gamma^{t'-t} r(s_{t'}, a_{t'}) \tag{1}$$

we identify the action sequence that maximizes the discounted sum of rewards over H future time steps. In practice, it is advantageous to solve this optimization problem at each time step, execute only the first action from the sequence, and then re-plan at the subsequent time step using updated state information. This control scheme is commonly known as MPC and is effective in compensating for model inaccuracies.

2) State Design: The state serves as the input for the building dynamics model to make predictions in the next time step. In our study, we divide the state into two distinct parts as shown in Table I: (a) the building state (s_{ti}), which encompasses the state variables that are influenced by our control actions, and (b) the environment state (e_{ti}), which includes the state variables that remain unaffected by our control actions.

Building State (s_{ti}) The building state vector for the ith zone encompasses the variables that undergo changes over time t. It includes the following items: indoor air temperature, indoor air relative humidity, PMV, heating energy consumption, and cooling energy consumption. These variables reflect the dynamic characteristics of the indoor environment and the energy consumption patterns within the specific zone.

Environmental State (e_{ti}) The environment state vector for the ith zone comprises the variables that undergo changes over time t. It includes the following items: outdoor air temperature, outdoor air relative humidity, diffuse solar radiation, direct solar radiation, solar incident angle, wind speed, wind direction, and occupancy flag. The occupancy flag serves as an indicator to determine the presence of individuals in the ith zone, and it is the only element in the vector that changes per zone. These variables capture the external conditions and occupancy information that influence the thermal dynamics and energy performance of the building.

Using our 5-zone building as an illustration, the state dimension encompasses 37 variables, including both the building state and environment state variables. To ensure uniformity and scale in the range of values, we apply min-max normalization to each item, transforming them into values within the range of 0 to 1. This normalization process enables effective comparison and integration of different state variables, facilitating accurate modeling and control of the building HVAC system.

3) Action Design: The action vector a_{ti} represents the variables that the controller utilizes to actively manipulate the building state s_{ti}. In our multi-zone system, the action state vector comprises the cooling temperature set-point and the heating temperature set-point, both measured in degrees Celsius, for each zone. These set-points determine the desired temperature range for cooling and heating operations in each specific zone.

When considering the dynamics of the system at each time step t, the action state vector varies to reflect the changes in set-points for each zone. The controller aims to find the most suitable combination of actions $a_{(t+1)i}$ for all zones based on the current state s_{ti} and e_{ti} and the chosen actions a_{ti}. The primary objective is to strike a balance between energy consumption and thermal comfort metrics.

In the case of our five-zone building, the action dimension encompasses 10 variables, including the cooling and heating temperature set-points for each zone. By effectively managing these variables, the controller can optimize the HVAC system's performance to ensure energy efficiency while maintaining a comfortable indoor environment.

4) Reward Design: The reward function plays a crucial role in optimizing the parameters that we aim to maximize when the agent takes an action a_{ti} to transition from the current building state s_{ti} to the next state $s_{(t+1)i}$. To ensure a comprehensive optimization, both thermal comfort and energy consumption factors are incorporated into the reward function.

TABLE I
BUILDING AND ENVIRONMENTAL STATE VARIABLES

Building State		**Environmental State**			
Name	**Unit**	**Name**	**Unit**	**Name**	**Unit**
Indoor air temperature	°C	Outdoor air temperature	°C	Solar incident angle	°
Indoor air relative humidity	%	Outdoor air relative humidity	%	Wind speed	m/s
Cooling energy consumption	kWh	Diffuse solar radiation	W/m^2	Wind direction	degree from north
Heating energy consumption	kWh	Solar incident angle	°	Occupancy flag	0 or 1
PMV	/	/	/	/	/

The reward function is defined as follows:

$$R = -\sum_{i=1}^{N}(\rho Norm(|PMV_i|) + Norm(E_i)), \qquad (2)$$

Here, E represents the heating and cooling energy consumption for each zone. To estimate the comfortable temperature range for the "standard" occupant in the current seasonal conditions, we employ Fanger's formula for the Predictive Mean Vote (PMV) [29], as outlined in the ASHRAE standard 55 [34]. The PMV values within the comfort range for Class C environments range from +/- 0.7.

The parameter ρ allows us to balance the relative importance between energy consumption and thermal comfort. During occupied periods, we set ρ to 4 since the range of human comfort and energy consumption varies compared to unoccupied periods, where we use $\rho = 0.1$. By adjusting ρ, we can capture the different priorities and requirements in terms of thermal comfort and energy efficiency based on the occupancy status.

The reward function serves as a measure to evaluate the actions taken, ensuring they meet the thermal comfort requirements for all occupants in the building. The variable N represents the total number of zones in the building. In the subsequent sections, we will simplify the notation by removing the zone index i to streamline the presentation and analysis.

C. Learning the Building Dynamics

To handle the high-dimensional state and action spaces, as well as the complex dynamics inherent in a multi-zone building, we need a parameterization approach for the building dynamics model. Therefore, we adopt a multi-layer neural network as our chosen representation for the dynamics function $\hat{f}_\theta(s_t, a_t)$, with θ being the parameter set. This neural network function is designed to provide predictions of the state changes that occur when executing action a_t from state s_t, considering a time step duration of Δt. Consequently, the predicted next state can be obtained as follows: $\hat{s}_{t+1} = s_t + \hat{f}_\theta(s_t, a_t)$.

When determining the appropriate Δt value, we need to strike a balance. Selecting a very small Δt could result in minimal state differences, rendering the learning process less effective. Conversely, if Δt is set too large, it can complicate the learning process by introducing greater complexity to the underlying continuous-time dynamics. Hence, finding an appropriate Δt is crucial to achieve meaningful learning and accurate predictions in the building dynamics model.

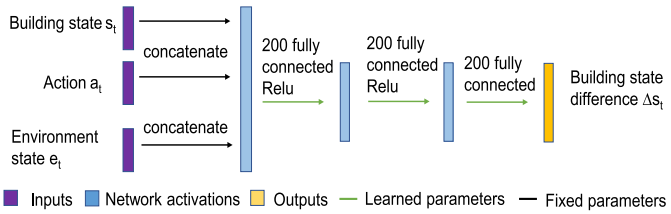


Fig. 5. Environment-conditioned neural network for our building dynamics model.

1) Environment-Conditioned Neural Network Architecture: To accurately predict the building dynamics while maintaining computational efficiency, we introduce a neural network model $\hat{f}_\theta(s_t, a_t)$ that incorporates environment information. In our approach, we propose a straightforward yet effective method of including the environment state e_t in the model. Specifically, we define an environment-conditioned dynamics model $\hat{f}_\theta(s_t, a_t, e_t)$, which takes into account not only the current building state s_t and action a_t but also the current environment state e_t. The model architecture is illustrated in Figure5.

To process the inputs, the building state vector s_t, action vector a_t, and environment state vector e_t are concatenated and passed through two hidden layers before reaching the final output layer. The dimensionality of these vectors was chosen to accurately reflect the dynamics of a 5-zone building, resulting in a comprehensive representation that includes 47 dimensions encompassing environmental conditions, control actions, and zone-specific parameters. This decision was based on an extensive evaluation of the model's ability to capture relevant dynamics while ensuring computational efficiency. Rather than directly outputting all the related states (building and environment), we focus on predicting the building state difference $\Delta\hat{s}_t$. This approach reduces the model's complexity in capturing unnecessary environmental changes. For the environment state inputs, such as weather data and occupancy, we use ground truth values to ensure prediction accuracy[16]. Incorporating this detailed environmental information allows us to enhance the model's predictive accuracy while maintaining computational traceability.

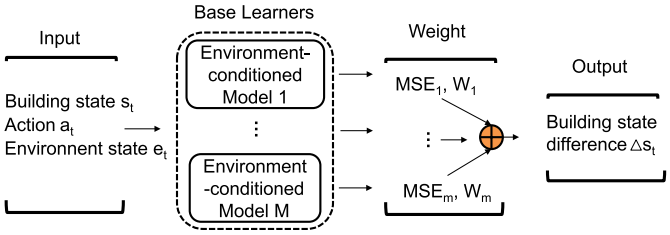


Fig. 6. Weighted ensemble learning for our building dynamics model.

The selection of network parameters in Figure 5, including the number of hidden layers and their sizes, is meticulously determined through a combination of grid search and empirical experimentation. This methodical approach allows us to finely tune the model to balance between capturing the intricate dynamics of a multi-zone HVAC system and maintaining a lean computational footprint.

2) Weighted Ensemble Learning: Capturing epistemic uncertainty in network weights has been recognized as crucial in model-based reinforcement learning, particularly when employing high-capacity models that may tend to overfit the training set and make erroneous extrapolations beyond it, as demonstrated in previous studies [31], [32]. To address this challenge, we propose a weighted ensemble learning algorithm that approximates the posterior distribution $p(\theta|D)$ using a collection of M models, each with its own set of parameters θ_i.

In the case of deep models, a straightforward approach is to initialize each model θ_i with a distinct random initialization θ_i^0 and employ different batches of data D_i at each training step. This allows the ensemble to encompass a diverse range of model configurations, facilitating the exploration of various uncertainties present in the learning process. By considering multiple models with different initializations and data subsets, we aim to capture a more comprehensive representation of the epistemic uncertainty inherent in the system.

