Fast Human-in-the-Loop Control for HVAC Systems via Meta-Learning and Model-Based Offline Reinforcement Learning

Liangliang Chen, *Student Member, IEEE*, Fei Meng, *Student Member, IEEE*, and Ying Zhang, *Senior Member, IEEE*

***Abstract*—Reinforcement learning (RL) methods can be used to develop a controller for the heating, ventilation, and air conditioning (HVAC) systems that both saves energy and ensures high occupants' thermal comfort levels. However, the existing works typically require on-policy data to train an RL agent, and the occupants' personalized thermal preferences are not considered, which is limited in the real-world scenarios. This paper designs a high-performance model-based offline RL algorithm for personalized HVAC systems. The proposed algorithm can quickly adapt to different occupants' thermal preferences with a few thermal feedbacks, guaranteeing the high occupants' personalized thermal comfort levels efficiently. First, we use a meta-supervised learning algorithm to train an occupant's thermal preference model. Then, we train an ensemble neural network to predict the thermal states of the considered zone. In addition, the obtained ensemble networks can indicate the regions in the state and action spaces covered by the offline dataset. With the personalized thermal preference model updated via meta-testing, model-based RL is used to derive the optimal HVAC controller. Since the proposed algorithm only requires offline datasets and a few online thermal feedbacks for training, it contributes to a more practical deployment of the RL algorithm to HVAC systems. We use the ASHRAE database II to verify the effectiveness and advantage of the meta-learning algorithm for modeling different occupants' thermal preferences. Numerical simulations on the EnergyPlus environment demonstrate that the proposed algorithm can guarantee personalized thermal preferences with a slight increase of power consumption of 1.91% compared with the model-based RL algorithm with on-policy data aggregation.**

***Index Terms*—HVAC systems, model-based offline reinforcement learning, meta-learning, human-in-the-loop control.**

I. Introduction

The HVAC systems are important units in buildings to regulate the indoor temperature, humidity, and air quality so that occupants feel comfortable. They are also major energy-consuming appliances in building systems, of which the energy consumption makes up almost 40% of the total energy consumption in the United States [1]. On the one hand, with the aim of reducing greenhouse gas emissions, it is of great significance to reduce the energy consumption of HVAC systems. On the other hand, the comfort levels of occupants should also be maintained while using energy-saving HVAC controllers.

The model predictive control (MPC) is widely used for HVAC systems. With MPC controllers, future time slots can be considered when computing the optimal control inputs, and the constraint requirements are easy to formulate [2], [3]. However, MPC methods require a sufficiently accurate and computationally tractable system model to find an optimal solution [4]. In addition, the optimization problem of MPC is not guaranteed to be convex such that more sophisticated optimization approaches are needed. Zhao et al. [5] designed an EnergyPlus model-based predictive control method. They used the exhaustive search algorithm to find the optimal solution with the assistance of EnergyPlus simulation. However, this brute-force optimization strategy is only computationally suitable for simple building models with discrete state and action variables. The authors in [6], [7] investigated the multi-objective optimization problems for building systems in which energy performance and occupant's thermal comfort are considered. The genetic algorithm was used to generate the control sequences based on the EnergyPlus simulation. However, two problems are associated with the algorithms in [5], [6], [7]. First, it is hard to obtain an EnergyPlus model for a certain building since the building's parameters need to be measured, such as the thicknesses of the roof and vertical walls, the absorption coefficient of solar radiation of the roof, etc. Even if we make the measurements, the simulation-based EnergyPlus models may be still much simpler than the real buildings [8], making the derived controllers suboptimal. Second, the brute-force method in [5] and genetic algorithms in [6] and [7] are computationally inefficient. The key issue is that there is no systematic and computationally efficient algorithm to solve the optimization problems associated with these black-box EnergyPlus models.

Reinforcement learning (RL) can be used to derive an optimal controller in an unknown environment by interacting with it [9], [10], [11]. Combined with the high function approximation

Manuscript received 15 May 2022; revised 8 November 2022; accepted 27 February 2023. Date of publication 1 March 2023; date of current version 8 September 2023. This work was supported in part by the National Science Foundation under Cyber-Physical Systems under Grant 1837021. Recommended for acceptance by B. Ravindran. *(Corresponding author: Ying Zhang.)*

Liangliang Chen and Ying Zhang are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: liangliang.chen@gatech.edu; yzhang@gatech.edu).

Fei Meng is with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong (e-mail: feimeng@link.cuhk.edu.hk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSUSC.2023.3251302, provided by the authors.

Digital Object Identifier 10.1109/TSUSC.2023.3251302

capacity of deep neural networks, deep RL techniques have achieved remarkable performances in Atari 2600 games [12], chess, shogi, Go [13], [14], and simulated robots [15], [16], [17]. Some researchers have applied deep RL techniques in the HVAC control field. Ref. [18] used the deep Q-network for building HVAC control. The simulation results on EnergyPlus models showed that the deep RL algorithm consumed less energy than a rule-based approach. The authors in [19] considered the non-stationary building environments and developed a deep Q-network-based RL method that can actively detect the change points of environments. The algorithms in [18], [19] are model-free, which directly learn the optimal value functions or policy from the transition dataset collected in the real environment. Model-free RL algorithms suffer from low sample efficiency. The authors in [20] compared some well-known model-free RL algorithms in the field of HVAC controller design. The simulation results indicated that SAC [17] is more data-efficient than TD3 [21], PPO [22], and TRPO [15]. However, it still requires ten months of online training data with SAC to obtain a converged indoor temperature if the data are collected every fifteen minutes. In contrast, the model-based RL algorithm learns a dynamics model of the environment during interactions and utilizes the learned model to derive a control policy [23], [24]. When training the policy, we can generate simulated data via the learned model. Thus, fewer environment samples are required compared with the model-free algorithms. Ref. [25] proposed a model-based RL algorithm for HVAC systems with the zone thermal dynamics modeled by a recurrent neural network. MPC with random-sampling shooting and imitation learning were utilized to determine the best actions under certain states. The authors in [26] derived a model-based RL strategy for HVAC systems by directly obtaining a neural network policy from the MPC sampled trajectories without imitating the results of random shooting. Note that both model-free and model-based algorithms in [18], [19], [20], [25], [26] are trained in online manners; namely, the RL agent has to interact with the environment to collect more data. However, these online learning strategies would cause two issues when they are used to obtain HVAC optimal controllers. First, the sample-efficient model-based RL algorithms still require online data collection. During the initial stage of collection, the non-optimal operation statuses of HVAC systems would degrade occupants' thermal comfort levels and waste energy. Second, insufficient explorations to the environment may result in a poor-quality RL algorithm. What is worse, if we use the non-optimal intermediate controller to explore the environment, the operations of the HVAC systems might be unsafe [27], [28].

In order to solve the above problem, offline RL methods were proposed by developing algorithms from an offline dataset without further interactions with the environment [29]. That is, there is no exploration to the environment during the policy learning, and we can only focus on exploiting the existing dataset [30]. Since this dataset may not cover all the state and action spaces, the learning process should be limited to the "known" regions where there are a sufficient number of data. Refs. [31] and [32] investigated model-based offline RL methods in which ensemble neural networks are used to identify the "known" and "unknown" spaces. The results showed that the model-based offline RL algorithms outperform the model-free counterparts in most simulated robot control tasks the authors tested on. In the field of HVAC control, using a finite number of historical data, Jiang et al. [33] designed a deep-Q-network-based RL algorithm for HVAC systems to reduce energy cost. In order to reduce the number of required samples, the authors used an action processor that leverages the information of daily electricity price. The authors in [34] proposed an offline RL algorithm for HVAC systems in multiple zones, which were modeled as a Markov game. Neural network black box models were learned to generate synthetic data for the subsequent policy training. Ref. [35] also trained an LSTM environment model from the historical HVAC operational dataset. An RL agent was trained via DDPG [16] by interacting with the obtained model. However, the learning procedures in [33], [34], [35] did not include strategies to avoid the "unknown" regions. In this case, the optimal policy may not be found [36], as we show in Section VI-C2 for the EnergyPlus environment we used. The authors in [27] proposed a model-free offline RL algorithm for HVAC systems by using the conservative Q-learning (CQL) [37]. Safety exploration is also considered in [27] when collecting the offline dataset. The model-based performance evaluation was then utilized to select the best policy after training. The authors in [28] applied the batch constrained Munchausen RL algorithm [38], [39] for safe HVAC control. Regression models based on the predicted mean vote (PMV) were used to predict occupants' thermal comfort levels. Note that the algorithms in [27], [28] are both model-free, which may be overly conservative since the algorithms learn only on the states in the offline dataset [40], [41]. In contrast, for model-based offline RL, the dynamics model of the environment is first learned to generate synthetic data. The policy is then trained by interacting with the learned model that has some generalization capabilities to unseen states in the offline dataset. In addition, various planning strategies can be used to derive a policy from the model [42], such as MPC [43] and policy optimization [44]. These planning algorithms give us more flexibility to achieve high performance in different scenarios. In this paper, we will investigate the design of model-based offline RL algorithms for HVAC systems.

Different occupants may have different thermal preferences [45], [46], [47]. The HVAC controllers in [18], [20], [25], [26], [27], [28], [33], [34], [35] did not consider the occupants' personalized thermal preferences. The thermal comfort metric in these literature, if considered, is the same for all possible occupants. For example, Ref. [19] utilized the predicted percentage of dissatisfied (PPD) as the thermal comfort metric. The value of the PPD is related to the PMV, which further depends on many factors such as metabolic rate, insulation, air temperature, etc. [48]. However, some variables, such as metabolic rate and clothing insulation, are hard to measure in practice. Moreover, even if we can assume that the factors affecting the PPD are available, different thermal preferences still impact the real comfortable levels of occupants [45], [46], [47]. In order to solve the issues above, we can provide occupants a feedback channel to message their thermal feelings, which can be used to improve the occupants' thermal comfort levels [49].

Some researchers have investigated preference learning and its application in human-in-the-loop control. For example, Ref. [50] proposed an active preference learning algorithm when a human decision maker can only express a preference by comparing the results of two candidate decisions. The algorithm actively provides suggestions for the next candidate decision considering the trade-off between exploration and exploitation of the decision space. The authors in [51] applied the algorithm in [50] to a path-based velocity planner in robotic sealing tasks. Based on the pairwise preferences provided by a user, the algorithm suggests the next set of parameters for the task execution so that the velocity planner can determine an appropriate value of reference velocity for the robotic sealing task. The authors in [52] also utilized the preference learning algorithm to model the knowledge of the human operator. One merit of the preference learning algorithm in [50], [51], [52] is its high computational efficiency since the proposed method only needs to solve a linear or quadratic programming problem. In addition, it is sample efficient due to the active learning. However, the preference learning algorithm in [50], [51], [52] was designed to handle pairwise preferences, which are essentially qualitative. When we try to learn the thermal comfort model of an occupant, we usually provide more selections of feedbacks. For example, a quantitative 7-point thermal sensation vote is leveraged in the ASHRAE datasets [53], [54], [55] to indicate different levels of feeling hot or cold. The authors in [56] proposed a meta-learning algorithm that can adapt a meta-trained deep neural network to a new task with a few training data. This paper will apply the meta-learning method to train the occupant's thermal preference model so that only a few feedback data are needed for a new occupant to identify his/her thermal preference. The requirement of only a few thermal feedbacks contributes to fewer interventions in occupants' lives. Compared with the pairwise preference learning algorithm in [50], [51], [52], meta-learning can leverage the information of thermal preferences of different occupants to improve the thermal comfort model of a specific occupant. A detailed description of the meta-learning algorithm we used is provided in Section II-A.

In this paper, we develop an efficient human-in-the-loop control strategy for HVAC systems by using meta-learning to learn the occupants' thermal preferences and offline model-based RL to regulate the controller. Both meta-learning and offline RL learning contribute to the fast learning property of the proposed algorithm. With meta-learning, we can derive the personalized thermal comfort model with only a few thermal feedbacks. With offline RL, we can leverage the historically collected dataset to develop an HVAC controller. In contrast, an online RL algorithm may require a long-time data collection process before obtaining a converged controller. The contributions of this paper are listed as follows.

i) We design an offline model-based RL algorithm for HVAC systems by modeling the thermal dynamics of a zone as a partially observable Markov decision process (POMDP). The proposed algorithm can be trained with historically collected data via a suboptimal controller such as a PID controller. This indicates that the designed controller can be obtained without further interactions with the building environment, which helps save energy and shorten the time to deploy for HVAC systems in practice.
ii) We model the thermal preference learning of different occupants as different tasks under the meta-learning framework. This allows us to learn a personalized thermal preference model with a few thermal feedbacks from a specific occupant. In addition, by combining meta-learning with model-based offline RL, the designed HVAC controller can be quickly regulated to accommodate personalized thermal preferences with only a few thermal feedbacks.
iii) We test the meta-supervised learning algorithm on the ASHRAE database II [54], [55]. The learned model is a mapping from indoor air temperature to the 7-point thermal sensation vote of an occupant. Compared with the best result in [28], the meta-supervised learning algorithm reduces the root mean square error (RMSE) $\sim 6.63\%$.
iv) The effectiveness of the RL algorithm is verified in an EnergyPlus simulation environment. The results show that the proposed algorithm can generally guarantee personalized thermal preferences, with only additional 1.91% power consumption on average compared with the model-based RL algorithm with on-policy data aggregation. In addition, the comparisons with the model-free CQL algorithm also demonstrate the advantage of our algorithm.

To the best of our knowledge, we are the first to use meta-learning to learn the personalized thermal comfort model. Moreover, we are also the first to combine meta-learning and offline model-based RL to derive a personalized HVAC control system. The experiment results on the ASHRAE database II and in an EnergyPlus environment demonstrate the superiority of our algorithm.

The remainder of this paper is organized as follows. Section II introduces some preliminaries in this paper. Section III presents the meta-learning method to learn the occupant's thermal preference model. Section IV gives the dynamics model learning based on the offline dataset. The model-based RL algorithm considering occupant's thermal preference is presented in Section V. Section VI demonstrates the effectiveness of the proposed algorithm via the simulation in an EnergyPlus environment, and Section VII concludes this paper. The appendices, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSUSC.2023.3251302, introduce the notations, the summarized algorithms and the overall algorithm architecture in this paper, the neural network architectures and hyperparameter settings of the algorithms, and the statistics of the in-distribution and out-of-distribution datasets. They are helpful in understanding this paper.

II. Preliminaries

This section will present some preliminaries in this paper, including meta-learning, model-based offline RL, and system model and control. A brief description of the proposed approach will also be presented.

A. Meta-Supervised Learning

Meta-learning aims to learn a learning procedure that can adapt to new tasks quickly with only a few data about the new tasks. The new tasks, i.e., the meta-testing tasks, should be drawn from the same task distribution $\rho(\mathcal{T})$ as the meta-training tasks. In this paper, we consider the model-agnostic meta-learning (MAML) algorithm [56]. In MAML, the goal of the meta-training is to find a good weight initialization of a neural network, with which a few gradient-based updates can make a significant adaptation.

For the gradient-based meta-learning, we can formulate it as

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathcal{T}\sim\rho(\mathcal{T})}\left[\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{\text{test}}, \boldsymbol{\theta}')\right],$$
$$\text{subject to} \quad \boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{\text{train}}, \boldsymbol{\theta}), \tag{1}$$

where α is the adaptation step size, $\mathcal{D}_{\mathcal{T}}^{\text{train}}$ and $\mathcal{D}_{\mathcal{T}}^{\text{test}}$ are the training and test data sets, respectively, in the meta-training phase. $\mathcal{L}(\mathcal{D}_{\mathcal{T}}, \boldsymbol{\theta})$ presents the loss for task $\mathcal{T}$ whose expression depends on the objective of meta-learning. In practice, the expectation in (1) can be approximated by the corresponding empirical value.

Remark 1. The minimizer $\boldsymbol{\theta}^*$ to the problem (1) encodes the internal information that is transferable from one task to another [56]. In addition, when we adapt the parameter $\boldsymbol{\theta}^*$ to a specific task using a few gradient steps, the model prediction should have sufficient changes so that the performance of this model on the new data is improved significantly. Considering these aspects, we use a fully-connected neural network as the model for occupants' thermal comfort due to its high nonlinearity and function approximation capacity. Note that the Gaussian process can also be utilized to encode the prior information, which can be achieved by assigning or learning an appropriate kernel function [57], [58]. However, a Gaussian process model is essentially a basic supervised learning model without being optimized to learn new tasks fast using previous learning experiences of other tasks. Thus, this paper uses the meta-supervised learning method, i.e., MAML, to learn personalized thermal preference models.

B. ASHRAE Global Thermal Comfort Database II

The ASHRAE global thermal comfort database II [54], [55] records 110,070 thermal votes from different occupants under various environmental and personal conditions[1]. These conditions include the air temperatures at varying distances (0.1 m, 0.6 m, 1.1 m) above the floor, relative humidity, the subject's height and weight, etc. The support of the thermal sensation vote is $\{-3, -2, -1, 0, 1, 2, 3\}$, which represent {cold, cool, slightly cool, neural, slightly warm, warm, hot}, respectively. Recently, the subjects' identities were added to some raw data of the database. This information enables us to test the performance of meta-learning on predicting the thermal sensation votes since we can model the thermal preference learning of different occupants as different tasks that implicitly share some commonalities.

1.The newly released dataset can be downloaded at https://datadryad.org/stash/dataset/doi:10.6078/D1F671.

In Section VI-A1, we will use this dataset to show the advantage of meta-supervised learning in reducing the model prediction error compared with the traditional simple supervised learning. Herein, we use the brief name "ASHRAE database II" for this database.

C. Model-Based Offline Reinforcement Learning

The RL framework is built on a Markov decision process (MDP) which can be described by a tuple $\{\mathcal{S}, \mathcal{A}, p_0, p, r, \gamma\}$. In this tuple, $\mathcal{S}$ and $\mathcal{A}$ denote the spaces of state and action, respectively, $p_0(\boldsymbol{s}) : \mathcal{S} \mapsto [0, 1]$ denotes the initial state distribution of the MDP, $p(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ represents the state transition probability, $r(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ denotes the reward function, and $\gamma \in [0, 1]$ is the reward discount factor that determines the present value of future rewards [9], respectively.

A deterministic policy in RL is function $\pi_{\boldsymbol{\theta}^\pi}(\boldsymbol{s}) : \mathcal{S} \mapsto \mathcal{A}$, parameterized by $\boldsymbol{\theta}^\pi$, that maps a state to an action. The goal of RL in this paper is to maximize the expected cumulative reward in a future horizon T by finding an optimal policy parameter $\boldsymbol{\theta}^{\pi*}$. The corresponding optimal policy is

$$\pi^*_{\boldsymbol{\theta}^\pi} = \arg\max_{\pi_{\boldsymbol{\theta}^\pi}} \sum_{t=0}^{T-1} \mathbb{E}_{\boldsymbol{s}_0\sim p_0(\cdot), \boldsymbol{s}_{t+1}\sim p(\cdot|\boldsymbol{s}_t, \pi_{\boldsymbol{\theta}^\pi}(\boldsymbol{s}_t))} \left[r(\boldsymbol{s}_t, \pi_{\boldsymbol{\theta}^\pi}(\boldsymbol{s}_t), \boldsymbol{s}_{t+1})\right], \tag{2}$$

where we assume $\gamma = 1$, since we are considering a finite horizon task.

Remark 2. The value of the discount factor γ would influence the algorithm performance and even stability. In the model-free RL case, γ determines the time scale of the cumulative reward we would like to maximize [9]. A small γ value will lead to a myopic objective, while a γ value very close to 1 would cause instability [59]. In the case of model-based RL, the stability issue with a large γ would be more obvious since there exist compounding model errors when a long rollout is sampled from the model [44], [60]. In order to avoid this issue in our offline RL algorithm, we intentionally maximize the cumulative reward over a finite time horizon T, instead of solving an empirical Bellman equation [37]. Although this would introduce bias to the optimization objective, the variance of cumulative reward would decrease, contributing to the stability of the algorithm.

When the action space $\mathcal{A}$ is continuous, policy gradient methods are commonly used to solve the optimization problem (2) [9]. In this paper, we choose the actor-critic framework to derive the optimal policy. In this framework, the critic's role is to estimate the value function of an intermediate/converged policy. Due to the continuous action space, an actor is trained along with the critic to find a policy maximizing the value function. During the training process, the critic and actor converge to the optimal value function and optimal policy, respectively. In addition, we use model-based RL strategies [43], [44], [61] to obtain the optimal policy. That is, we train both a dynamics model and a policy. The learned dynamics model can generate synthetic data for policy training.

Interactions with the environment to collect more samples are required when training an online model-based RL algorithm.

For instance, during the initial learning period, it is necessary to explore the state and action spaces sufficiently. However, for HVAC systems, the actions that are wrongly explored may lead to an unsafe state, e.g., an unreasonably high indoor air temperature, which makes the occupants extremely uncomfortable. Offline model-based RL aims to learn a good policy with only offline data. Since no further interaction with the environment is allowed, we need to avoid the state entering the region that is not covered or sufficiently covered by the offline dataset. To this end, we not only need to use the model to make predictions but also should estimate the epistemic uncertainties of the predictions. Different strategies for estimating a neural network model's epistemic uncertainty can be used, such as ensemble networks [62], [63], Bayesian deep learning [64], [65], and probabilistic neural networks [60], etc. In this work, we will use ensemble neural networks to estimate epistemic uncertainty.

Remark 3. For the model-based RL in our paper, the model to be learned is the joint model of thermal dynamics of the considered zones and the HVAC systems. After this model is identified, traditional model-based optimal control techniques are also candidates to derive the HVAC controllers. However, these model-based optimal control methods typically require a simple identified model, e.g., a linear model, or have high computational complexities, while the real models are typically complicated and highly nonlinear [18], [25], [66]. For example, the authors in [67] used the linear quadratic regulator (LQR) [68] to design a hierarchical optimal controller with the optimization objective of occupants' comfort and energy consumption. In order to use the LQR controller, the authors linearized the identified model around the equilibrium points of the system. As Ref. [67] pointed out, this linearization may introduce significant errors when the range of thermal zone temperature is wide. Based on the linearization method, the LQR controller in [69] was only used to generate the event-triggered control actions in the initial period. The generated control inputs were further utilized for training a more advanced adaptive critic-based neural network controller. We can also use more complicated physical-based models, e.g., EnergyPlus, or a black-box model, e.g., a neural network, to describe the thermal dynamics of the considered zones. However, the optimization algorithms would be computationally inefficient. For example, the authors in [5] used the exhaustive search method, and the authors in [6], [7] used a genetic algorithm. In contrast, by leveraging deep RL, the high nonlinearity of the model would not bring about too much complexity for controller development, since RL algorithms learn a controller by trial and error and do not exploit the model structures. In addition, the neural network-based policy is computationally efficient to obtain the optimal control input after training. These are the reasons why we do not use traditional model-based optimal control methods and choose RL in this paper.

D. System Model and Control

In the RL learning framework, the general thermal dynamics model of a zone can be described as

$$\boldsymbol{s}_{t+1} = f(\boldsymbol{s}_t, \boldsymbol{a}_t) + \boldsymbol{\epsilon}, \tag{3}$$

$$r_{t+1} = g(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}), \tag{4}$$

where $f(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ and $g(\cdot,\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ are unknown function, and $\boldsymbol{\epsilon}$ is an unknown additive noise term. Note that the functions f and g and the noise $\boldsymbol{\epsilon}$ are all related to the configuration parameters of the considered zone. In Section VI, we use a two-zone data center in the EnergyPlus environment to show the algorithm performance in this paper. However, the proposed algorithm can be used for other zone configurations.

Remark 4. The reward function is typically related to the power consumption and occupants' thermal comfort levels in the zone. The power consumption is assumed to be known in the algorithm in this paper. However, since different occupants may have different thermal preferences, the reward term related to the thermal comfort level is unknown, although we can assume that the indoor air temperature is easy to measure and thus known. The relationship between thermal comfort level and system state, e.g., indoor air temperature, is learned via meta-learning in this paper. This is where the human-in-the-loop strategy is used.

If we would like to model the zone thermal dynamics as an MDP, the state variables may include indoor air temperature and humidity of the considered zone, outdoor air temperature, wall temperatures of the zone, and working statuses of heating sources in the zone, to name a few. Due to the high complexity and unpredictable disturbances of the zone thermal dynamics, it is impractical to consider every possible state variable. In this case, we can model the zone thermal dynamics as a POMDP and only take some easily measured variables as observation variables. In order to compensate for the lacking of unmeasured states, we can use segments of historical observation and action trajectories to replace the state when training the dynamics model and policy networks [70]. In the remainder of this paper, the symbol $\boldsymbol{s}$ is used to represent observation variables when there is no confusion arising. In this setting, the input of the thermal dynamics model is $(\{\boldsymbol{s}_t\}_{t=t'-H}^{t'-1}, \{\boldsymbol{a}_t\}_{t=t'-H}^{t'-1})$ and the policy is a function of the tuple $(\{\boldsymbol{s}_t\}_{t=t'-H}^{t'-1}, \{\boldsymbol{a}_t\}_{t=t'-H}^{t'-1}, \boldsymbol{s}_{t'})$, where H is the length of the historical segment.

E. Proposed Approach

This paper will design an offline model-based RL algorithm for HVAC systems that can adapt to different occupants' thermal preferences with a few thermal feedbacks. We build the HVAC control algorithm based on the following offline datasets.

- A historical building operation dataset of, for example, indoor and outdoor air temperatures and operation statuses of HVAC systems. This dataset can be collected in advance with a traditional controller such as a PID controller.
- An offline dataset of thermal feedbacks from different occupants with various thermal preferences.

With the above offline datasets, this paper first considers learning an occupants' thermal comfort model via meta-supervised learning. This model should be able to quickly adapt to different occupants' thermal preferences with only a few thermal feedbacks from the corresponding occupants. Second, a thermal dynamics model is learned from the offline historical

building operation dataset. This model should not only have a good prediction accuracy in the known region but also provide indications when predictions are made in the unknown region. Finally, a model-based RL algorithm is derived based on the learned zone's thermal dynamics model and the thermal comfort model. The designed model-based RL algorithm can regulate the HVAC systems to meet occupants' personalized thermal comfort requirements with low power consumption.

III. Meta-Supervised Learning for Occupant's Thermal Comfort Model

In order to guarantee the occupant's personalized thermal comfort, an HVAC system needs to adapt to the new thermal preference quickly. In addition, we would like this aim to be achieved with occupant feedbacks as few as possible since more feedbacks not only make the adaptation slower but also bring inconvenience to occupants. In this section, motivated by the MAML algorithm [56], we will develop an algorithm that quickly learns the occupant's thermal feedback model.