In our approach, we have M environmental-conditioned models, as depicted in Figure 6. All M models receive the same input, which includes the building and environment states and actions. To evaluate the performance of each model, we calculate the mean squared error (MSE) over the past C timesteps (4 in our case) for each model compared to the ground truth for N states using Equation 3. The ensemble algorithm notation is illustrated in Table II.

$$MSE = \sum_{i=1}^{C}\sum_{j=1}^{N} \phi^C \left| f_\theta(s_{i,j}, a_{i,j}) - \hat{f}_{true} \right|^2 \tag{3}$$

Here, we introduce a temporal discount factor ϕ (0.9 in our case) to assess the importance of past model errors relative to the current model error. The temporal discount factor ranges between 0 and 1, as more recent prediction cases carry greater weight for the current prediction. After obtaining the MSE for each model over the past C timesteps, we normalize the MSE values to a 0-1 scale using the $Norm(x)$ process, defined as $Norm(x) = (x - x_{min})/(x_{max} - x_{min})$. We then calculate the weight ratio W for all models using Equation4.

TABLE II
ENSEMBLE PARAMETERS

Symbol	Description
N	number of state
ϕ	discount factor
M	the number of ensemble model
E	number of Ensembler
W	weight for Ensembler
P	current state prediction
C	number of past timesteps
MSE	model square error

$$W = \frac{1 - Norm(MSE_i)}{\sum_{i=1}^{M}(1 - Norm(MSE_i))} \tag{4}$$

The sum of the weights for all models is equal to 1. Subsequently, we leverage Equation 5 to predict the next state.

$$s_{t+1} = \sum_{i=1}^{M} W_i f_{\theta_i}(s, a) \tag{5}$$

This dynamic weighting mechanism allows our method to adaptively adjust the weights when aggregating the M models ($M = 5$ in our case) during the prediction process, ensuring our ensemble is optimally sized to manage the complexity of a 5-zone building with a total dimensionality of 47, encompassing both state and action spaces. The selection of this dimensionality and the number of models was guided by a comprehensive evaluation aimed at achieving a balance between model complexity and computational efficiency. This evaluation included analyzing the impact of different dimension sets on the model's performance, ensuring that the chosen configuration offers the most accurate representation of the building's dynamics while maintaining computational traceability.

D. Training the Building Dynamics Model

In this section, we outline the steps involved in preprocessing the training data and training the proposed Ensemble Neural Network (ENN) model.

1) Data Collection: To collect the training dataset $D(s_t, a_t, s_{t+1})$, we employ the rule-based controller to execute actions at each time step. During this execution, we record the resulting data τ, which consists of the state-action pairs $(s(0), a(0), s(1), a(1), \ldots, s(T-2), a(T-2), s(T-1)))$, capturing a sequence of length T. It is important to note that these recorded data differ significantly from the data that the controller will actually execute when planning with the learned dynamics model and a specific reward function $r(s_t, a_t)$ (as discussed in Section III-E). This distinction highlights the ability of model-based methods to learn from off-policy data, enabling them to generalize and make accurate predictions beyond the specific data encountered during training.

2) Data Preprocessing: To prepare the collected data $\{\tau\}$ for training, we first divide it into training data inputs (s_t, a_t) and their corresponding output labels $s_{t+1} - s_t$. In building HVAC control, the states can encompass various measurements such as temperature, humidity ratio, and energy consumption, each with its own range of values. Training a neural network model using these raw values directly may result in imbalanced losses, as the weights assigned to different measurements can vary significantly. To address this issue, we normalize the data by subtracting the mean value of the states/actions and dividing by their standard deviation. This normalization process is represented as $x' = \frac{x-\bar{x}}{\sigma(x)}$, where x refers to a state or action. By normalizing the data in this way, we ensure that each input feature contributes proportionately during training, regardless of its original range.

3) Training the ENN Dynamics Model: The ENN model is composed of an ensemble of models, each with randomly initialized parameters $\theta_1, \theta_2, \ldots, \theta_M$. To ensure that the models behave differently on the same dataset D, we use different batches of data at each training step. The dynamics model $\hat{f}_\theta(s_t, a_t)$ is trained using stochastic gradient descent[35], where the objective is to minimize the Mean Square Error (MSE) between the predicted delta observation and the ground truth delta observation. The MSE loss function is defined as follows:

$$\varepsilon(\theta) = \frac{1}{D} \sum_{(s_t, a_t, s_{t+1}) \in D} \frac{1}{2} \| (s_{t+1} - s_t) - \hat{f}_\theta(s_t, a_t) \|^2 \quad (6)$$

For training the ENN model, we utilize 5-year weather data from Fresno, CA and Chicago, IL, while using a completely different one-year dataset for testing. During training, the ENN model is provided with ground truth information on future environment states, such as weather and occupancy[16]. We employ the Adam optimizer with a learning rate of 10^{-3} for gradient-based optimization. The batch size for training is set to 512, and a discount factor γ of 0.99 is used. The training process involves 40 epochs. Each dynamics model in the ensemble consists of a neural network with two fully-connected hidden layers of size 200, using the rectified linear unit (relu) activation function, and a final fully-connected output layer. The weights and biases are initialized using the Xavier initialization process. In our experiments, we use 1000 samples for the MPC controllers (RS, CEM, and MPPI). The control cycle or timestep is set to 15 minutes, a commonly used value in traditional HVAC control [36]. Convergence is achieved by 4.75×10^4 time-steps, as explained in Section IV-C1.

E. Online Control Action Planning

In our approach, we employ online planning using MPC to determine actions based on our model predictions. Given the building state s_t at time t, the MPC controller utilizes a prediction horizon H and an action sequence $a_{t:t+H} = \{a_t, \ldots, a_{t+H}\}$. The ENN model $\hat{f}_\theta(s_t, a_t)$ provides predictions for the resulting data $s_{t:t+H}$. At each time step t, the MPC controller selects the first action a_t from the sequence of optimized actions $A_t^H = \arg\max_{A_t^H} \sum_{t'=t}^{t+H-1} r\left(\hat{s}_{t'}, a_{t'}\right)$. To compute the optimal action sequence, we employ the Model Predictive Path Integral (MPPI) control method [27].

1) Model Predictive Path Integral (MPPI) Controller: The MPPI control method has demonstrated successful autonomous control in various applications, including vehicle control. MPPI utilizes an importance-sampling weighted algorithm and employs an update rule that efficiently incorporates a larger number of samples into the distribution update. As derived by recent research on model-predictive path integral [27], the update rule for time step t, considering K predicted trajectories, can be expressed as follows:

$$a_t^{i+1} = a_t^i + \sum_{k=1}^{K} \omega(\varepsilon^k) \epsilon_t^k \quad (7)$$

Here, ω represents the importance-sampling weight for each trajectory, and ϵ denotes the noise used for exploration. The action for time step t of the $(i+1)th$ trajectory is obtained by adding the action for time step t of the ith trajectory with the noise-weighted average over the sampled trajectories. This formulation enables effective trajectory updates and promotes exploration during control.

As depicted in Algorithm 1, the initial control sequence is determined by either initializing the input buffer with zeros or utilizing a secondary controller, such as a rule-based method, and using its inputs as the initial control sequence. To begin, we sample H noise values from a normal distribution. Subsequently, we generate K trajectories for a finite horizon of length H using Brownian motion. For each generated trajectory, a cost is computed and stored in the memory (lines 2-7). This process allows us to evaluate the performance of different trajectories based on the defined cost function and retain this information for subsequent steps of the algorithm.

In model predictive control, the optimization and execution processes occur simultaneously. Initially, a control sequence is computed, and the first element of the sequence is executed. This iterative process continues, with each subsequent iteration utilizing the un-executed portion of the previous control sequence as the importance-sampling trajectory. To ensure that at least one trajectory has non-zero mass, guaranteeing the presence of a trajectory with the lowest cost, we subtract the minimum cost among all sampled trajectories from the cost function (line 9). It is important to note that subtracting a constant does not affect the location of the minimum.

The second loop calculates the noise-weighted average over the K sampled trajectories (lines 10-11). This step incorporates the exploration noise into the trajectory selection process, allowing for exploration of different control actions. The third loop computes an optimal input sequence by selecting the trajectory with the least cost for the finite horizon of length H (lines 12-13). The top value of the resulting sequence is then provided as input to the actuators (line 14).

Subsequently, the entire input control sequence is left-shifted by one position (lines 15-16). To maintain the length of the buffer, the initial control value a_{init} is appended to the input control sequence (line 17). Finally, the states are updated based on the predictions provided by the ENN model, enabling the model to capture the dynamics of the system and adjust the control actions accordingly.

Algorithm 1 MPPI Controller

Input: ENN dynamics model $\hat{f}_\theta(s_t, a_t)$;
K: Number of samples, H: Length of horizon;
$(a_0, a_1, \ldots a_{H-1})$: Initial control sequence;
λ: Control hyper-parameter ;
Output: The control sequence $a_{t:t+H}$;

$s_0 \leftarrow GetStateEstimate()$;
for k = *0,1,...,K -1* **do**
  s $\leftarrow s_0$;
  Sample noise $\varepsilon^k = \{\epsilon_0^k, \epsilon_0^k, \ldots \epsilon_{H-1}^k\} \sim \mathbb{N}(\mu, \sigma)$;
  for t = *1,...,H* **do**
    $s_t \leftarrow \hat{f}_\theta(s_{t-1}, a_{t-1} + \epsilon_{t-1}^k)$;
    $Cost(\varepsilon^k)$ += $-reward$ defined by equation2 ;
$\beta \leftarrow min_k[Cost(\varepsilon^k)]$;
$\eta \leftarrow \sum_{k=0}^{K-1} exp(-\frac{1}{\lambda}(Cost(\varepsilon^k) - \beta))$;
for k = *0,1,...,K -1* **do**
  $\omega(\varepsilon^k) \leftarrow \frac{1}{\eta} exp(Cost(\varepsilon^k) - \beta)$;
for t = *0,1,...,H -1* **do**
  $a_t^* = a_t + \sum_{k=1}^{K} \omega(\varepsilon^k)\epsilon_t^k$;
SendToActuators(a_0);
for t = *0,1,...,H -1* **do**
  $a_{t-1} = a_t$;
$a_{t-1} = Initialize(a_{t-1})$;

F. Putting It All Together

We provide a summary of the working flow of the Model-Based Building Control MB^{2}C approach as follows. Initially, we gather a historical dataset D by employing a rule-based policy. The model parameters $\theta_1, \theta_2, \ldots, \theta_M$ for the ENN are randomly initialized. Subsequently, we train the ENN model using the collected dataset, utilizing Equation6 for optimization. Finally, we deploy the trained ENN model alongside our MPPI controller in a real building for HVAC control.

During each control execution, we start by obtaining the current state of the building from various sensors, such as a temperature sensor for zone temperature. Next, the MPPI controller samples the best action sequence using an H-horizon approach. The state is then propagated through the ENN model, solving the optimization problem defined in Equation 1. Finally, we execute the first action from the optimal action sequence in the building by appropriately adjusting the corresponding actuators. This iterative process enables continuous control and adaptation within the building system based on the learned dynamics from the ENN model.

During the operation of MB^{2}C in the building, we have the opportunity to collect building operation data, which consists of records of control action execution, denoted as $D(s_t, a_t, s_{t+1})$. This data includes information about the current state, the control action taken, and the resulting next state. We incorporate this newly collected data into a sliding window, which maintains a two-month history of data.

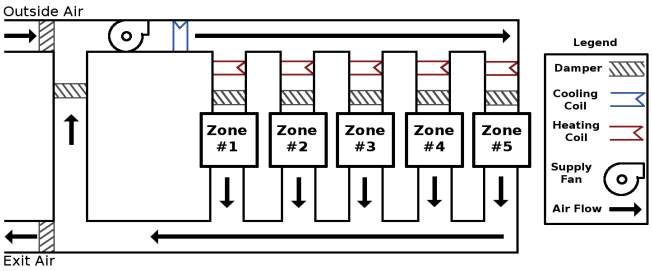


Fig. 7. HVAC Single Duct VAV Terminal Reheat Layout [12].

By doing so, we can adapt to the seasonality patterns present in the data, particularly weather-related information.

To update the ENN model, we randomly divide the training dataset into batches. These batches are then fed into the model, and we update the model weights through forward and backward propagation, employing techniques such as gradient descent. This process, known as one epoch training, involves traversing all the batches of data. We repeat this iterative training process for multiple epochs, typically 40 epochs in our current implementation, until the model converges and achieves the desired level of accuracy. The iterative in-situ updating process allows us to continuously improve the accuracy of our building dynamic model by incorporating new data and adjusting the model parameters. It enables the model to adapt to the changing dynamics and variations observed in the building's operation, leading to enhanced performance and control accuracy over time.

IV. Evaluation

In this section, we present a comprehensive set of experiments conducted in EnergyPlus to assess the performance of MB^{2}C along with three baseline methods. These experiments aim to evaluate the effectiveness of MB^{2}C in comparison to the baselines using a range of performance metrics.

A. Platform Setup

Building Example and its Dynamics Model in EnergyPlus In this study, we assess the performance of MB^{2}C in a specific building located in Fresno, California. The building has a total area of 463 m^2 and consists of a single floor with five distinct thermal zones. All four facades of the building are equipped with windows to facilitate natural lighting and ventilation. The HVAC system employed in our modeling is a single duct central cooling HVAC system with terminal reheat, as illustrated in Figure 7. The system begins with the supply fan located in the air handler unit (AHU), which is responsible for delivering conditioned air to the zones. The air supplied by the fan first passes through a cooling coil, where it is cooled to the minimum temperature required for each specific zone.

Before entering a zone, the air flows through a variable air volume (VAV) unit, which regulates and controls the amount of air directed into the zone. Terminal reheat is achieved through a heating coil, which increases the temperature of the air before it is discharged into the zone. Each zone is assigned a discharge setpoint temperature, and the VAV ensures that the air is heated to meet this temperature requirement for each respective zone. To maintain a constant static pressure within

the zones, a portion of the supplied air is mixed with the current air within the zone, while the excess air is exhausted out of the zone. The return air from each zone is then mixed in the return duct, and some of it may enter the economizer for further processing.

Since we cannot conduct control experiments in the real building, we utilize EnergyPlus version 8.6, a powerful building simulation software, to create a virtual building model. This allows us to conduct simulations using Typical Meteorological Year 3 (TMY3) weather data. In our implementation, we adhere to the default control logic of EnergyPlus for setting the setpoint of the AHU. Our focus is specifically on controlling the heating and cooling setpoints in the VAV boxes. By manipulating these setpoints, we can effectively regulate the thermal conditions within the simulated building.

EnergyPlus has emerged as a widely adopted tool for evaluating HVAC control algorithms [16], [17], [19], [24]. We have chosen EnergyPlus for several reasons. Firstly, due to practical constraints, we lack access to a physical building where we can conduct experiments. However, once we complete the training of the ENN model, MB^{2}C can be readily deployed in a real building. Secondly, EnergyPlus provides us with the convenience of generating a substantial amount of historical training data using a rule-based method. This data is instrumental in training the ENN model, enabling us to capture the dynamics of the building system accurately. Moreover, in order to compare MB^{2}C with model-free DRL approaches, it is essential to have a sizable training dataset. Model-free DRL methods typically require a large number of samples to achieve good performance, as they are not sample efficient. In our case, we require training data spanning 5200 days (equivalent to 14+ years), which would be impractical to obtain solely from real buildings. Lastly, EnergyPlus offers us the flexibility to evaluate the performance of various control algorithms across different locations, seasons, and weather profiles. This versatility allows us to gain insights into the robustness and adaptability of MB^{2}C in different environmental conditions, further enhancing our understanding of its effectiveness in real-world scenarios.

MB^{2}C System Components The MB^{2}C system, as illustrated in Figure 4, consists of two primary components: the building dynamics model ENN and the MPPI controller. Additionally, we incorporate a data storage mechanism to collect and update building operation data for in-situ model refinement. All three components are implemented using TensorFlow, a widely-used open-source machine learning library in Python. To establish a connection between EnergyPlus and MB^{2}C, we utilize the building control virtual testbed (BCVTB) [37]. BCVTB facilitates the interaction and communication between EnergyPlus and our MB^{2}C system. During each control cycle, we execute the control action by setting the temperature to a specific set point for each zone within our EnergyPlus building model.

B. Experiment Setting

The parameter settings for MB^{2}C are presented in Table III. The control timestep for HVAC control is set to 15 minutes, which is a commonly used interval in classic building control[36],[38]. Although using shorter timesteps can potentially improve the accuracy of building dynamics models, it is important to consider the practical limitations of HVAC equipment. According to EnergyPlus documentation, control periods shorter than 10-15 minutes can cause physical damage to equipment such as heat pumps[36].

TABLE III
PARAMETER SETTINGS IN MB^{2}C

Batch Size	512
Time Step for Control	15min
Train/Validation Split Ratio	80%/20%
Discount Factor γ	0.99
Learning Rate	0.001
Number of Hidden Layers	2
Number of Neurons for Each Layer	200
Number of Data Samples	1000
Length of Horizon	20

For training the ENN model, we utilize weather data from two different cities: Fresno, CA and Chicago, IL. These cities were chosen due to their distinct weather characteristics. Fresno experiences intense solar radiation and significant temperature variations, while Chicago is classified as having a hot-summer humid continental climate with four distinct seasons.

To evaluate the performance of MB^{2}C, we compare it against three baseline methods. All four control approaches are executed using the same weather data for simulation, ensuring a fair comparison of their performance in controlling the building's HVAC system.

1) Rule-based Method: We implement a rule-based method to generate training data and for comparison evaluation. This rule-based method follows our current campus building control policy. In this approach, we assign different zone temperature set-points, with each zone having separate heating and cooling set-points. During the warm-up stage, the heating set-point is set to 70°F, and the cooling set-point is set to 74°F. To ensure a comfortable range of temperature control, the cooling set-point is limited to a range of 72°F to 80°F, while the heating set-point is limited to a range of 65°F to 72°F. These limits help maintain a suitable indoor temperature within the building while allowing for energy-efficient operation.