Many factors affect the thermal comfort level of an occupant, such as air temperature, relative humidity, metabolic rate, clothing insulation, etc. [71], [72]. In this section and the simulation in Section VI, we consider indoor air temperature as the only factor affecting thermal preference level for simplicity. We assume that each occupant corresponds to a specific indoor temperature T^{ref} that he/she feels the most comfortable. In order to adapt the controller to different thermal preferences, the predicted occupant's comfort level will serve as a reward term when designing the RL algorithm. We denote this reward term at time step t' as $\widehat{r}^{\text{o}}_{t'}$, which is related to thermal feedbacks. To collect thermal feedbacks from occupants, we can provide them with a feedback device with five selections "very uncomfortable," "uncomfortable," "moderate," "comfortable," and "very comfortable," for example. These five feedback selections can be quantified as -2, -1, 0, $+1$, and $+2$, respectively,[2] which can be considered as the true reward $r^{\text{o}}_{t'}$ at different values of indoor temperatures, i.e.,

$$r^{\text{o}}_{t'} = f(T^{\text{in}}_{t'}), \tag{5}$$

with $f(\cdot)$ being an unknown function which we can use a neural network $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(\cdot)$ parameterized by $\boldsymbol{\theta}_{\mathcal{R}}$ to approximate.

Note that the domain of function f is the discrete set $\{-2,-1,0,1,2\}$ while $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}$ is learned to be a continuous function with respect to $T^{\text{in}}_{t'}$. The reasons why we allow $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}$ to be continuous are stated as follows. An alternative to this "regression" method is a "classification" approach, where different indoor temperatures are assigned to a thermal feedback in $\{-2,-1,0,1,2\}$. However, although we consider that the occupant's thermal feedback is a function of indoor temperature $T^{\text{in}}_{t'}$, the comfort level of an occupant should not be formulated literally in a "classification" way. First, indoor temperature is not the only factor affecting an occupant's thermal sensation. Second, thermal feedback is subjective where there exists some randomness. Fig. 1 illustrates the relationship between $f(T^{\text{in}})$, indicated as cross mark "×," and $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(T^{\text{in}})$, sketched as the curve. The reference temperature T^{ref} is assumed to be 23°C. We might get multiple possible thermal votes at a certain indoor temperature T^{in}. For example, when $T^{\text{in}} = 21°\text{C}$, we might have $f = -2$ with probability 0.75 and $f = -1$ with probability 0.25. Similarly, when $T^{\text{in}} = 22°\text{C}$, we might have $f = -1$, $f = 0$, and $f = +1$ with probabilities 0.25, 0.5, and 0.25, respectively. For the learned function $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}$, we would like to have $-2 < \widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(T^{\text{in}}) < -1$ for $T^{\text{in}} = 21°\text{C}$. In addition, when $T^{\text{in}} = 21.5°\text{C}$, it would be more reasonable to have $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(T^{\text{in}}) \approx -0.5$, instead of $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(T^{\text{in}}) = 0$ or $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(T^{\text{in}}) = -1$. Another benefit we can gain by using a regression model is that the reward space of the subsequent RL would be continuous. Since our RL agent is updated with policy gradient, a continuous reward space would be helpful for policy convergence. Note that both regression and classification models have been used to predict thermal sensation votes. Please refer to Ref. [73] for a detailed literature review and comparison. In Section VI-A, we will show that this "regression" strategy works effectively with meta-supervised learning, although the data labels are discrete.

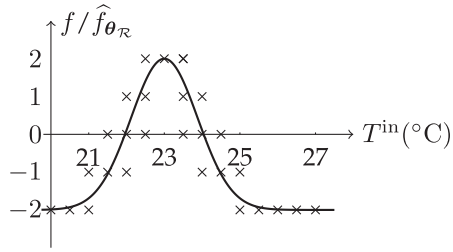

Fig. 1. The relationship between thermal feedbacks $f(T^{\text{in}})$ and the approximation function $\widehat{f}_{\boldsymbol{\theta}_{\mathcal{R}}}(T^{\text{in}})$.

The detailed meta-supervised learning algorithm is shown in Algorithm 1 in Appendix B.1, available in the online supplemental material.

IV. Dynamics Model Learning for the Offline Reinforcement Learning

In the settings of this paper, we consider that only an offline dataset is available for deriving the model-based RL algorithm. This section will show what dynamics model should be learned from the offline dataset to develop an RL algorithm.

Suppose that we have collected an offline dataset[3] $\mathcal{D}$ of size $N_{\mathcal{D}}$, with each datum in the form of tuple $(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}, \boldsymbol{s}_{t'+1})$. On one hand, we would like to learn a model that performs well for the in-distribution data. On the other hand, the model should indicate its inefficiency when required to make predictions

2.Since an RL algorithm is trained to maximize the cumulative reward at a certain future time horizon, it can only achieve the best cumulative reward after sufficient training. Thus, although we can provide more detailed feedback options, for example, cooler or hotter instead of just uncomfortable, the RL algorithm cannot leverage this information to update its parameters.

3.This offline dataset can be collected via implementing a low-level controller. In the numerical simulation in this paper, we use a PID controller to get the simulated offline data. See the details in Section VI-B.

on the out-of-distribution data. Motivated by [31], [32], [60], we learn an ensemble of N_{m} deterministic networks $\{\widehat{T}^n_{\boldsymbol{\theta}_n}(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1})\}^{N_{\mathrm{m}}}_{n=1}$, with $\widehat{T}^n_{\boldsymbol{\theta}_n} : (\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}) \mapsto \boldsymbol{s}_{t'+1}, \quad n \in \{1, \ldots, N_{\mathrm{m}}\}$, being the n-th network with parameter vector $\boldsymbol{\theta}_n$ trained independently of other networks. In practice, we can use recurrent neural networks and seek $\boldsymbol{\theta}_n$ to minimize the mean squared error

$$\mathcal{E}_n(\boldsymbol{\theta}_n) \triangleq \frac{1}{N_{\mathcal{D}}} \sum_{D_{t'} \in \mathcal{D}} \Big\| \boldsymbol{s}_{t'+1} - \widehat{T}^n_{\boldsymbol{\theta}_n}\left(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}\right)\Big\|^2_2, \quad (6)$$

with $D_{t'} \triangleq (\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}, \boldsymbol{s}_{t'+1})$, for each $n \in \{1, \ldots, N_{\mathrm{m}}\}$, independently.

The ensemble networks after training can be used to make predictions via

$$\widehat{s}_{t'+1} = \frac{1}{N} \sum_{n=1}^{N} \widehat{T}^n_{\boldsymbol{\theta}_n}\left(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}\right). \quad (7)$$

That is, the final prediction is the average of the predictions of all individual networks $\widehat{T}^n_{\boldsymbol{\theta}_n}$, $n = 1, \ldots, N$. Note that using the average outputs helps to improve the prediction accuracy for in-distribution states and actions [60].

Note that the trained ensemble networks can not only achieve performance improvement by averaging over all individual networks' outputs but also capture the epistemic uncertainty via the disagreement of different networks' outputs [60]. The epistemic uncertainty can be leveraged to test if inferences are made for out-of-distribution inputs. For the in-distribution inputs, all the networks are well-trained, and the prediction differences between them are small. However, since different initializations of neural networks and training processes, the predictions of different networks for the out-of-distribution inputs may have large disagreements. We would like to avoid making out-of-distribution predictions with respect to the offline dataset since these predictions may have large errors that hurt the subsequent planning. To this end, we measure the epistemic uncertainty as

$$r^{\mathrm{u}}_{t'+1} \triangleq -\frac{1}{\binom{N}{2}} \sum_{n_1, n_2 \in \{1, \ldots, N\}, n_1 \neq n_2} \Big\| \widehat{T}^{n_1}_{\boldsymbol{\theta}_{n_1}}\left(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}\right) - \widehat{T}^{n_2}_{\boldsymbol{\theta}_{n_2}}\left(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1}\right)\Big\|^2_2, \quad (8)$$

which will serve as a reward term in the planning procedure to punish the actions leading to the out-of-distribution data region.

Remark 5. The reward term (8) only concerns the epistemic uncertainty, i.e., the subjective uncertainty due to a lack of data. Note that the aleatoric uncertainty is intrinsic to the data and should not serve as a punishment for the RL learning. We are not sure whether the regions with high data variances, i.e., the regions with high aleatoric uncertainties, would produce high expected cumulative rewards or not. If these regions are sufficiently explored in the offline dataset, we can exploit the model on these regions when training the RL agent. However, modeling the aleatoric uncertainty may help improve the performance of the RL agent, especially in the case when the transition dynamics of the environment is multi-modal [24], [74]. The aleatoric uncertainty can be estimated via Bayesian deep learning [65], probabilistic neural networks [60], and quantile regression [75], etc. In addition, a well-calibrated model may also help when considering both epistemic and aleatoric uncertainties [76], [77]. In our paper, the learned model is not calibrated since we are only concerned with whether a region is fully covered by the offline dataset and thus we can determine whether exploit the learned model in this region or not. The effectiveness of this strategy was demonstrated in [31], [32]. Section VI will show that an ensemble of 3 deterministic networks is sufficient for learning an offline RL algorithm for the considered two-zone data center EnergyPlus simulation environment.

V. Model-Based Reinforcement Learning With Personalized Occupants' Thermal Preferences

In this section, we will design a controller for the HVAC system considering occupants' thermal feedbacks. The controller is developed via planning with the learned thermal dynamics model in Section IV. In addition, by adding r^{o}_t in (5) as a reward term, the controller can adapt to new thermal preference feedbacks quickly with only a few feedbacks required.

The reward function in RL indicates which actions are good under a certain state and regulates the controller to achieve higher cumulative rewards in a certain future horizon. In this paper, we design the reward function at time step t' as

$$\begin{aligned} &r_{t'}\left(\{\widehat{\boldsymbol{s}}_t\}^{t'-1}_{t=t'-H}, \{\boldsymbol{a}_t\}^{t'-1}_{t=t'-H}, \widehat{\boldsymbol{s}}_{t'}\right) \\ &\triangleq \lambda_1 r^{\mathrm{t}}_{t'}\left(\widehat{\boldsymbol{s}}_{t'}\right) + \lambda_2 r^{\mathrm{e}}_{t'}\left(\boldsymbol{a}_{t'-1}, \widehat{\boldsymbol{s}}_{t'}\right) + \lambda_3 \widehat{r}^{\mathrm{o}}_{t'}\left(\widehat{\boldsymbol{s}}_{t'}\right) \\ &\quad + \lambda_4 r^{\mathrm{u}}_{t'}\left(\{\widehat{\boldsymbol{s}}_t\}^{t'-1}_{t=t'-H}, \{\boldsymbol{a}_t\}^{t'-1}_{t=t'-H}\right), \end{aligned} \quad (9)$$

where $\lambda_1, \ldots, \lambda_4$ denote the weight parameters, $r^{\mathrm{t}}_{t'}(\widehat{\boldsymbol{s}}_{t'})$ represents the reward term which is negative if the state $\widehat{\boldsymbol{s}}_{t'}$ would typically make occupants uncomfortable, $r^{\mathrm{e}}_{t'}(\boldsymbol{a}_{t'-1}, \widehat{\boldsymbol{s}}_{t'})$ denotes the power consumption if action $\boldsymbol{a}_{t'-1}$ is taken at time step $t'-1$ and the new obtained state is $\widehat{\boldsymbol{s}}_{t'}$, $\widehat{r}^{\mathrm{o}}_{t'}(\widehat{\boldsymbol{s}}_{t'})$ denotes the reward term related to the occupant's thermal preference, and $r^{\mathrm{u}}_{t'}(\{\widehat{\boldsymbol{s}}_t\}^{t'-1}_{t=t'-H}, \{\boldsymbol{a}_t\}^{t'-1}_{t=t'-H})$, defined as (8), is used to constrain the states during planning into the region where the epistemic uncertainty is low. Section VI-C will show how these reward terms are designed in a specific application scenario. In this section, we assume that the reward function $r_{t'}(\{\widehat{\boldsymbol{s}}_t\}^{t'-1}_{t=t'-H}, \{\boldsymbol{a}_t\}^{t'-1}_{t=t'-H}, \widehat{\boldsymbol{s}}_{t'})$ is known.