2) Model-free DRL: We implement Proximal Policy Optimization (PPO)[28]as MF-RL based method for multi-zone control. One advantage of PPO is its stability and robustness to both hyperparameters and network architectures [28]. Moreover, PPO has shown superior performance compared to other policy gradient algorithms such as Natural Policy Gradients (NPG) [39] and Trust Region Policy Optimization (TRPO) [40].

3) Model-based DRL with RS: In the conventional model-based approach, we utilize a deterministic neural network to capture the building dynamics, allowing us to model the

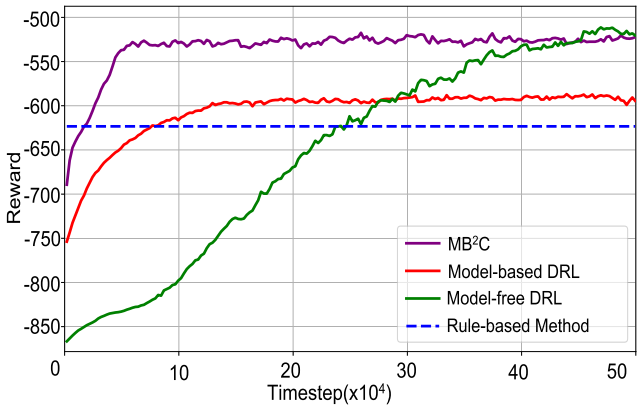


Fig. 8. MB^{2}C Achieves both Data-Efficiency and High Performance.

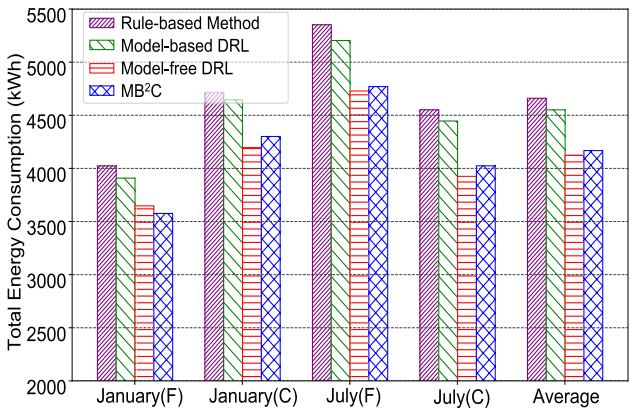


Fig. 9. Energy Consumption of MB^{2}C and the Other Baselines.

system's behavior accurately. Additionally, we employ the RS method to determine optimal heating and cooling setpoints. This combination of the neural network for building dynamics modeling and the RS method for setpoint selection has been proven effective in previous research [24] for single-zone HVAC control.

Both the Model-free DRL and Model-based DRL with RS methods share the same state, action, and reward definitions as the MB^{2}C approach. This ensures consistency in the control framework across these different algorithms. By maintaining the same state representation, action space, and reward structure, we enable fair comparisons and evaluations of their performance in controlling the HVAC system in the building.

C. Experiment Results

We evaluate and compare MB^{2}C with the aforementioned baselines using a comprehensive set of performance metrics. These metrics encompass convergence analysis, energy efficiency, and thermal comfort. Additionally, we conduct an in-depth analysis of MB^{2}C's performance, examining factors such as daily energy consumption for each zone, the effectiveness of its key components, and the impact of its parameter settings.

1) Convergence Analysis: We begin by examining the data efficiency of MB^{2}C and the other three baselines. In this analysis, we do not confine MB^{2}C to a sliding window of two months, as the MFRL method requires a substantial amount of training data. Figure 8 illustrates the accumulated reward for each control method over multiple episodes during the training process. Each episode corresponds to one month of data, equivalent to 2976 time-steps. The reward function is calculated at each time-step, and the reward shown in Figure 8 represents the cumulative reward for one episode, i.e., the sum of rewards over 2976 time-steps. The results depicted in Figure 8 demonstrate that the episode reward increases and eventually stabilizes as the number of training episodes increases. When the episode reward plateaus, it indicates that further improvements to the learned control policy are unlikely, signifying convergence of the training process.

As depicted in Figure 8, MB^{2}C outperforms the rule-based method after approximately 1.75×10^4 time-steps. At this stage, the ENN model is trained using offline historical data and deployed in real buildings, leveraging the MPPI controller for exploration to further enhance its performance. On the other hand, the model-based DRL and model-free DRL methods require 7.5×10^4 and 23.75×10^4 time-steps, respectively, to surpass the rule-based method. MB^{2}C achieves $4.28\times$ and $13.57\times$ greater data efficiency compared to the model-based DRL and model-free DRL methods, respectively.

In terms of convergence time, MB^{2}C converges faster than both the model-based DRL and model-free DRL methods. MB^{2}C achieves convergence at 4.75×10^4 time-steps, while the model-based DRL method requires 11.5×10^4 time-steps, and the model-free DRL method requires 50×10^4 time-steps. Therefore, MB^{2}C demonstrates $2.4\times$ and $10.52\times$ greater data efficiency than the model-based DRL and model-free DRL methods, respectively, while achieving comparable performance to the model-free DRL method.

2) Energy Efficiency: Figure9illustrates the energy consumption outcomes of the four control methods. The findings indicate that, on average, MB^{2}C achieves energy savings of 10.65% and 8.23% compared to the rule-based method and model-based DRL, respectively. In comparison to the model-free DRL method, MB^{2}C demonstrates comparable performance in terms of energy consumption. These energy savings are attributed to MB^{2}C's accurate modeling of complex building dynamics and its ability to identify optimal heating and cooling setpoints, resulting in more efficient HVAC operation.

The energy consumption varies across different seasons and cities, indicating the influence of weather conditions on HVAC usage. In Fresno, the building consumes 4770.04 kWh in July, which is 33.39% more energy compared to January when it consumes 3576.07 kWh. This discrepancy can be attributed to the outdoor air temperature range. In July, the range at Fresno is 15°C to 42°C, necessitating continuous cooling during daylight hours to maintain thermal comfort. Conversely, in January, the range is -1°C to 18°C, allowing for energy savings by utilizing outside air within the optimal range of thermal comfort.

In Chicago, the building consumes 4300.47 kWh in January, which is 6.86% more energy compared to July. The colder weather in January, with an outdoor air temperature range of -20°C to 15°C, leads to higher heating requirements and

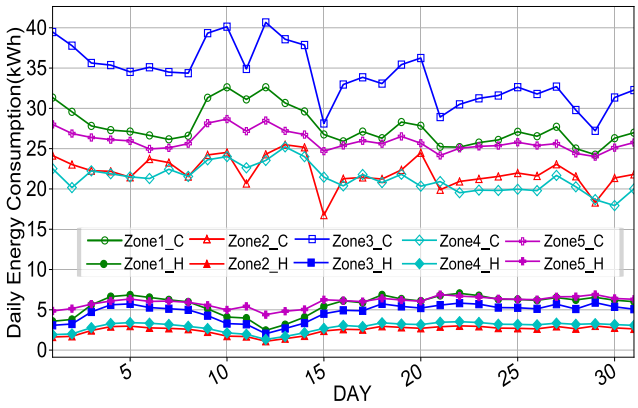


Fig. 10. Daily energy consumption for five zones.

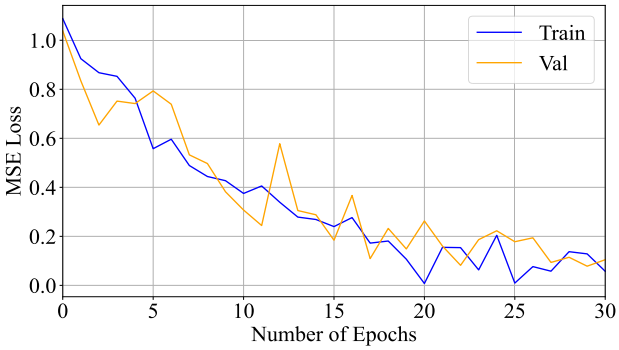


Fig. 11. ENN dynamics model loss.

thus increased energy consumption. In July, both Merced and Chicago experience a similar outdoor air temperature range of 15°C to 42°C and 15°C to 40°C, respectively. However, the energy consumption in Fresno is 18.53% higher than in Chicago. This difference can be attributed to the average day and night temperature variations. Fresno experiences larger fluctuations between day and night temperatures compared to Chicago, resulting in increased energy demand for maintaining thermal comfort throughout the day.

3) Thermal Comfort: TableIVprovides the average PMV values for all five zones in January and July, considering the weather data from Fresno and Chicago. It is observed that all four control methods effectively maintain the PMV values within the desired range of -0.7 to 0.7 for the majority of the time. The model-based method exhibits a slightly higher average violation rate of 1.97% compared to the other three methods. This is primarily due to the controller's exploration of random actions, which can occasionally result in suboptimal thermal comfort conditions. On the other hand, MB^{2}C achieves a notably low average violation rate by capitalizing on the enhanced accuracy of the ENN model and the improved effectiveness of the MPPI controller.