We use the method proposed in [26] to derive a model-based offline RL algorithm. The objective to maximize is the expected cumulative reward in an MPC prediction horizon T. In this paper, we design a deterministic policy $\pi_{\boldsymbol{\theta}^\pi} : \mathcal{S}^H \times \mathcal{A}^H \mapsto \mathcal{A}$, which can be a neural network with parameter $\boldsymbol{\theta}^\pi$. At time step k, we seek an optimal value of $\boldsymbol{\theta}^\pi$ to maximize

$$J_k \triangleq \mathbb{E}\left[R_k \mid \{\widehat{\boldsymbol{s}}_t\}^k_{t=k-H}, \pi_{\boldsymbol{\theta}^\pi}\right], \quad (10)$$

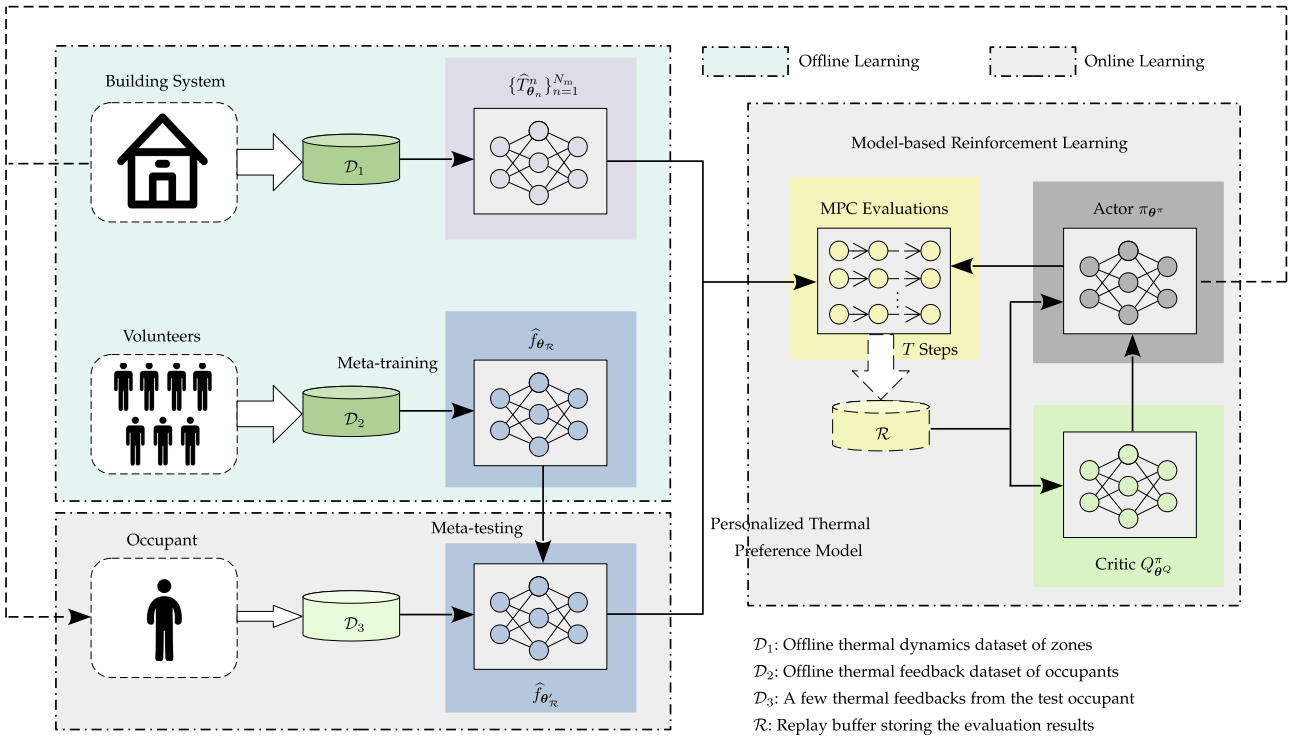

Fig. 2. The architecture of the algorithm.

with

$$R_k \triangleq \sum_{t'=k}^{k+T-1} r_{t'}\left(\{\widehat{\boldsymbol{s}}_t\}_{t=t'-H}^{t'-1}, \{\boldsymbol{a}_t\}_{t=t'-H}^{t'-1}, \widehat{\boldsymbol{s}}_{t'}\right). \tag{11}$$

We use the actor-critic architecture to solve the corresponding optimization problem, in which the actor network is $\pi_{\boldsymbol{\theta}^\pi}: \mathcal{S}^H \times \mathcal{A}^H \times \mathcal{S} \mapsto \mathcal{A}$ and the critic network, parameterized by $\boldsymbol{\theta}^Q$, is $Q^\pi_{\boldsymbol{\theta}^Q}: \mathcal{S}^H \times \mathcal{A}^H \times \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, respectively.

The critic network is trained in practice via minimizing the empirical loss [26]

$$\widehat{\mathcal{L}}(\boldsymbol{\theta}^Q) \triangleq \frac{1}{N'}\sum_{n=1}^{N'}\Bigg(Q^\pi_{\boldsymbol{\theta}^Q}\left(\{\widehat{\boldsymbol{s}}_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}_{n,t}\}_{t=-H}^{-1}, \widehat{\boldsymbol{s}}_{n,0}, \boldsymbol{a}_{n,0}\right) - \sum_{t'=0}^{T-1} r_{t'}\left(\{\widehat{\boldsymbol{s}}'_{n,t}\}_{t=t'-H}^{t'-1}, \{\boldsymbol{a}'_{n,t}\}_{t=t'-H}^{t'-1}, \widehat{\boldsymbol{s}}'_{n,t'}\right)\Bigg)^2, \tag{12}$$

where the tuple $(\{\widehat{\boldsymbol{s}}_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}_{n,t}\}_{t=-H}^{-1}, \widehat{\boldsymbol{s}}_{n,0})$ can be sampled from a replay buffer and $(\{\widehat{\boldsymbol{s}}'_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}'_{n,t}\}_{t=-H}^{-1}, \widehat{\boldsymbol{s}}'_{n,0}, \boldsymbol{a}'_{n,0})$ is set to be $(\{\widehat{\boldsymbol{s}}_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}_{n,t}\}_{t=-H}^{-1}, \widehat{\boldsymbol{s}}_{n,0}, \widehat{\boldsymbol{a}}_{n,0})$ when $t'=0$. For $t'=1,\ldots,T-1$, the action arguments of $r_{t'}$ are obtained from the policy $\pi_{\boldsymbol{\theta}^\pi}$ and the state arguments follow from the ensemble dynamics model $\{\widehat{T}^n_{\boldsymbol{\theta}_n}\}_{n=1}^{N_\mathrm{m}}$.

Remark 6. The loss (12) is an empirical approximation of the MSE between the action value function $Q^\pi_{\boldsymbol{\theta}^Q}$ and cumulative rewards $\sum_{t'=0}^{T-1} r_{t'}$ in the future T horizons. Thus, we can treat the optimization of $\boldsymbol{\theta}^Q$ as a regression problem. However, the target $\sum_{t'=0}^{T-1} r_{t'}$ is not fixed but related to the current policy, which is updated during iterations. Thus, we need to alternatively update the actor and critic during the learning process until both of them converge.

The empirical policy gradient to maximize J_k in (10) can be formulated as [26]

$$\widehat{\nabla_{\boldsymbol{\theta}^\pi} J_k} = \frac{1}{N'}\sum_{n=1}^{N'}\Big(\nabla_{\boldsymbol{a}} Q^\pi_{\boldsymbol{\theta}^Q}\left(\{\widehat{\boldsymbol{s}}_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}_{n,t}\}_{t=-H}^{-1}, \widehat{\boldsymbol{s}}_{n,0}, \boldsymbol{a}\right)\Big|_{\boldsymbol{a}=\pi_{\boldsymbol{\theta}^\pi}\left(\{\widehat{\boldsymbol{s}}_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}_{n,t}\}_{t=-H}^{-1}\widehat{\boldsymbol{s}}_{n,0}\right)} \times \nabla_{\boldsymbol{\theta}^\pi}\pi_{\boldsymbol{\theta}^\pi}\left(\{\widehat{\boldsymbol{s}}_{n,t}\}_{t=-H}^{-1}, \{\boldsymbol{a}_{n,t}\}_{t=-H}^{-1}, \widehat{\boldsymbol{s}}_{n,0}\right)\Big). \tag{13}$$

To calculate (12) and (13), we can randomly get samples from a replay buffer $\mathcal{R}$ which stores the recently collected data by interacting the policy with the learned dynamics model. The detailed algorithm is summarized in Algorithm 3 in Appendix B.3, available in the online supplemental material. This algorithm combines the actor-critic method with MPC, with the prediction horizon being T [26].

The architecture of the algorithm in this paper is presented in Fig. 2. Two training procedures need to be completed in the offline learning phase. First, we can find a sufficient number of volunteers with different thermal preferences to collect thermal feedback data and use these data to meta-train a thermal preference model of occupants. Second, the zone thermal dynamics model will be trained with the offline historical dataset. When the occupant's thermal preference needs to be identified, the

thermal preference model can be updated via meta-testing with a small number of feedbacks. Then, we can use the zone thermal dynamics model and the updated occupant's thermal preference model to retrain the actor-critic networks so that the personalized HVAC controller for the occupant can be obtained.

Remark 7. The algorithm designed in this paper can also be generalized to other domains. For example, in the human-robot collaboration settings, the authors in [78], [79] used safety-based fuzzy local controllers to identify the human intentions of motion so that the robot can assist the human operator in a better way. In [78], [79], the human intentions of motion were inferred using the interaction force, the derivative of the interaction force, and the end-effector velocity. However, human intentions may be too complicated in many cases, in which we can leverage a learning algorithm to obtain an intention prediction model. In addition, we can also possibly use the meta-learning algorithm to adapt a pre-trained intention prediction model to a new scenario. In this case, less human-robot interaction data may be required. Another setting is that we can give a feedback channel to the human operator and use the collected feedbacks to improve the performance of assistance the robot provides.

VI. Numerical Simulations on EnergyPlus

In this section, we will show the effectiveness of the proposed algorithm in an EnergyPlus environment[4]. All experiments and simulations are performed on Ubuntu 20.04 LTS with 32 GB RAM, 3.7 GHz×10 Intel Core™ i9-10900 K processor, and NVIDIA GeForce RTX 3080 GPU. The EnergyPlus testbed we used is introduced in Appendix C, available in the online supplemental material. Table VI in Section A.2, available in the online supplemental material, lists some terms and notations in this section.

A. Meta-Supervised Learning for Occupants' Thermal Feedbacks

This section will show the performances of the meta-supervised learning algorithm in learning the occupant's personalized thermal comfort model with both the ASHRAE database II and the synthetic data.

1) Experiments With the ASHRAE Database II: In this section, we will use the ASHRAE database II [54], [55] to illustrate the effectiveness of the meta-supervised learning algorithm, i.e., Algorithm 1, in improving the thermal sensation vote prediction accuracy with only a few personalized thermal feedback data. We consider the air temperature measured in the occupied zone as the only feature of the model and leave the multiple feature cases to our future work. Before using the data for model learning, we preprocess the database with the following procedures.

i) Since we consider the air temperature in the occupied zone as the only predictor of the model, the first step of our data preprocessing is to remove the environmental and personal conditions other than air temperature for each raw datum in the database.

ii) The ASHRAE database II has 110,070 thermal vote records from different individuals under different conditions. However, there might be some missed condition values for a certain record, denoted as "NA" in the original data file. The second step of our data preprocessing is to eliminate the raw data for which the value of indoor air temperature is indicated as "NA".

iii) In Algorithm 1, we need $K + K'$ thermal votes from each occupant to learn his/her thermal preference. In our experiment, we set $K = 5$ and $K' = 3$, respectively. Thus, if an occupant has less than $K + K' = 8$ thermal sensation votes in the dataset, then we will not consider this occupant when training our algorithm. In addition, if an occupant has more than 8 thermal votes, we use the first 8 data and eliminate the rest. After this data preprocessing step, we obtained $(5 + 3)$ data/occupant $\times$ $1,726$ occupant $= 13,808$ data[5].

iv) The air temperature is normalized with

$$T^{\text{in}} \triangleq \frac{T_0^{\text{in}} - \text{mean}(\{T_0^{\text{in}}\})}{\text{std}(\{T_0^{\text{in}}\})}, \tag{14}$$

with T_0^{in} being the original air temperature, and $\text{mean}(\{T_0^{\text{in}}\})$ and $\text{std}(\{T_0^{\text{in}}\})$ being the mean and standard deviation of T_0^{in}, respectively, over the dataset after preprocessing steps i) – iii).

The corresponding thermal sensations are normalized with

$$f(T^{\text{in}}) \triangleq \frac{\text{Original Thermal Sensation Under } T^{\text{in}}}{3}, \tag{15}$$

after which we have $f(T^{\text{in}}) \in [-1, 1]$.

After preprocessing steps i) – iv), we will use 8 tuples in the form of $(T^{\text{in}}, f(T^{\text{in}}))$ for each occupant to train and test the meta-supervised learning algorithm. In addition, we use the simple regression model as a benchmark that learns the relationship between T^{in} and $f(T^{\text{in}})$ but does not consider the occupants' identities. That is, the benchmark simply assumes that there is only one occupant who provides 13,808 thermal sensation votes under various conditions. This model method was also used in Sections 4.1 and 4.2 in [28].

Table IX in Appendix D.2, available in the online supplemental material, presents the hyperparameter settings when using the meta-supervised learning algorithm to learn the occupant's thermal preference model with ASHRAE database II. Note that we use a 3-layer fully-connected network as the architecture. For the simple supervised regression algorithm that does not consider the occupants' identities, we use the same neural network architecture and learning hyperparameters as shown in Table IX, expect that there are no concepts of K, K', and α.

Fig. 3 shows the training and validation RMSEs of both the meta-supervised algorithm, denoted as MAML, and the simple

4.See https://github.com/NREL/EnergyPlus/tree/v9.2.0.

5.There is a tradeoff between the number of occupants and the number of thermal sensation votes per occupant. More samples from an occupant and more occupants are both helpful to learn a better model. However, since the ASHRAE database II is fixed, if we require more thermal votes from an occupant, the number of occupants would be less after data preprocessing. For example, if we set $K + K' = 9$, then we can only have 9 data/occupant $\times$ $1,281$ occupant $=$ $11,529$ data, which is less than 13,808 in our experiment.

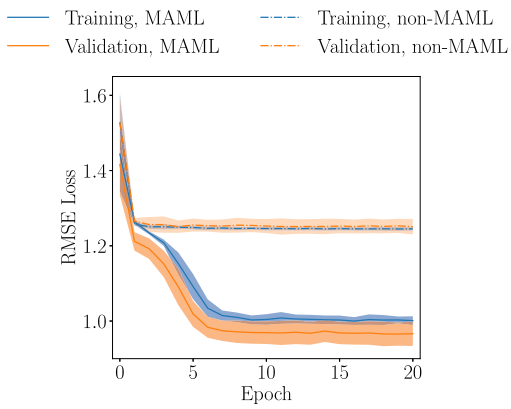

Fig. 3. The performance comparison of the meta-supervised learning algorithm (MAML) and the basic regression algorithm (non-MAML) on the ASHRAE database II.

regression algorithm, denoted as non-MAML. The RMSE loss in Fig. 3 is unnormalized back to the 7-point scale. We implemented both algorithms 5 times with different splits of training and validation datasets and initializations of neural network parameters. The curves in Fig. 3 are the means of the losses, with the shaded regions indicating one standard deviations around the mean. Fig. 3 illustrates that the meta-supervised learning algorithm can reduce both the training and validation RMSE losses from ~ 1.25 to ~ 1.00, which is a $\sim 20\%$ decrease compared with the simple supervised learning without using $K = 5$ personalized thermal feedbacks to adapt the model. In addition, the authors in [28] presented some results of using different regression models to predict the thermal sensation votes with the ASHRAE RP-884 thermal comfort dataset [53], which is a subset of the ASHRAE dataset. The best MSE Liu et al. [28] reported is 1.147, i.e., 1.071 in the sense of RMSE. Our meta-supervised learning algorithm reduces the RMSE $\sim 6.63\%$, compared with the best result in [28].