4) Neural Network Architecture: To investigate the impact of different neural network architectures on energy consumption and thermal comfort, we conduct experiments using July weather data from Fresno. Four neural networks were tested, each with a different number of hidden layers: 1, 2, 3, and 4. The results of these experiments, presented in Table V, demonstrate the energy consumption and thermal comfort achieved by MB^{2}C with each neural network configuration.

Analyzing the experimental outcomes, we observe that neural networks with more hidden layers generally provided better thermal comfort, as indicated by results closer to 0 on the thermal comfort scale. However, this improvement in comfort comes at the cost of higher energy consumption. The underlying reason is that MB^{2}C aims to strike a balance between energy consumption and thermal comfort, recognizing that increased energy usage can enhance people's perceived comfort.

Considering these findings, we select a neural network with 2 hidden layers for MB^{2}C. This choice is motivated by its ability to minimize energy consumption while still meeting the requirement for thermal comfort. By striking a suitable compromise between energy efficiency and comfort, MB^{2}C demonstrates optimal performance with this neural network configuration.

5) Effect of Ensemble Model Size: We conduct a series of experiments to evaluate the impact of different ensemble sizes on energy consumption and thermal comfort. Utilizing July weather data from Fresno, we test ensemble configurations with 3, 4, 5, and 6 models to determine the optimal balance between energy efficiency and thermal comfort. The results, detailed in TableVI, illustrate the performance variations across different ensemble sizes. The analysis reveals that increasing the number of ensemble models initially leads to improvements in both energy consumption and thermal comfort. Specifically, transitioning from 3 to 5 models, we observe a noticeable enhancement in thermal comfort metrics with a concurrent reduction in energy consumption. This improvement plateaus beyond five models, as evidenced by the minimal changes when moving to 6 models. The decision to employ 5 models in our weighted ensemble is thus underpinned by their collective ability to achieve significant reductions in energy use while optimizing for thermal comfort. This ensemble size offers the best compromise between computational efficiency and the model's predictive accuracy and control capabilities.

6) ENN Dynamics Model Loss: Figure11illustrates the loss curves of our ENN model used for predicting building dynamics in an HVAC control system. The x-axis represents the number of training epochs, while the y-axis shows the loss value, typically reflecting the mean squared error between the model's predictions and the actual data. Two curves are depicted: one for the training loss and another for the validation loss. As the number of epochs increases, both curves exhibit a downward trend, indicating that the model is learning and improving its predictive accuracy over time. The training loss curve shows a consistent decrease, suggesting the model's increasing fit to the training data. The validation loss curve decreases alongside the training loss, which points to the model's generalization capabilities.

7) Daily Energy Consumption for Five Zones: We conduct an analysis of the daily energy consumption of MB^{2}C in July for five zones in Fresno. Figure 10 illustrates the heating and cooling energy recorded for each zone on a daily basis. The top five hollow line symbols represent the trend of cooling

TABLE IV
THERMAL COMFORT STATISTICAL RESULTS FOR RULE-BASED, MODEL-BASED, MODEL-FREE AND MB^{2}C SCHEMES

Location	Comfort	Metric	Rule-based method		Model-based method		Model-free based method		MB^{2}C	
			January	July	January	July	January	July	January	July
Fresno	PMV	Mean	-0.36	-0.20	-0.32	-0.19	-0.11	-0.03	-0.04	0.13
		Std	0.26	0.36	0.31	0.34	0.15	0.18	0.11	0.14
		Violation rate	1.22%	1.51%	2.12%	1.71%	0	0.14%	0.40%	0.58%
Chicago	PMV	Mean	-0.17	-0.30	-0.26	-0.18	-0.25	0.07	-0.23	0.05
		Std	0.23	0.33	0.24	0.31	0.17	0.19	0.07	0.20
		Violation rate	1.20%	2.04%	1.9%	2.13%	0.95%	0	0.46%	1.23%

TABLE V
EFFECT OF DIFFERENT NETWORK ARCHITECTURE

Number of hidden layers	Energy Consumption (kWh)	Thermal Comfort (PMV)
1	4911	0.11
2	4820	0.15
3	4988	0.09
4	5032	0.08

TABLE VI
EFFECT OF ENSEMBLE MODEL SIZE

Number of Ensemble Models	Energy Consumption (kWh)	Thermal Comfort (PMV)
3	4915	0.21
4	4911	0.17
5	4820	0.15
6	4819	0.16

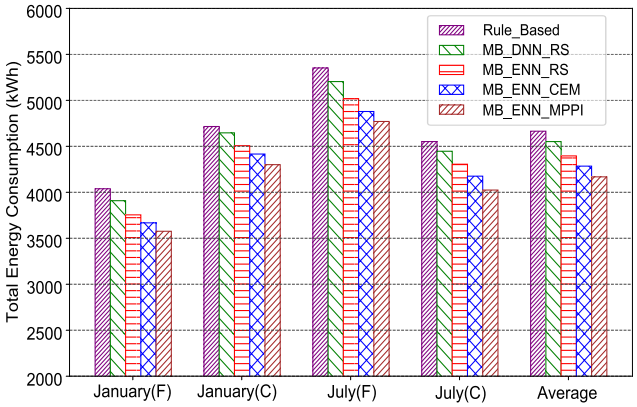


Fig. 12. Performance decomposition.

energy for the respective zones, while the bottom five solid lines indicate the trend of heating energy. It is worth noting that the third zone exhibits higher energy consumption compared to the other zones. This disparity arises because the third zone is south-oriented, resulting in more direct sunlight exposure throughout the day.

Additionally, we observe that both heating and cooling are required on certain days due to significant temperature variations between day and night. During daylight hours, with an average outdoor temperature of 38°C, more energy is needed for cooling purposes. Conversely, during nighttime hours, with an average outdoor temperature of 15°C, some heating is necessary to maintain thermal comfort, particularly as our simulations encompass an office-like environment where students occasionally work at night. These findings highlight the dynamic nature of energy requirements in response to changing external conditions and the specific characteristics of each zone within the building.

8) Performance Decomposition: We implement three variations of MB^{2}C, each employing a different control method: RS (MB_ENN_RS), CEM (MB_ENN_CEM), and MPPI (MB_ENN_MPPI). In addition, we compare these approaches with the rule-based method and an existing model-based DRL method (MB_DNN_RS).

In the MB_ENN_CEM version, we employ the Cross-entropy method (CEM) [41] as the controller. Initially, it starts with the RS method and performs multiple iterations $m \in \{0 \dots M\}$ of action sampling at each time step. The top J highest-scoring action sequences from each iteration are used to update and refine the mean and variance of the sampling distribution for the next iteration. After M iterations, the optimal heating and cooling actions are determined as the resulting mean of the action distribution.

Figure 12 showcases the energy consumption of the four methods across two different months and two different locations (Fresno and Chicago). Note that all the evaluated methods maintained thermal comfort within the -0.7 to 0.7 PMV comfort range. Comparing the results with the rule-based method, MB_DNN_RS only achieves a modest energy savings of 2.42%. However, when we replace the building dynamics model in MB_DNN_RS with the proposed model MB_ENN_RS, an additional 3.34% energy savings can be achieved, highlighting the effectiveness of the proposed model. Furthermore, by replacing the RS method with the CEM method and the MPPI method, both using the proposed model, we observe energy savings of 2.39% and 4.89%, respectively. These findings underscore the efficiency of the MPPI controller in maximizing energy savings.

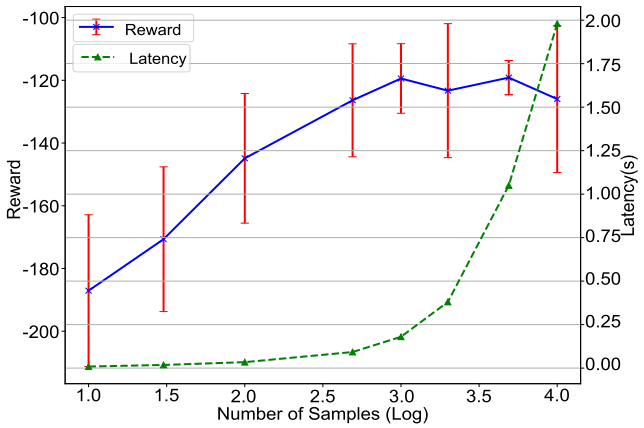


Fig. 13. Samples of MPPI controller.

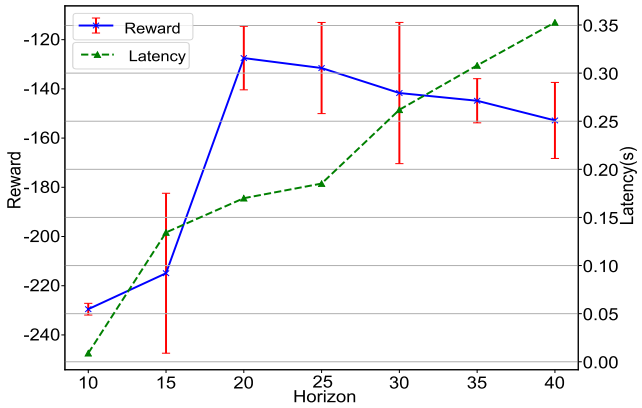


Fig. 14. Horizon of MPPI controller.