2) Experiments With Synthetic Data: In Section VI-A1, we demonstrated that the meta-learning algorithm could reduce the prediction loss of personalized thermal sensation votes in the ASHRAE database II. However, the ASHRAE database II does not provide the most preferred indoor temperatures of different occupants. Thus, we do not have a metric to test if the HVAC systems under our RL agent can satisfy different requirements of thermal preferences. In order to overcome this problem, we construct a generative model and use it to generate synthetic data for meta-learning.

According to the analyses in [45], the probability of feeling comfortable with respect to indoor air temperature is similar to a Gaussian function. The position of the center and standard deviation of the Gaussian function differ with different thermal preference patterns. In this section, we generate the thermal feedbacks with

$$f(T^{\text{in}}) = \text{clip}\left(\text{round}\left(4\exp\left(-\frac{(T^{\text{in}}-T^{\text{ref}})^2}{2\sigma^2}\right) -2+\mathcal{N}(0,\sigma_\epsilon^2)\right),-2,2\right), \tag{16}$$

where T^{ref} and σ are occupant-specific parameters, which are sampled according to $T^{\text{ref}} \sim \mathcal{U}[20.5^\circ\text{C}, 25.5^\circ\text{C}]$ and $\sigma \sim \mathcal{U}[1.5^\circ\text{C}, 2.0^\circ\text{C}]$. When generating samples, we only consider $T^{\text{in}} \in [19.0^\circ\text{C}, 27.0^\circ\text{C}]$. That is, any $T^{\text{in}} \notin [19.0^\circ\text{C}, 27.0^\circ\text{C}]$ will be considered directly as the indoor temperature making the occupant uncomfortable.

Remark 8. There are two steps to generate simulated data for the meta-supervised learning algorithm. In the first step, we specify the thermal preference parameters, i.e., T^{ref} and σ in (16), of an occupant. In the second step, for the specific values of T^{ref} and σ, we generate K samples, which simulate the case that a specific occupant gives K feedbacks.

Some explanations about (16) are given as follows.

i) The expression (16) is based on the Gaussian function

$$g(T^{\text{in}}) \triangleq 4\exp\left(-\frac{(T^{\text{in}}-T^{\text{ref}})^2}{2\sigma^2}\right) - 2, \tag{17}$$

with T^{ref} and σ being parameters. Note that $g(T^{\text{in}})$ is a continuous function in the range $[-2, 2]$, and reaches its maximum at $T^{\text{in}} = T^{\text{ref}}$.

ii) We sample T^{ref} and σ from their corresponding uniform distributions in the first data generation step. After that, the function $g(T^{\text{in}})$ is a deterministic function of T^{in}. However, according to Section III, we assume that there are only five thermal feedbacks that are mapped to values in the set $\{-2, -1, 0, 1, 2\}$. In addition, there are some subjective uncertainties when an occupant provides feedbacks. To simulate this scenario, we first add a small zero-mean Gaussian noise $\mathcal{N}(0, \sigma_\epsilon^2)$ to $g(T^{\text{in}})$ in (17), and then round the function value to the nearest integer. In this paper, we set $\sigma_\epsilon = 0.1$.

iii) Note that the operations of adding Gaussian noise and then rounding may lead function value being -3 or 3 in some extreme cases. We finally clip the function value in the range $[-2, 2]$. Finally, the range of the function $f(T^{\text{in}})$ in (16) is $\{-2, -1, 0, 1, 2\}$.

In our simulation, we set $K = 10$, which indicates that we expect to use 10 feedbacks when making adaptions for the controller. We use a 3-layer fully connected network as the model with hyperparameters given in Table X in Appendix D.2, available in the online supplemental material. Note that in order to train the model, we need to have 800 volunteers in practice, from each of whom $K + K' = 10 + 35 = 45$ thermal feedbacks should be collected at different indoor air temperatures.

In order to test the performance of meta-training, we generate $K = 10$ thermal feedbacks from each of the 6 different scenarios of (T^{ref}, σ): i. $(21.0^\circ\text{C}, 1.5^\circ\text{C})$; ii. $(22.0^\circ\text{C}, 1.6^\circ\text{C})$; iii. $(23.0^\circ\text{C}, 1.7^\circ\text{C})$; iv. $(24.0^\circ\text{C}, 1.8^\circ\text{C})$; v. $(25.0^\circ\text{C}, 1.9^\circ\text{C})$; vi. $(25.5^\circ\text{C}, 2.0^\circ\text{C})$.

The sampling of T^{ref} within a limited interval has an issue that the meta-testing performance with T^{ref} near the bounds of the interval will not be as good as that with T^{ref} near the center of the interval. The reasons are analyzed as follows. Let us consider our settings in which T^{ref}'s in the meta-training dataset are sampled from the distribution $\mathcal{U}[20.5^\circ\text{C}, 25.5^\circ\text{C}]$. The meta-testing performance with a specific T^{ref} depends on how well the model

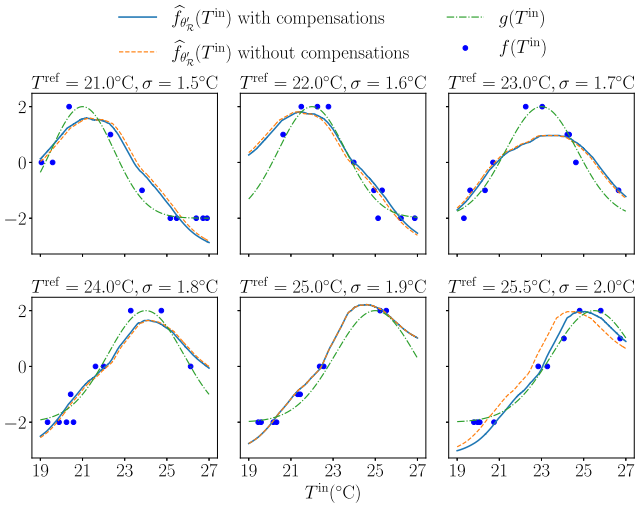

Fig. 4. The test performance of the meta-learning algorithm to learn different occupants' thermal preference models.

is trained around this specific value of T^{ref} so that generalization can be achieved. For example, the meta-training data with $T^{\text{ref}} \in [22.0°\text{C}, 24.0°\text{C}]$ are helpful to train a model which can adapt well with $T^{\text{ref}} = 23.0°\text{C}$ during meta-testing. However, for the meta-testing with $T^{\text{ref}} = 25.5°\text{C}$, only the meta-training data with $T^{\text{ref}} \in [24.5°\text{C}, 25.5°\text{C}]$ are helpful. In addition, the meta-training data with $T^{\text{ref}} \in [24.5°\text{C}, 25.5°\text{C}]$ are in fact biased when considering $T^{\text{ref}} = 25.5°\text{C}$ in meta-testing, which causes that the model after adaptation in meta-testing will implicitly identify a value of T^{ref} smaller than $25.5°\text{C}$. To compensate for this bias, after gradient adaptation steps during meta-testing, we use a grid search-based method to translate T^{ref} so that smaller losses can be achieved on the training data in meta-testing. Note that this is an additional step that will not hurt the meta-learning algorithm since the translation can be found to be zero if there is no need for this compensation. In addition, there is no requirement for an increased number of data. Fig. 4 shows the benefits of the compensation. The meta-testing algorithm with compensations is provided in Algorithm 2 in Appendix B.2, available in the online supplemental material.

Fig. 4 shows the meta-testing results. With $K = 10$ thermal feedbacks from an occupant, the model $\widehat{f}_{\boldsymbol{\theta}'_{\mathcal{R}}}(T^{\text{in}})$ can reasonably predict different patterns of thermal preferences after 5 gradient adaptation steps and compensations. The compensations to the model $\widehat{f}_{\boldsymbol{\theta}'_{\mathcal{R}}}(T^{\text{in}})$ improve the fitting performance when the T^{ref} is near to the bounds of the sampling interval $[20.5°\text{C}, 25.5°\text{C}]$ of T^{ref} for the meta-training data. Note that the thermal preference model can be further fine-tuned if the occupants would like to provide more thermal feedbacks. The trained model $\widehat{f}_{\boldsymbol{\theta}'_{\mathcal{R}}}(T^{\text{in}})$ will be used to regulate the RL algorithm in Section VI-C.

B. Thermal Dynamics Model Training With Offline Data

This section will show the performance of dynamics model learning with the simulated offline dataset generated by the EnergyPlus environment.

TABLE I
THE COMPARISONS OF THE AVERAGE TOTAL POWER CONSUMPTION AND AVERAGE PEAK POWER FOR THE PID CONTROLLER WITH AND WITHOUT EXPLORATIONS (SF)

	Avg. Power (kW)	Avg. Daily Peak (kW)
Without Explorations	110.01	129.59
With Explorations	110.55 ± 0.01	138.40 ± 0.13
Perf. Degradation	$0.49\% \pm 0.01\%$	$6.80\% \pm 0.10\%$

* Avg. = Average, Perf. = Performance.

The offline data are collected with a controller described as follows. This low-level controller needs to decide which action to make under a certain state. There are two requirements for this controller. i) The values of $T^{\text{in}}_{\text{west}}$ and $T^{\text{in}}_{\text{east}}$ should be in a reasonable range so that the occupants will typically not feel very uncomfortable. ii) The state and action spaces should be fully explored without violating of requirement i). In our simulation, we use PID controllers[6] to generate the actions $T^{\text{set}}_{\text{west}}$ and $T^{\text{set}}_{\text{east}}$, and get F_{west} and F_{east} via

$$F^{\text{west}} = (10.0\text{kg/s} - 2.5\text{kg/s}) \times F^{\text{west}}_0 + 2.5\text{kg/s}, \quad (18a)$$

$$F^{\text{east}} = (10.0\text{kg/s} - 2.5\text{kg/s}) \times F^{\text{east}}_0 + 2.5\text{kg/s}, \quad (18b)$$

with

$$F^{\text{west}}_0 \sim \text{Beta}(2.0, 8.0), \quad F^{\text{east}}_0 \sim \text{Beta}(2.0, 8.0). \quad (19)$$

Note that the values 2.5kg/s and 10.0kg/s in (18) are in accordance with $\boldsymbol{a}^{\min}$ and $\boldsymbol{a}^{\max}$ in (26).

Fig. 9 in Appendix E, available in the online supplemental material, shows the percentage of offline data with respect to indoor temperatures in the west and east zones. Most indoor air temperatures for both zones fall within the interval $[19°\text{C}, 28°\text{C}]$. The weather file associated with the offline data in Fig. 9 is the one with San Francisco (SF). For the other four cities, the offline datasets have similar statistics in terms of indoor air temperature. The simulated time length when generating the offline dataset is 1 year.

Remark 9. The samplings of F^{west}_0 and F^{east}_0 from the beta distribution (19) serve as explorations to the environment. The explorations of F^{west} and F^{east} would also add the diversities of $T^{\text{set}}_{\text{west}}$ and $T^{\text{set}}_{\text{east}}$, since diverse states would be encountered. If we do not consider explorations, we can fix $F^{\text{west}} = 4.0\text{kg/s}$ and $F^{\text{east}} = 4.0\text{kg/s}$. In this case, the variations of the offline dataset would be much smaller since the PID controller is essentially a deterministic controller. The learned thermal dynamics model would be only accurate in a small range due to the low variations of the offline data, which will influence the performance of the subsequent model-based RL agent. On the other hand, we also need to consider the safety of explorations. We consider two metrics in this paper to measure the safety of explorations: 1) the average power consumption and 2) the average daily peak power in the offline data. The authors in [27] proposed a safety criterion, i.e., the performance degradation of the controller with explorations should not be larger than a threshold $\epsilon = 10\%$ than the baseline controller without explorations. Table I shows the values of these two metrics for the PID controllers with and

6.See https://github.com/vermouth1992/mbrl-hvac.

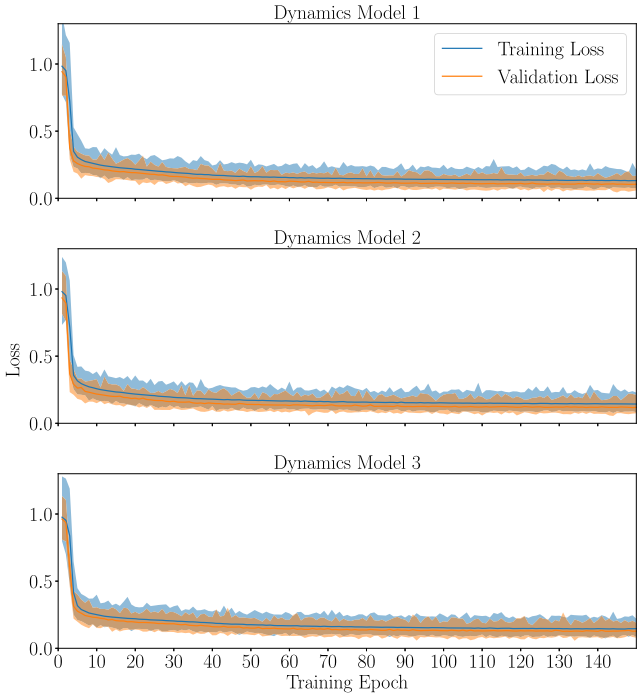

Fig. 5. The training and validation losses of the three individual dynamics model networks with the offline dataset (SF).

without explorations. Since the explorations (19) are random, we run the PID controller with explorations 5 times and present the one standard deviation areas around the means for both metrics. The performance degradations of both metrics are significantly less than 10%, which indicates that the explorations are safe.

The architecture of each individual dynamics model network $\widehat{f}^{n}_{\boldsymbol{\theta}_n}$, $n = 1, \ldots, N$, is given in Fig. 7 in Appendix D.1, available in the online supplemental material. Table XI in Appendix D.2, available in the online supplemental material, provides the parameter settings when training the thermal dynamics model.

We independently trained $N = 3$ deterministic networks by minimizing the MSE in (6). In order to increase the discrepancies of predictions for the three networks for the out-of-distribution datasets, we use different weight decays when training different networks. To be specific, the weight decay for the n-th network is set to be $(n-1) \times 10^{-4}$, $n = 1, 2, 3$. Fig. 5 presents the training and validation losses of the three individual dynamics models with the offline dataset. Both the training and validation losses decrease as the training epoch increases. In order to justify the use of the epistemic uncertainty metric (8), we tested the trained dynamics model on the in-distribution and out-of-distribution datasets, respectively. The statistics of the indoor air temperatures of the in-distribution and out-of-distribution are given as Figs. 9 and 10 in Appendix E, available in the online supplemental material, respectively. Table II shows the average discrepancies of the different networks' predictions defined as

$$\bar{d} \triangleq \frac{1}{N_{\mathcal{D}'}} \sum_{D_{t'} \in \mathcal{D}'} \frac{1}{\binom{N}{2}} \sum_{n_1, n_2 \in \{1,\ldots,N\}, n_1 \neq n_2} \left\| \widehat{T}^{n_1}_{\boldsymbol{\theta}_{n_1}} \left(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1} \right) - \widehat{T}^{n_2}_{\boldsymbol{\theta}_{n_2}} \left(\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1} \right) \right\|_2^2, \tag{20}$$

with $D_{t'} \triangleq (\{\boldsymbol{s}_t\}^{t'}_{t=t'-H+1}, \{\boldsymbol{a}_t\}^{t'}_{t=t'-H+1})$, on these two datasets. Note that we trained the thermal dynamics model with ensemble networks five times and took tests on the obtained ensemble networks. The results show that the value of $\bar{d}$ in the out-of-distribution dataset is much higher than that in the in-distribution dataset. This indicates that the designed uncertainty metric can punish actions leading to the out-of-distribution data region.

TABLE II
THE AVERAGE DISCREPANCIES OF THE DIFFERENT MODELS' PREDICTIONS ON THE IN-DISTRIBUTION AND OUT-OF-DISTRIBUTION DATASETS

Training #	Average discrepancy of models' predictions $\bar{d}$	
	In-distribution dataset	Out-of-distribution dataset
1	0.0673	2.8016
2	0.0666	3.8589
3	0.0536	4.3874
4	0.0523	3.4244
5	0.0652	8.9693
Average	0.0610	4.6883

C. Model-Based Reinforcement Learning With Different Thermal Preferences

This section will show the performance of the model-based RL algorithm in Section V with the considerations of different thermal preferences. In order to obtain different occupants' thermal preferences, we allow each occupant to provide 10 thermal feedbacks and adapt the meta-trained model in Section VI-A to form the final occupants' thermal preference prediction models. In addition, we assume that there are two individuals in the two zones who may have different thermal preferences.

In our simulation, the reward function terms in (9) are defined as follows. Compared with the reward functions in the existing literature about RL for HVAC systems, such as [18], [25], [26], [28], [80], one advantage of this reward function is that it reflects the personalized thermal preference via the term $\widehat{r}^{\mathrm{o}}_{t'}$.

$$r^{\mathrm{t}}_{t'} = \sum_{l \in \{\mathrm{west}, \mathrm{east}\}} \left(\left[T^{\mathrm{in}}_{\mathrm{lower}} - \widehat{T}^{\mathrm{in}}_{l,t'} \right]_+ + \left[\widehat{T}^{\mathrm{in}}_{l,t'} - T^{\mathrm{in}}_{\mathrm{upper}} \right]_+ \right), \tag{21}$$

$$r^{\mathrm{e}}_{t'} = \sum_{l \in \{\mathrm{west}, \mathrm{east}\}} \left(\widehat{P}_{\mathrm{ITE},l,t'} + \widehat{P}_{\mathrm{HVAC},l,t'} \right), \tag{22}$$

$$\widehat{r}^{\mathrm{o}}_{t'} = \sum_{l \in \{\mathrm{west}, \mathrm{east}\}} \widehat{f}_{\boldsymbol{\theta}^l_{\mathcal{R}}} \left(\widehat{T}^{\mathrm{in}}_{l,t'} \right) \tag{23}$$

$$r^{\mathrm{u}}_{t'} = \bar{d}, \tag{24}$$

where i) we set $T^{\mathrm{in}}_{\mathrm{lower}} = 19.0^{\circ}\mathrm{C}$ and $T^{\mathrm{in}}_{\mathrm{upper}} = 27.0^{\circ}\mathrm{C}$, ii) $\widehat{T}^{\mathrm{in}}_{l,t'}$ denotes the indoor air temperature in the zone l, with $l \in \{\mathrm{west}, \mathrm{east}\}$, at time step t', iii) $\widehat{P}_{\mathrm{ITE},l,t'}$ and $\widehat{P}_{\mathrm{HVAC},l,t'}$ are defined in the similar manner as $\widehat{T}^{\mathrm{in}}_{l,t'}$, iv) the learned occupant's thermal preference model $\widehat{f}_{\boldsymbol{\theta}^l_{\mathcal{R}}}$ is related to the zone l since different thermal preferences are assumed for occupants

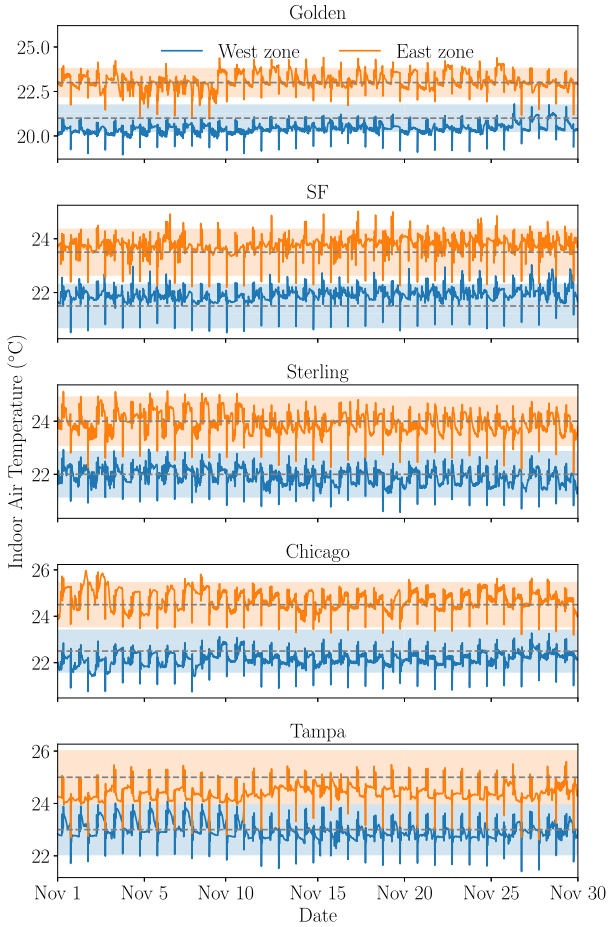

Fig. 6. The indoor air temperatures of west and east zones under the weather files in five different cities.

in different zones, and v) $\bar{d}$ is defined in (20). The hats above variables $\widehat{T}^{\text{in}}_{l,t'}$, $\widehat{P}_{\text{ITE},l,t'}$, and $\widehat{P}_{\text{HVAC},l,t'}$ indicate that these variables are predicted by the dynamics model $\{\widehat{T}^{n}_{\boldsymbol{\theta}_n}\}_{n=1}^{N_{\text{m}}}$, instead of the real values collected from the environment. In addition, we set $\lambda_1 = 1.0$, $\lambda_2 = 1.0 \times 10^{-5}$, $\lambda_3 = 0.25$, and $\lambda_4 = 0.5$, respectively, in our simulation.

Table VIII in Appendix D.2, available in the online supplemental material, shows the combinations of weather files and occupants with different thermal preferences in our simulations. The architectures of the critic and actor networks are given in Fig. 8 in Appendix D.1, available in the online supplemental material. Table XII in Appendix D.2, available in the online supplemental material, presents the training configurations.

1) Result Analyses of the Proposed Algorithm: Fig. 6 shows the indoor air temperatures of west and east zones under the weather files in five different cities. The shaded areas indicate the temperature ranges in which the occupants should typically feel comfortable, which is predefined instead of learned. That is, if we do not consider the subjective Gaussian noise in (16), an occupant should give feedback $+2$ when the indoor air temperature is in the shaded region. It is shown in Fig. 6 that the indoor air temperatures are typically within the comfortable ranges. The dashed lines denote the centers of the preferred indoor temperature ranges. Note the differences between these comfortable indoor temperature ranges. Fig. 6 indicates that the indoor air temperature would be around different values that reflect different thermal preferences of occupants. This further illustrates the effectiveness of the reward term $\widehat{r}^{\text{o}}$ and thus the meta-learning algorithm with a few thermal feedbacks. The periodical and occasional violations of the comfortable ranges are due to the data center CPU power changes, as is presented in Table VII in Appendix C, available in the online supplemental material.

In order to show the performance improvement of our algorithm, we compare it with the PID controller without explorations. That is, we use PID controllers to get the actions $T^{\text{set}}_{\text{west}}$ and $T^{\text{set}}_{\text{east}}$ and fix both F^{west} and F^{east} to be 4.0 kg/s. The target indoor air temperature varies according to different thermal preferences in Table VIII. The comparisons of our algorithm with the PID controller are given in Table V, in which the total power consumptions, square indoor temperature deviations, and rewards are the averaged values after the convergence of algorithms under the weather files in November for all the five cities. Our algorithm achieves 56.11% less average square temperature deviation with 1.44% less power consumption than the PID controller.

2) Influences of Reward Weights to the Controller Performance: The reward weights $\lambda_1, \ldots, \lambda_4$ have large impacts on the controller performance since they determine the relative importance of different factors when training the algorithm. Table III presents the results of average total power consumption and average square temperature deviation with different values of reward weights. We use the weather data in SF and run the Algorithm 5 times for each combination of weights. Table III shows the ranges of one standard deviations around the means. For each weight combination, both power consumption and square temperature deviation are averaged over the results after algorithm convergence with the weather data in November. In addition, the square temperature deviation is also averaged over the two zones. The base temperature values to calculate the temperature deviations are 21.5 °C and 23.5 °C for the west and east zones, respectively, which are in accordance with Table VIII. Note that we did not change the value of λ_1 since only the relative importance of these weights matters. The result analyses are as follows.

i) When we increase the value of λ_2 or decrease the value of λ_3, i.e., we care more about energy consumption and less about occupants' thermal comfort, there are obvious increases for the average square temperature deviations. However, the power consumptions do not drop significantly. Likewise, when we decrease the value of λ_2 or increase the value of λ_3, i.e., we are more concerned about the thermal comfort, the power consumptions slightly increase, while the average square temperature deviations do not have significant decreases. This indicates that the weights λ_2 and λ_3 can reflect the balance between power consumption and occupants' thermal comfort. In order to achieve the best performance of both metrics, we need to find good combinations of these two hyperparameters. Note that we used relatively large scaling values when

TABLE III
THE CONTROLLER PERFORMANCES WITH DIFFERENT REWARD WEIGHTS (SF)

Reward Weights ($[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$)	Avg. Total Power (kW)	Avg. Sq. Temp. Dev. ($^\circ$C^2)
$[1.0, 1.0 \times 10^{-5}, 0.25, 0.50]$	109.70 ± 0.44	0.1510 ± 0.0295
$[1.0, \mathbf{5.0 \times 10^{-5}}, 0.25, 0.50]$	109.71 ± 1.89	0.6060 ± 0.3804
$[1.0, \mathbf{0.2 \times 10^{-5}}, 0.25, 0.50]$	110.12 ± 0.14	0.1764 ± 0.0808
$[1.0, 1.0 \times 10^{-5}, \mathbf{1.25}, 0.50]$	110.18 ± 0.84	0.1691 ± 0.1010
$[1.0, 1.0 \times 10^{-5}, \mathbf{0.10}, 0.50]$	109.43 ± 0.49	0.2700 ± 0.0674
$[1.0, 1.0 \times 10^{-5}, 0.25, \mathbf{10.0}]$	110.32 ± 0.42	0.3711 ± 0.1770
$[1.0, 1.0 \times 10^{-5}, 0.25, \mathbf{2.00}]$	109.96 ± 0.28	0.1308 ± 0.0491
$[1.0, 1.0 \times 10^{-5}, 0.25, \mathbf{0.10}]$	109.75 ± 1.52	0.3039 ± 0.0793
$[1.0, 1.0 \times 10^{-5}, 0.25, \mathbf{0.00}]$	121.65 ± 12.04	14.4958 ± 14.2441

* Avg. = Average, Sq. = Square, Temp. = Temperature, Dev. = Deviation.