9) Parameter Setting: MB^{2}C relies on two critical parameters that can significantly impact its performance.

a) The number of samples in the MPPI algorithm: Figure13provides a detailed analysis of the MPPI controller's performance as the number of sample trajectories is varied. To assess the impact of different trajectory numbers (10, 30, 100, 500, 1000, 2000, 5000, 10000), we conduct multiple runs of the MPPI controller using the ground truth model. Each configuration is repeated 10 times, allowing us to calculate the mean and standard reward for each trajectory count. From the results depicted in Figure 13, we observe a distinct pattern: the reward increases rapidly as the number of trajectories grows until reaching approximately 1000 trajectories (power of 3 in the figure). Beyond this point, the reward improvement becomes more gradual, suggesting that the MPPI algorithm has converged. This finding indicates that 1000 trajectories are sufficient to achieve near-optimal performance in terms of reward.

In addition to evaluating reward, we also examine the latency associated with selecting an action under different trajectory counts. It becomes evident that the latency increases exponentially as the number of trajectories increases. This observation highlights the trade-off between computational efficiency and performance. Taking into account the trade-off between achieving the best reward and minimizing latency, we determine that 1000 trajectories strike an optimal balance. This choice ensures a substantial reward improvement while keeping the computational burden manageable.

b) The length of horizon in the MPC process: The horizon parameter in Algorithm 1 refers to the number of steps to look ahead in the MPC process. To assess the impact of different horizon lengths H on the performance of the MPPI Controller, we conduct experiments and analyze the results. Figure 14 provides an overview of the reward achieved by the MPPI Controller for different horizon lengths. As the horizon length increases, we observe a corresponding increase in the reward, reaching its highest value when the horizon length is 20. However, as we further increase the horizon length beyond 20, the reward starts to decrease.

This pattern can be explained by the trade-off between considering future dynamics and the accumulation of prediction errors. With a small horizon, the controller tends to make more immediate and greedy actions, overlooking the potential impact of future dynamics. On the other hand, a large horizon may result in worse actions due to the accumulation of prediction errors as the horizon becomes longer. Considering both the prediction errors and action performance, as well as the desire for short latency, we choose a horizon length of 20. This selection strikes a balance between accounting for future dynamics and minimizing the negative effects of prediction errors, ultimately leading to improved action performance and overall reward.

TABLE VII
EXECUTION OVERHEAD

	Model Inference	Action Selectction	Total Latency
Rule-based Method	N/A	N/A	0.44ms
Model-based DRL	71.76ms	21.41ms	93.17ms
Model-free DRL	N/A	N/A	2.54 ms
MB^{2}C	294.72ms	38.46ms	333.18ms

10) Execution Overhead: TableVIIprovides a comparison of the execution latency for different HVAC control methods during a single timestep. The latency of the model-based method and MB^{2}C encompasses two components: the prediction latency of the building dynamics model and the action selection latency incurred by the controller. In contrast, the rule-based method and model-free method have significantly smaller latencies since they do not involve a dynamics model or controller.

For MB^{2}C, the latency of the model (ENN model) is 294.72ms, while the latency of the controller is 38.46ms. Both values are higher compared to the existing model-based method. The increased latency in MB^{2}C can be attributed to two main factors. Firstly, the ENN model requires additional time to evaluate prediction results and calculate the weights of the models within the ensemble learning framework for building dynamics modeling. Secondly, the MPPI controller needs to compute noise-related weights and evaluate various action sequences.

However, despite the higher latency, it is important to note that MB^{2}C is still capable of generating an executable control

action within one second. This is a significant achievement, particularly when considering that the typical control cycle is set to 15 minutes. Therefore, MB^{2}C effectively meets the requirement of generating control actions promptly, allowing for efficient HVAC control in real-time scenarios.

V. Related Work

There are a number of approaches to solve HVAC energy optimization problems in the literature, including model predictive control [12], [13], [42], [43], Model-free RL for HVAC control [17], [19], [20], [44], [45], [46], Model-based DRL for HVAC control [24] and Multi-agent DRL for HVAC control [47], [48], [49].

A. MPC for HVAC Control

MPC is an iterative approach that solves an optimal control problem by considering a receding time horizon. In the context of HVAC control, previous research has proposed various MPC frameworks aimed at minimizing energy consumption while ensuring occupant comfort. For instance, [12] introduces an MPC approach specifically designed for HVAC control, focusing on energy minimization and comfort constraints. More recently, a novel MPC framework called OFFICE [13] has been developed, which addresses the trade-off between energy cost and occupant comfort in building management. OFFICE employs a gray-box approach, utilizing a parametrized first-principles model, where the model parameters are dynamically learned and updated over time. Building on existing MPC frameworks for HVAC control, the work [42] introduces an economic model predictive control (EMPC) approach that optimizes energy consumption and indoor comfort through a lattice piecewise linear approximation for the PMV index.

In contrast, our approach adopts a black-box methodology, where a neural network learns the complex relationships between system inputs and outputs from scratch. Furthermore, the MPC controller employed in our method differs from OFFICE. While OFFICE employs an interior-point method based on a differentiable function to determine the optimal solution, we utilize an MPPI controller. The MPPI controller incorporates sample noise as an exploration mechanism around default values, enabling it to search for the best optimization solution.

B. Model-free DRL for HVAC Control

Reinforcement Learning has been applied to many areas [50], [51], [52], [53], [54], [55], e.g., smart city [52], mobile application usage prediction [55], autonomous ground vehicle (AGV) parking [54], sensor configuration [50] and obstacle avoidance [53].

MFRL techniques have emerged as promising approaches for achieving optimal HVAC controls. These schemes involve the agent actively interacting with the environment and learning the policy through extensive trial and error. For instance, RL has been effectively utilized to determine thermostat set-points that strike a balance between occupant comfort and energy efficiency [20]. Furthermore, a DRL-based control method is successfully implemented and deployed in a real-life office building, specifically for managing radiant heating systems [19]. The research has also explored a comprehensive building control framework that encompasses HVAC, lighting, window opening, and blind inclination, employing the branching dueling Q-network (BDQ) algorithm [17], [56]. However, despite these advancements, the practical application of RL in HVAC control faces challenges related to sample complexity. This complexity arises from the substantial training time required to develop control strategies, particularly when dealing with tasks characterized by a large state-action space. Le et al. [57] introduce a DRL-based control method for air free-cooled data centers in tropical regions. Vazquez-Canteli et al. [58] focus on developing a multi-agent RL implementation to optimize load shaping in grid-interactive connected buildings. Zhang et al. [19] successfully implement and deploy a DRL-based control method specifically for radiant heating systems in a real-life office building. Gao et al. [45] propose an approach based on deep deterministic policy gradients (DDPGs) to learn thermal comfort control policies. While these studies significantly enhance HVAC control performance, it is important to note that they predominantly concentrate on improving the HVAC subsystem alone. On the other hand, Gnu-RL [16] takes a different direction by utilizing a differentiable MPC policy, which incorporates domain knowledge related to planning and system dynamics. This unique approach makes Gnu-RL both data-efficient and interpretable. However, it is worth mentioning that Gnu-RL assumes the local linearization of water-based radiant heating system dynamics, and its effectiveness may not extend to more complex problems, such as the one addressed in our study.

C. Model-Based DRL for HVAC Control and Complementary Model-Free Approaches

In an effort to tackle the challenge of sample complexity, researchers have turned to model-based RL techniques for HVAC control [24], [59]. In their study, Zhang et al. [24] put forth an innovative MBRL approach that involves training a neural network to learn the intricate dynamics of the system. Subsequently, they incorporate the acquired system dynamics into an MPC framework, utilizing the rolling horizon optimization method to execute control actions. Chen et al. [59] propose a novel learning-based control strategy, named MBRL-MC, for the HVAC system, which synthesizes MBRL with MPC [59]. By initially learning a thermal dynamic model of the zone through supervised learning, and subsequently designing a NN planning framework that integrates RL with MPC, this approach diverges from traditional MBRL methods. It circumvents the compounding error issue by avoiding the imitation of MPC's random shooting outcomes and eschews the bootstrapping technique in critical network updates for enhanced stability. While MBRL methods exhibit promising performance in scenarios where the action and state dimensions are low, such as in single-zone buildings, they often fall short of achieving the same

level of performance as model-free methods when applied to multi-zone buildings with high state and action dimensions. This discrepancy can be attributed to the inherent challenges in accurately modeling and capturing the complexities of multi-zone HVAC systems, which hinders the effectiveness of the MBRL approach.