TABLE IV
THE AVERAGE INDOOR AIR TEMPERATURES OF THE WEST AND EAST ZONES AND AVERAGE SQUARE TEMPERATURE DEVIATIONS WHEN $\lambda_4 = 0$ (SF)

	$T^{\text{in}}_{\text{west}}$ ($^\circ$C)	$T^{\text{in}}_{\text{east}}$ ($^\circ$C)	Sq. Temp. Dev. ($^\circ$C^2)
1	21.28	31.36	31.32
2	25.21	22.96	7.87
3	22.56	24.15	1.02
4	21.78	31.44	31.96
5	21.32	24.00	0.31

* Sq. Temp. Dev. = Square Temperature Deviation.

changing λ_2 and λ_4 in Table III. During our simulations, we found that the controller performance is rather robust to the selections of these two weights. Based on the default weights $\lambda_2 = 1.0 \times 10^{-5}$ and $\lambda_3 = 0.25$, if we slightly change the weights, then the final controller performance would not change too much.

ii) Compared with λ_2 and λ_3, the scaling of λ_4, which punishes model exploitations in the out-of-distribution region with respect to the offline dataset, has a larger impact on the controller performance. If we set it to $\lambda_4 = 0$, then there is significant performance deterioration for the learned controller in both power consumption and temperature deviation aspects. The average indoor temperatures and square temperature deviations for our 5 runs when $\lambda_4 = 0$ are shown in Table IV. In the first and fourth runs, the indoor air temperatures in the east zone are above 30 $^\circ$C, which are out-of-distribution states according to Fig. 9 in Appendix E, available in the online supplemental material. This implies that the reward term r^{u} is necessary in the offline training case to avoid possible model exploitations in the out-of-distribution region. In the second run, even if we do not observe out-of-distribution states, the indoor air temperature in the west zone is not well regulated. When $\lambda_4 \neq 0$, we also need to choose an appropriate value of λ_4. If it is too small, the out-of-distribution actions will not be punished enough. If it is too large, then the reward term r^u may dominate the whole reward, which weakens the trade-off between the power consumptions of HVAC systems and occupants' thermal comfort levels. As illustrated in Table III, both the small $\lambda_4 = 0.10$ and the large $\lambda_4 = 10.0$ would increase the average square temperature deviation.

D. Comparisons With the Model-Based Online and Model-Free Offline Reinforcement Learning Algorithms

In this section, we will compare our algorithm with the model-based RL with on-policy data aggregation [26] and a model-free offline RL algorithm, i.e., conservative Q-learning [27], [37], respectively.

1) Comparisons With the Model-Based Reinforcement Learning Algorithm With On-Policy Data Aggregation: One potential drawback of the offline RL is that the learning is limited to the "known" region that the offline dataset covers [31], [32]. This limited exploration may lead to a suboptimal policy since the global optimal policy may be in the "unknown" region [36]. That is, although Fig. 6 shows that the indoor temperatures are typically within the ranges that occupants should be comfortable with, these results might come with increased power consumptions.

In this section, we compare the power consumption of our algorithm with its "online version," i.e., we allow the policy to collect more data from the EnergyPlus environment and then use the aggregated dataset to update further the thermal dynamics model [81]. For the compared algorithm, we first train the thermal dynamics model with the offline dataset. In the first 35 iterations, only the actor and critic networks are trained. Then the actor is used to collect more transition data from the EnergyPlus environment, which will be used to update the thermal dynamics model in the remaining iterations. In addition, we let the reward term $r^{\text{u}}_t = 0$, $\forall t$, for the compared algorithm such that the exploration is not limited by the offline dataset. The number of iterations for the actor and critic networks is 330, while the dynamics model is trained every 7 iterations with training epochs being 50. The other training hyper-parameters of the compared algorithm are the same as those in Table XII in Appendix D.2, available in the online supplemental material.

The comparison results in Table V show that the average power consumption of our algorithm is only 1.91%, averaged over the five cities, more than that of the RL algorithm with on-policy data aggregation. This indicates that with only the offline dataset, the power consumption of our algorithm is still similar to that of the counterpart trained after a long time of on-policy data collection. However, the average square indoor air temperature deviation of our algorithm is 33.59% smaller on average than its online counterpart. We hypothesize that this phenomenon is due to the over-exploration of the online policy

TABLE V
THE COMPARISONS BETWEEN THE PROPOSED AND SOME EXISTING ALGORITHMS

Algorithm	Metric	City					Average
		Golden	SF	Sterling	Chicago	Tampa	
Model-based offline algorithm (The proposed algorithm)	Avg. Total Power (kW)	96.12	109.78	98.16	100.44	120.61	105.02
	Avg. Sq. Temp. Dev. ($°C^2$)	0.3441	0.1467	0.0870	0.1848	0.1952	0.1916
	Avg. Reward	0.6234	0.6707	0.5937	0.6961	0.5429	0.6254
Model-based online algorithm	Avg. Total Power (kW)	91.33	107.87	96.75	99.88	119.40	103.05
	Avg. Sq. Temp. Dev. ($°C^2$)	0.3368	0.3203	0.3285	0.2483	0.2086	0.2885
	Avg. Reward	0.8125	0.5072	0.6380	0.8146	0.5749	0.6694
Model-free offline algorithm (CQL)	Avg. Total Power (kW)	98.20	110.23	100.38	103.83	123.30	107.19
	Avg. Sq. Temp. Dev. ($°C^2$)	1.2727	0.4861	0.5316	0.8817	0.8656	0.8075
	Avg. Reward	0.5352	0.5152	0.5987	0.5246	0.3526	0.5053
CQL ($\lambda_3 = 1.25$)	Avg. Total Power (kW)	97.67	110.80	100.00	102.93	121.39	106.56
	Avg. Sq. Temp. Dev. ($°C^2$)	1.4734	0.6626	0.6601	0.5423	0.8397	0.8356
PID controller without explorations	Avg. Total Power (kW)	92.47	108.37	100.44	105.73	125.73	106.55
	Avg. Sq. Temp. Dev. ($°C^2$)	0.7390	0.4394	0.3347	0.3276	0.3420	0.4365

* Avg. = Average, Sq. = Square, Temp. = Temperature, Dev. = Deviation.

on the dynamics regions with low power consumptions. Note that the thermal dynamics model in our algorithm can also be updated when more on-policy data are collected after deployment in practice [82], [83]. In addition, we can also leverage various exploration strategies, such as the upper confidence bound (UCB) algorithm [84], [85], ϵ-greedy algorithm [9], during the process of collecting online data. Thus, for our algorithm, we can also reduce the increased power consumption further with more on-policy data available.

2) Comparisons With the Model-Free Conservative Q-Learning Algorithm: Ref. [27] presented an offline RL strategy for HVAC systems based on the model-free CQL algorithm [37]. In this section, we compare it with our mode-based offline RL algorithm to demonstrate the energy efficiency and superiority in indoor air temperature regulation of our algorithm.

We use the same network architectures (see Fig. 8 in Appendix D.1, available in the online supplemental material) of both actor and critic for the CQL algorithm as those in our algorithm. The settings of hyperparameters of CQL are shown in Table XIII in Appendix D.2, available in the online supplemental material. Since the CQL algorithm in [37] is built on the SAC algorithm, some hyperparameters in Table XIII are adapted from the SAC algorithm [17]. We tried our best to tune the hyperparameters in Table XIII for fair comparisons.

The results in Table V show that the performance, especially the average square temperature deviation, of CQL is significantly poorer than that of the proposed algorithm. We also tried to increase the reward weight λ_3 from 0.25 to 1.25 to see if the average square temperature deviation drops. Unfortunately, we did not observe this result. The reasons are analyzed as follows.

Based on the SAC algorithm, CQL uses an empirical Bellman equation iteratively and considers a regularization term when minimizing the Bellman error at each iteration. Thus, the objective it optimizes is the discounted cumulative reward in an infinite horizon[7] $\sum_{t=1}^{\infty} \gamma^t r_t$. However, the dynamics of the environment of HVAC systems is highly stochastic. For example, the weather data would be of high randomness. In this case, the long-term predictions would be unreliable, especially when the dataset is offline with a limited number of data. This implies that it is better to optimize the finite discounted cumulative reward $\sum_{t=1}^{T} \gamma^t r_t$ since the term $\sum_{t=T+1}^{\infty} \gamma^t r_t$ would not help but increasing the variance [60]. When tuning the hyperparameters of CQL, we found that the discount factor γ has a large impact on policy performance. If we increase it from $\gamma = 0.85$ to 0.95, the policy performance would be worse. For our model-based RL algorithm, we can use an MPC planning algorithm with a finite T. However, the current CQL framework cannot incorporate this setting. In our paper, we used the MPC prediction horizon $T = 5$. It is worth noting that increasing T would also hurt the final policy performance.

In addition, although the performance of CQL is worse than the PID controller without explorations, it is much better than the PID controller with explorations which generates the offline dataset. The improvement can be found by comparing the average square temperature deviations in Table V and Fig. 9 in Appendix E, available in the online supplemental material.

Remark 10. This is a general remark about the nonstationarities of the environment and occupants' thermal preferences. i) The thermal dynamics of the environment may be nonstationary due to various factors, such as seasonal variations. Two strategies can be used to handle this issue. The first one is to include data from different conditions in the offline dataset used to train the dynamics model. The second one is training the dynamics model with only the data from a specific condition that is considered. When the environmental conditions have changed, we retrain the model and policy with a new dataset. In our simulation, we adopted the first strategy. That is, in order to handle seasonal variations of the environment dynamics, we collect the offline dataset with a noisy PID controller *for one year*. We select this strategy because it is more robust than the second one. For example, the controller specific to a certain condition may perform poorly when the environment changes unexpectedly. ii) The occupants' thermal comfort models may also be nonstationary. According to Fanger's PMV model [48], many factors influence an occupant's thermal comfort. These factors include air flow rate, metabolic rate, clothing level, etc. This paper considers

7.We omit the policy entropy term introduced by using SAC and regularization term due to the conservative learning for brevity, which would not influence the analysis here.

air temperature as the only factor influencing thermal comfort. If an occupant's thermal preference model has changed, then he/she can provide more feedbacks so that the model can be updated. One potential future research direction is adding more factors to the meta-learning framework. When these factors are not available, we may use some existing models to predict them. For example, Schiavon et al. [72] proposed a model to predict the clothing levels of occupants based on the outdoor air and indoor operative temperatures.

VII. Conclusion

This paper designed a human-in-the-loop energy-efficient HVAC control algorithm via meta-learning and model-based offline RL. The occupant's thermal preference model is trained via meta-learning, and the thermal dynamics model is fitted by an ensemble neural network. A model-based RL algorithm that combines MPC and actor-critic framework is used to obtain the optimal HVAC controller. The proposed algorithm contributes to a more practical deployment of the RL algorithm to HVAC systems since it is trained mainly with offline datasets with only a few online thermal feedbacks required.

We used the ASHRAE database II to demonstrate the effectiveness and advantage of the meta-learning algorithm in predicting personalized thermal sensation votes. Under the meta-learning framework, we use $K = 5$ personalized votes to adapt the meta-trained model. The obtained result indicates that the meta-learning algorithm is better than all the 5 regression algorithms in [28]. In addition, by the numerical simulations with EnergyPlus, we showed that our RL algorithm could guarantee the occupants' personalized thermal preferences and consumes little additional power compared with the model-based RL algorithm with on-policy aggregation. These results indicate that the use of offline datasets without further on-policy data aggregation may not introduce too much suboptimality in some scenarios. However, it gives us the advantage of training the algorithm much shorter time. In contrast, a long time is typically needed to collect on-policy data in practical HVAC control situations.

There are some possible directions for future work. First, to further reduce the required number of thermal feedbacks from an occupant, we may design a strategy combining meta-learning and active learning based on uncertain estimates to schematically acquire thermal feedbacks [50], [51], [86], [87]. Second, hierarchical RL may be able to be used to design an HVAC controller that has different outputs under different scenarios. For example, in the EnergyPlus environment in this paper, the scheduled CPU power of the two-zone data center varies at different times in a day. In this case, a hierarchical RL algorithm might be designed to use different control strategies at different times so that more thermal comfort and/or less power consumption can be achieved.

References

[1] S. Koebrich, T. Bowen, and A. Sharpe, "2018 renewable energy data book," National Renewable Energy Lab. (NREL), Golden, CO, USA, Tech. Rep. NREL/BK-6A20-75284, 2020. [Online]. Available: https://www.nrel.gov/docs/fy20osti/75284.pdf

[2] L. Chen and Y. Zhang, "Accelerated distributed model predictive control for HVAC systems," *Control Eng. Pract.*, vol. 110, 2021, Art. no. 104782.

[3] G. Bianchini, M. Casini, D. Pepe, A. Vicino, and G. G. Zanvettor, "An integrated model predictive control approach for optimal HVAC and energy storage operation in large-scale buildings," *Appl. Energy*, vol. 240, pp. 327–340, 2019.

[4] Y. Yao and D. K. Shekhar, "State of the art review on model predictive control (MPC) in heating ventilation and air-conditioning (HVAC) field," *Building Environ.*, vol. 200, 2021, Art. no. 107952.

[5] J. Zhao, K. P. Lam, B. E. Ydstie, and O. T. Karaguzel, "Energyplus model-based predictive control within design–build–operate energy information modelling infrastructure," *J. Building Perform. Simul.*, vol. 8, no. 3, pp. 121–134, 2015.

[6] F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, and G. P. Vanoli, "A new methodology for cost-optimal analysis by means of the multi-objective optimization of building energy performance," *Energy Buildings*, vol. 88, pp. 78–90, 2015.

[7] F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, and G. P. Vanoli, "Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort," *Energy Buildings*, vol. 111, pp. 131–144, 2016.

[8] Z. Zhang, A. Chong, Y. Pan, C. Zhang, and K. P. Lam, "Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning," *Energy Buildings*, vol. 199, pp. 472–490, 2019.

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

[10] G. Wen, C. P. Chen, and B. Li, "Optimized formation control using simplified reinforcement learning for a class of multiagent systems with unknown dynamics," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 7879–7888, Sep. 2019.

[11] G. Wen, L. Xu, and B. Li, "Optimized backstepping tracking control using reinforcement learning for a class of stochastic nonlinear strict-feedback systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1291–1303, Mar. 2023.

[12] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[13] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[14] D. Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[15] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 31st Int. Conf. Mach. Learn.*, Lille, France: PMLR, 2015, pp. 1889–1897.

[16] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Representations*, San Juan, Puerto Rico, USA, 2016, pp. 1–14.

[17] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden: PMLR, 2018, pp. 1861–1870.

[18] T. Wei, Y. Wang, and Q. Zhu, "Deep reinforcement learning for building HVAC control," in *Proc. 54th Annu. Des. Automat. Conf.*, Austin, TX, USA, 2017, pp. 1–6.

[19] X. Deng, Y. Zhang, and H. Qi, "Towards optimal HVAC control in non-stationary building environments combining active change detection and deep reinforcement learning," *Building Environ.*, vol. 168, 2022, Art. no. 108680.

[20] M. Biemann, F. Scheller, X. Liu, and L. Huang, "Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control," *Appl. Energy*, vol. 298, 2021, Art. no. 117164.

[21] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden: PMLR, 2018, pp. 1587–1596.

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

[23] L. Kaiser et al., "Model-based reinforcement learning for Atari," in *Proc. 8th Int. Conf. Learn. Representations*, Virtual Conference, Formerly Addis Ababa, Ethiopia, 2020, pp. 1–28.

[24] T. M. Moerland, J. Broekens, and C. M. Jonker, "Model-based reinforcement learning: A survey," 2020, *arXiv:2006.16712*.

[25] 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 Buildings Cities Transp.*, New York, NY, USA, 2019, pp. 287–296.

[26] 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.
[27] C. Zhang, S. R. Kuppannagari, and V. K. Prasanna, "Safe building HVAC control via batch reinforcement learning," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 4, pp. 923–934, Fourth Quarter 2022.
[28] H.-Y. Liu, B. Balaji, S. Gao, R. Gupta, and D. Hong, "Safe HVAC control via batch reinforcement learning," in *Proc. ACM/IEEE 13th Int. Conf. Cyber- Phys. Syst.*, Milan, Italy, 2022, pp. 181–192.
[29] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," 2020, *arXiv:2005.01643*.
[30] G. An, S. Moon, J.-H. Kim, and H. O. Song, "Uncertainty-based offline reinforcement learning with diversified Q-ensemble," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 7436–7447.
[31] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims, "MOReL: Model-based offline reinforcement learning," in *Proc. 34th Conf. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2020, pp. 21810–21823.
[32] T. Yu et al., "MOPO: Model-based offline policy optimization," in *Proc. 34th Conf. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2020, pp. 14129–14142.
[33] Z. Jiang et al., "Building HVAC control with reinforcement learning for reduction of energy cost and demand charge," *Energy Buildings*, vol. 239, 2021, Art. no. 110833.
[34] C. Blad, S. Bøgh, and C. S. Kallesøe, "Data-driven offline reinforcement learning for HVAC-systems," *Energy*, vol. 261, 2022, Art. no. 125290.
[35] Z. Zou, X. Yu, and S. Ergan, "Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network," *Building Environ.*, vol. 168, 2020, Art. no. 106535.
[36] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy Q-learning via bootstrapping error reduction," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 11761–11771.
[37] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," in *Proc. 34th Conf. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, 2020, pp. 1179–1191.
[38] H.-Y. Liu, B. Balaji, R. Gupta, and D. Hong, "Offline reinforcement learning with munchausen regularization," in *Proc. Offline Reinforcement Learn. Workshop Neural Inf. Process. Syst.*, 2021, pp. 1–6.
[39] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA: PMLR, 2019, pp. 2052–2062.
[40] T. Yu et al., "COMBO: Conservative offline model-based policy optimization," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 28954–28967.
[41] M. Rigter, B. Lacerda, and N. Hawes, "RAMBO-RL: Robust adversarial model-based offline reinforcement learning," in *Proc. 36th Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022, pp. 1–14.
[42] A. Argenson and G. Dulac-Arnold, "Model-based offline planning," in *Proc. 9th Int. Conf. Learn. Representations*, 2021, pp. 1–25.
[43] 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. Automat.*, Brisbane, Australia, 2018, pp. 7559–7566.
[44] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 12498–12509.
[45] S. Lee, I. Bilionis, P. Karava, and A. Tzempelikos, "A Bayesian approach for probabilistic classification and inference of occupant thermal preferences in office buildings," *Building Environ.*, vol. 118, pp. 323–343, 2017.
[46] S. Lee, P. Karava, A. Tzempelikos, and I. Bilionis, "Inference of thermal preference profiles for personalized thermal environments with actual building occupants," *Building Environ.*, vol. 148, pp. 714–729, 2019.
[47] S. Lee, J. Joe, P. Karava, I. Bilionis, and A. Tzempelikos, "Implementation of a self-tuned HVAC controller to satisfy occupant thermal preferences and optimize energy use," *Energy Buildings*, vol. 194, pp. 301–316, 2019.
[48] P. O. Fanger, *Thermal Comfort: Analysis and Applications in Environmental Engineering*. Copenhagen, Denmark: Danish Technical Press, 1970.
[49] J. J. Aguilera, O. B. Kazanci, and J. Toftum, "Thermal adaptation in occupant-driven HVAC control," *J. Building Eng.*, vol. 25, 2019, Art. no. 100846.
[50] A. Bemporad and D. Piga, "Global optimization based on active preference learning with radial basis functions," *Mach. Learn.*, vol. 110, no. 2, pp. 417–448, 2021.
[51] L. Roveda et al., "Pairwise preferences-based optimization of a path-based velocity planner in robotic sealing tasks," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6632–6639, Oct. 2021.
[52] L. Roveda, P. Veerappan, M. Maccarini, G. Bucca, A. Ajoudani, and D. Piga, "A human-centric framework for robotic task learning and optimization," *J. Manuf. Syst.*, vol. 67, pp. 68–79, 2023.
[53] R. J. De Dear, "A global database of thermal comfort field experiments," *ASHRAE Trans.*, vol. 104, 1998, Art. no. 1141.
[54] V. F. Ličina et al., "Development of the ASHRAE global thermal comfort database II," *Building Environ.*, vol. 142, pp. 502–512, 2018.
[55] T. Parkinson et al., *ASHRAE Glob. Database Thermal Comfort Field Meas.*. Dryad, 2022. [Online]. Available: https://doi.org/10.6078/D1F671
[56] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, Australia: PMLR, 2017, pp. 1126–1135.
[57] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006.
[58] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. Berlin, Germany: Springer, 2006.
[59] Z. Xu, H. P. van Hasselt, and D. Silver, "Meta-gradient reinforcement learning," in *Proc. 32nd Conf. Adv. Neural Inf. Process. Syst.*, Montréal, Canada, 2018, pp. 2402–2413.
[60] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Proc. 32nd Conf. Adv. Neural Inf. Process. Syst.*, Montréal, Canada, 2018, pp. 4759–4770.
[61] A. Rajeswaran, I. Mordatch, and V. Kumar, "A game theoretic framework for model based reinforcement learning," in *Proc. 37th Int. Conf. Mach. Learn.*, Vienna, Austria: PMLR, 2020, pp. 7953–7963.
[62] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 6405–6416.
[63] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped DQN," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 4033–4041.
[64] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA: PMLR, 2016, pp. 1050–1059.
[65] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on Bayesian neural networks—A tutorial for deep learning users," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 29–48, May 2022.
[66] P. Cole and R. Hart, "Intro to commercial building HVAC systems and energy code requirements," 2019. [Online]. Available: https://www.energycodes.gov/sites/default/files/2019--09/HVAC_Systems_Webinar_Transcript.pdf
[67] M. Maasoumy, A. Pinto, and A. Sangiovanni-Vincentelli, "Model-based hierarchical optimal control design for HVAC systems," in *Proc. ASME Dyn. Syst. Control Conf.*, Arlington, VA, USA, 2011, pp. 271–278.
[68] A. Bryson, *Applied Optimal Control: Optimization, Estimation and Control*. Boca Raton, FL, USA: CRC, 1975.
[69] N. K. Dhar, N. K. Verma, and L. Behera, "Adaptive critic-based event-triggered control for HVAC system," *IEEE Trans. Ind. Inform.*, vol. 14, no. 1, pp. 178–188, Jan. 2018.
[70] L. Meng, R. Gorbet, and D. Kulić, "Memory-based deep reinforcement learning for POMDPs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Prague, Czech Republic, 2021, pp. 5619–5626.
[71] M. A. Humphreys and J. F. Nicol, "The validity of ISO-PMV for predicting comfort votes in every-day thermal environments," *Energy Buildings*, vol. 34, no. 6, pp. 667–684, 2002.
[72] S. Schiavon and K. H. Lee, "Dynamic predictive clothing insulation models based on outdoor air and indoor operative temperatures," *Building Environ.*, vol. 59, pp. 250–260, 2013.
[73] M. Luo et al., "Comparing machine learning algorithms in predicting thermal sensation using ASHRAE comfort database II," *Energy Buildings*, vol. 210, 2020, Art. no. 109776.
[74] T. M. Moerland, J. Broekens, and C. M. Jonker, "Learning multimodal transition dynamics for model-based reinforcement learning," in *Proc. 1st Scaling-Up Reinforcement Learn. Workshop*, 2017, pp. 1–18.

[75] N. Tagasovska and D. Lopez-Paz, "Single-model uncertainties for deep learning," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 6414–6425.
[76] Y. Gal, "Uncertainty in deep learning," PhD dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2016. [Online]. Available: https://www.cs.ox.ac.uk/people/yarin.gal/website/thesis/thesis.pdf
[77] A. Malik, V. Kuleshov, J. Song, D. Nemer, H. Seymour, and S. Ermon, "Calibrated model-based deep reinforcement learning," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA: PMLR, 2019, pp. 4314–4323.
[78] A. Mauri et al., "Mechanical and control design of an industrial exoskeleton for advanced human empowering in heavy parts manipulation tasks," *Robotics*, vol. 8, no. 3, 2019, Art. no. 65.
[79] L. Roveda, S. Haghshenas, M. Caimmi, N. Pedrocchi, and L. M. Tosatti, "Assisting operators in heavy industrial tasks: On the design of an optimized cooperative impedance fuzzy-controller with embedded safety rules," *Front. Robot. AI*, vol. 6, 2019, Art. no. 75.
[80] T. Moriyama, G. D. Magistris, M. Tatsubori, T.-H. Pham, A. Munawar, and R. Tachibana, "Reinforcement learning testbed for power-consumption optimization," in *Proc. Asian Simul. Conf.*, Kyoto, Japan: Springer, 2018, pp. 45–59.
[81] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, 2011, pp. 627–635.
[82] W. Xu, K. Xu, H. Bastani, and O. Bastani, "Safely bridging offline and online reinforcement learning," 2021, *arXiv:2110.13060*.
[83] L. Roveda et al., "Model-based reinforcement learning variable impedance control for human-robot collaboration," *J. Intell. Robotic Syst.*, vol. 100, no. 2, pp. 417–433, 2020.
[84] K. Ciosek, Q. Vuong, R. Loftin, and K. Hofmann, "Better exploration with optimistic actor critic," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, Vancouver, Canada, 2019, pp. 1785–1796.
[85] R. Y. Chen, S. Sidor, P. Abbeel, and J. Schulman, "UCB exploration via Q-ensembles," 2017, *arXiv:1706.01502*.
[86] B. Settles, *Active Learning*. San Rafael, CA, USA: Morgan & Claypool, 2012.
[87] S. Ravi and H. Larochelle, "Meta-learning for batch mode active learning," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–6.

Liangliang Chen (Student Member, IEEE) received the BBA degree in business administration in 2017, the BS degree in automation in 2017, and the MEng degree in control engineering in 2019, all from the Harbin Institute of Technology, Harbin, China. He is currently working toward the PhD degree in electrical and computer engineering with the Georgia Institute of Technology, Atlanta, GA, USA. His current research interests include deep reinforcement learning and HVAC systems and control.

Fei Meng (Student Member, IEEE) received the BEng in electrical engineering and automation, and the MEng degree in control engineering from the Harbin Institute of Technology, Weihai and Harbin, China, in 2016 and 2019, respectively. He is currently working toward the PhD degree with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong SAR, China. His research interests include learning-based motion planning and control.

Ying Zhang (Senior Member, IEEE) received the MS degree in materials engineering from the University of Illinois at Chicago, in 2001, the MS degree in electrical engineering from the University of Massachusetts Lowell, in 2002, and the PhD degree in systems engineering from the University of California, Berkeley, in 2006. She is currently a professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology. Her research interests include the areas of sensors and smart wireless sensing systems, power management for energy harvesting wireless sensor networks, intelligent monitoring and diagnostic systems, artificial intelligence, information retrieval and data mining, and the Internet of Things.