Some complementary model-free approaches offer promising solutions for HVAC control, particularly in complex and dynamic environments. Michailidis et al.[60] showcase a proactive control strategy in a high-inertia building, leveraging simulation-assisted methodologies to optimize energy use and thermal comfort, demonstrating the effectiveness of MFRL in environments with complex dynamics. Similarly, Baldi et al. [61] introduce Parametrized Cognitive Adaptive Optimization (PCAO) for designing efficient "plug-and-play" building optimization and control (BOC) systems. PCAO excels by learning optimal BOC strategies with minimal human input, showing significant improvements in energy efficiency and thermal comfort compared to traditional methods. These studies offer invaluable insights aligned with our research objectives. The innovative approaches they propose, particularly in using estimators for reducing data requirements and increasing iteration efficiency, present a compelling methodology that complements our study's focus.

D. Multi-Agent DRL for HVAC Control

To address the challenges posed by unknown thermal dynamics models and parameter uncertainties, such as outdoor temperature, electricity price, and the number of occupants, researchers propose innovative HVAC control algorithms for multi-zone commercial buildings. In one approach, presented by [47], they utilize a multi-agent deep reinforcement learning (MADRL) framework [48] with an attention mechanism [62]. This approach enables flexible and scalable coordination among different agents, allowing for effective control in the presence of complex dynamics. Another study, conducted by [49], leverages a multi-agent reinforcement learning algorithm to tackle the optimization problem of minimizing building HVAC energy consumption while ensuring comfort constraints are met. This is achieved through dynamic adjustments of both the building and chiller set-points. The work [63] introduces a novel occupant-centric approach to multi-zone HVAC control that leverages MADRL to intelligently schedule cooling and heating setpoints, considering stochastic occupant behavior models, such as dynamic clothing insulation adjustments, metabolic rates, and occupancy patterns. However, it is important to note that both approaches, [47] and [49], are based on model-free Multi-agent DRL, which typically requires a significant amount of training data spanning several years to achieve satisfactory results.

VI. Discussion

A. Building Model Calibration

Currently, our evaluation of existing control methods involves utilizing the five-zone building model available in EnergyPlus. However, we have not yet calibrated this model due to the absence of historical operational data for the specific building. The buildings implemented in EnergyPlus are based on first principles thermodynamic models, which should provide a performance similar to that of a real building. By using the same EnergyPlus building model as a ground truth, we can fairly compare the performance of different control methods. It's important to note that for the evaluation conducted in our paper, this approach provides a reasonable and fair comparison for "a particular building." If the proposed MB^{2}C were to be deployed in a real building, we would need to first learn the dynamics model using historical data from an actual building. Once the dynamics model is developed, we can deploy it in the real building for control purposes. If we were to conduct simulations to test MB^{2}C prior to real-world deployment, we would need to create a calibrated EnergyPlus model that accurately represents the target building [17], [19]. This calibration process ensures that the simulated results align closely with the behavior and characteristics of the specific building under consideration.

B. Occupancy and Weather Model

In MB^{2}C, we utilize the ground-truth values of weather and occupancy for the ENN dynamics model. It is important to note that MB^{2}C may exhibit a slightly optimistic bias since we assume perfect prediction for weather and occupancy. While errors in prediction can impact the controller's performance, we believe that the overall deviation from actual results will not be significant, taking into account the model prediction errors. There are two reasons for this: Firstly, the existing occupancy and weather prediction models [64], [65] demonstrate minimal prediction errors. This indicates a high level of accuracy in the predictions, further supporting the reliability of our approach. Secondly, the MPPI controller generates an optimal trajectory over the planning horizon. It only considers the first optimal action and recalculates at each time step based on new observations. This adaptive approach effectively mitigates the impact of model errors over time, preventing their compounding effects.

VII. Conclusion

This paper presents MB^{2}C, an innovative model-based DRL HVAC control system designed for multi-zone buildings. Our approach involves developing a novel building dynamics model, which consists of an ensemble of multiple neural network models conditioned on the environment. To perform HVAC control, we employ a model predictive path integral control method. In our study, we conduct a comprehensive performance comparison of MB^{2}C with rule-based methods, as well as state-of-the-art model-based and model-free DRL schemes. The results demonstrate that MB^{2}C outperforms rule-based approaches, achieving energy savings of 10.65%. Moreover, MB^{2}C demonstrates comparable performance to state-of-the-art model-based and model-free DRL methods while ensuring the thermal comfort of occupants, and in some cases, even improving it. Notably, one significant advantage of MB^{2}C is its remarkable reduction in the required training set. By leveraging MB^{2}C, we can achieve a substantial reduction

in training data, with an order of magnitude decrease of 10.52 times, without compromising performance.

Despite these promising results, we recognize certain limitations within our study. The assumptions of perfect prediction for weather and occupancy may not fully capture the unpredictable nature of real-world conditions. Furthermore, the scalability of MB^{2}C to various building configurations and climates remains an area for further exploration. Moving forward, future research will aim to refine the predictive accuracy of our model under varying environmental conditions and explore the integration of stochastic models to better adapt to real-time changes in occupancy and weather. We also plan to investigate the application of MB^{2}C across a broader spectrum of building types and environmental conditions, further validating its versatility and effectiveness in improving energy efficiency and occupant comfort. These steps will contribute to the ongoing development of intelligent HVAC control systems, pushing the boundaries of what is achievable in energy conservation and environmental sustainability.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] X. Ding, W. Du, and A. E. Cerpa, "MB2C: Model-based deep reinforcement learning for multi-zone building control," in *Proc. 7th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2020, pp. 50–59.

[2] N. Klepeis et al., "The national human activity pattern survey (NHAPS): A resource for assessing exposure to environmental pollutants," *J. Exposure Sci. Environ. Epidemiol.*, vol. 11, pp. 231–252, Jul. 2001.

[3] Z. Xu, Q.-S. Jia, and X. Guan, "Supply demand coordination for building energy saving: Explore the soft comfort," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 656–665, Apr. 2015.

[4] Q.-S. Jia, H. Wang, Y. Lei, Q. Zhao, and X. Guan, "A decentralized stay-time based occupant distribution estimation method for buildings," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1482–1491, Oct. 2015.

[5] K. Yang, Y. Chen, and W. Du, "OrchLoc: In-orchard localization via a single LoRa gateway and generative diffusion model-based fingerprinting," in *Proc. 22nd Annu. Int. Conf. Mobile Syst., Appl. Services*, Jun. 2024, pp. 304–317.

[6] K. Yang, Y. Chen, X. Chen, and W. Du, "Link quality modeling for LoRa networks in orchards," in *Proc. 22nd Int. Conf. Inf. Process. Sensor Netw.*, May 2023, pp. 27–39.

[7] S. Bengea, A. Kelman, F. Borrelli, R. Taylor, and S. Narayanan, "Model predictive control for mid-size commercial building HVAC: Implementation, results and energy savings," in *Proc. 2nd Int. Conf. Building Energy Environ.*, 2012, pp. 979–986.

[8] W. Goetzler, R. Shandross, J. Young, O. Petritchenko, D. Ringo, and S. McClive, "Energy savings potential and RD&D opportunities for commercial building hvac systems," Navigant Consulting, Burlington, MA, USA, Tech. Rep. DOE/EE-1703; 7849, 2017.

[9] D. Minoli, K. Sohraby, and B. Occhiogrosso, "IoT considerations, requirements, and architectures for smart buildings—Energy optimization and next-generation building management systems," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 269–283, Feb. 2017.

[10] G. Bedi, G. K. Venayagamoorthy, and R. Singh, "Development of an IoT-driven building environment for prediction of electric energy consumption," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4912–4921, Jun. 2020.

[11] J. Salpakari and P. Lund, "Optimal and rule-based control strategies for energy flexibility in buildings with PV," *Appl. Energy*, vol. 161, pp. 425–436, Jan. 2016.

[12] A. Beltran and A. E. Cerpa, "Optimal HVAC building control with occupancy prediction," in *Proc. 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings*, Nov. 2014, pp. 168–171.

[13] D. A. Winkler, A. Yadav, C. Chitu, and A. E. Cerpa, "OFFICE: Optimization framework for improved comfort & efficiency," in *Proc. 19th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2020, pp. 265–276.

[14] N. N. Kota, J. M. House, J. S. Arora, and T. F. Smith, "Optimal control of HVAC systems using DDP and NLP techniques," *Optim. Control Appl. Methods*, vol. 17, no. 1, pp. 71–78, Jan. 1996.

[15] X. Lü, T. Lu, C. J. Kibert, and M. Viljanen, "Modeling and forecasting energy consumption for heterogeneous buildings using a physical–statistical approach," *Appl. Energy*, vol. 144, pp. 261–275, Apr. 2015.

[16] B. Chen, Z. Cai, and M. Bergés, "Gnu-RL: A practical and scalable reinforcement learning solution for building HVAC control using a differentiable MPC policy," *Frontiers Built Environ.*, vol. 6, pp. 1–18, Nov. 2020.

[17] X. Ding, W. Du, and A. Cerpa, "OCTOPUS: Deep reinforcement learning for holistic smart building control," in *Proc. 6th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2019, pp. 326–335.

[18] X. Ding, A. Cerpa, and W. Du, "Exploring deep reinforcement learning for holistic smart building control," *ACM Trans. Sensor Netw.*, vol. 20, no. 3, pp. 1–28, May 2024.

[19] Z. Zhang and K. P. Lam, "Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system," in *Proc. 5th Conf. Syst. Built Environ.* New York, NY, USA: Association for Computing Machinery, Nov. 2018, pp. 148–157.

[20] J. Y. Park and Z. Nagy, "HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points," in *Proc. 11th ACM Int. Conf. Future Energy Syst.*, Jun. 2020, pp. 434–437.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[22] Z. An, X. Ding, A. Rathee, and W. Du, "CLUE: Safe model-based RL HVAC control using epistemic uncertainty estimation," in *Proc. 10th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2023, pp. 149–158.

[23] Z. An, X. Ding, and W. Du, "Go beyond black-box policies: Rethinking the design of learning agent for interpretable and verifiable HVAC control," 2024, *arXiv:2403.00172*.

[24] C. Zhang, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Building HVAC scheduling using reinforcement learning via neural network based model approximation," in *Proc. 6th ACM Int. Conf. Syst. Energy-Efficient Build. Cities Transp.*, 2019, pp. 287–296.

[25] S. Goyal and P. Barooah, "A method for model-reduction of non-linear thermal dynamics of multi-zone buildings," *Energy Buildings*, vol. 47, pp. 332–340, Apr. 2012.

[26] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7559–7566.

[27] G. Williams et al., "Information theoretic MPC for model-based reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1714–1721.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[29] P. O. Fanger, "Thermal comfort. analysis and applications in environmental engineering," *Thermal Comfort. Anal. Appl. Environ. Eng.*, May 1970.

[30] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.

[31] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.

[32] S. Levine. *Model-based Reinforcement Learning*. Accessed: Oct. 4, 2020. [Online]. Available: http://rail.eecs.berkeley.edu/deeprlcourse/

[33] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Proc. Conf. Robot Learn.*, 2020, pp. 1101–1112.

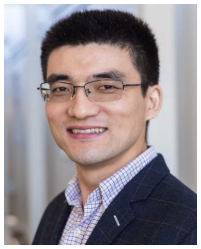
[34] *Thermal Environmental Conditions for Human Occupancy*, Standard 55-2004, ASHRAE Inc, 2004.

[35] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[36] M. Avci, M. Erkoc, A. Rahmani, and S. Asfour, "Model predictive HVAC load control in buildings using real-time electricity pricing," *Energy Buildings*, vol. 60, pp. 199–209, May 2013.
[37] M. Wetter, "Co-simulation of building energy and control systems with the building controls virtual test bed," *J. Building Perform. Simul.*, vol. 4, no. 3, pp. 185–203, Sep. 2011.
[38] Y. Ma, J. Matuško, and F. Borrelli, "Stochastic model predictive control for building HVAC systems: Complexity and conservatism," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 101–116, Jan. 2015.
[39] S. M. Kakade, "A natural policy gradient," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001, pp. 1–8.
[40] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 37, Lille, France, 2015, pp. 1889–1897.
[41] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L'Ecuyer, "The cross-entropy method for optimization," in *Handbook of Statistics*. Amsterdam, The Netherlands: Elsevier, 2013.
[42] H. Li, J. Xu, Q. Zhao, and S. Wang, "Economic model predictive control in buildings based on piecewise linear approximation of predicted mean vote index," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–12, Jun. 2004.
[43] B. Sun, P. B. Luh, Q.-S. Jia, Z. Jiang, F. Wang, and C. Song, "Building energy management: Integrated control of active and passive heating, cooling, lighting, shading, and ventilation systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 588–602, Jul. 2013.
[44] B. Sun, P. B. Luh, Q.-S. Jia, and B. Yan, "Event-based optimization within the Lagrangian relaxation framework for energy savings in HVAC systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1396–1406, Oct. 2015.
[45] G. Gao, J. Li, and Y. Wen, "DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8472–8484, Sep. 2020.
[46] A. H. Hosseinloo, S. Nabi, A. Hosoi, and M. A. Dahleh, "Data-driven control of COVID-19 in buildings: A reinforcement-learning approach," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–9, Sep. 2004.
[47] L. Yu et al., "Multi-agent deep reinforcement learning for HVAC control in commercial buildings," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 407–419, Jan. 2021.
[48] Z. Ding, T. Huang, and Z. Lu, "Learning individually inferred communication for multi-agent cooperation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 22069–22079.
[49] S. Nagarathinam, V. Menon, A. Vasan, and A. Sivasubramaniam, "MARCO–multi-agent reinforcement learning based COntrol of building HVAC systems," in *Proc. 11th ACM Int. Conf. Future Energy Syst.*, Jun. 2020, pp. 57–67.
[50] F. Fraternali, B. Balaji, Y. Agarwal, and R. K. Gupta, "ACES: Automatic configuration of energy harvesting sensors with reinforcement learning," *ACM Trans. Sensor Netw.*, vol. 16, no. 4, pp. 1–31, Nov. 2020.
[51] M. Liu, X. Ding, and W. Du, "Continuous, real-time object detection on mobile devices without offloading," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 976–986.
[52] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018.
[53] K. Wang, C. Mu, Z. Ni, and D. Liu, "Safe reinforcement learning and adaptive optimal control with applications to obstacle avoidance problem," *IEEE Trans. Autom. Sci. Eng.*, 2004.
[54] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, and S. Chai, "Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver," *IEEE Trans. Autom. Sci. Eng.*, pp. 1633–1647, Jun. 2022.
[55] Z. Shen, K. Yang, W. Du, X. Zhao, and J. Zou, "DeepAPP: A deep reinforcement learning framework for mobile application usage prediction," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, Nov. 2019, pp. 153–165.
[56] Y. Lei et al., "A practical deep reinforcement learning framework for multivariate occupant-centric control in buildings," *Appl. Energy*, vol. 324, Oct. 2022, Art. no. 119742.
[57] D. V. Le, R. Wang, Y. Liu, R. Tan, Y.-W. Wong, and Y. Wen, "Deep reinforcement learning for tropical air free-cooled data center control," *ACM Trans. Sensor Netw.*, vol. 17, no. 3, pp. 1–28, Aug. 2021.
[58] J. R. Vazquez-Canteli, G. Henze, and Z. Nagy, "MARLISA: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings," in *Proc. 7th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2020, pp. 170–179.
[59] L. Chen, F. Meng, and Y. Zhang, "MBRL-MC: An HVAC control approach via combining model-based deep reinforcement learning and model predictive control," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19160–19173, Oct. 2022.
[60] I. T. Michailidis, S. Baldi, M. F. Pichler, E. B. Kosmatopoulos, and J. R. Santiago, "Proactive control for solar energy exploitation: A German high-inertia building case study," *Appl. Energy*, vol. 155, pp. 409–420, Oct. 2015.
[61] S. Baldi, I. Michailidis, C. Ravanis, and E. B. Kosmatopoulos, "Model-based and model-free 'plug-and-play' building energy efficient control," *Appl. Energy*, vol. 154, pp. 829–841, Sep. 2015.
[62] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proc. 36th Int. Conf. Mach. Learn.* K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, Jun. 2019, pp. 2961–2970.
[63] X. Liu, Y. Wu, and H. Wu, "Enhancing HVAC energy management through multi-zone occupant-centric approach: A multi-agent deep reinforcement learning solution," *Energy Buildings*, vol. 303, Jan. 2024, Art. no. 113770.
[64] H. Rajabi, X. Ding, W. Du, and A. Cerpa, "TODOS: Thermal sensOr data-driven occupancy estimation system for smart buildings," in *Proc. 10th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2023, pp. 198–207.
[65] H. Rajabi, Z. Hu, X. Ding, S. Pan, W. Du, and A. Cerpa, "MODES: Multi-sensor occupancy data-driven estimation system for smart buildings," in *Proc. 13th ACM Int. Conf. Future Energy Syst.*, Jun. 2022, pp. 228–239.

Xianzhong Ding received the B.S. degree in computer science from Taishan University in 2014, the M.S. degree in computer science from Shandong University, China, in 2018, and the Ph.D. degree in computer science from the University of California, Merced, CA, USA, in 2023. He is currently a Post-Doctoral Researcher with the Lawrence Berkeley National Laboratory. His research interests include cyber-physical systems, resource optimization, and mobile computing.

Alberto Cerpa (Member, IEEE) received the Engineering degree in electrical engineering from Buenos Aires Institute of Technology, Argentina, in 1995, the M.S. degree in electrical engineering and the M.S. degree in computer science from the University of Southern California (USC), in 1998 and 2000, respectively, and the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), in 2005. He is currently an Associate Professor and the Graduate Group Chair of the University of California at Merced, Merced. His research interests include wireless sensor networks, embedded networked systems, cyber-physical systems, computer networks, and operating systems.



Wan Du (Member, IEEE) received the B.E. and M.S. degrees in electrical engineering from Beihang University, China, in 2005 and 2008, respectively, and the Ph.D. degree in electronics from the University of Lyon (École Centrale de Lyon), France, in 2011. He was a Research Fellow with Nanyang Technological University, Singapore, from 2012 to 2017. He is currently an Assistant Professor with the University of California at Merced, Merced. His research interests include the Internet of Things, distributed networking systems, and mobile computing.