Constrained Differentiable Cross-Entropy Method for Safe Model-based Reinforcement Learning

Sam Mottahedi*
The Pennsylvania State University
University Park, PA, USA
mpm5815@psu.edu

Gregory S. Pavlak*†
The Pennsylvania State University
University Park, PA, USA
gxp93@psu.edu

ABSTRACT

Reinforcement learning agents must explore their environments to learn optimal policies through trial and error. Due to challenges in simulating the complexities of the real world, there is a growing trend of training reinforcement learning (RL) agents directly in the real world instead of mostly or entirely in simulation. Safety concerns are paramount when training RL agents directly in the real world. This paper proposes MPC-CDCEM, a model-based reinforcement algorithm (RL) that allows the agent to safely interact with the environment and explore without additional assumptions on system dynamics. The algorithm uses a Model Predictive Control (MPC) framework with a differentiable cross-entropy optimizer, which induces a differentiable policy that considers the constraints while addressing the objective mismatch problem in model-based RL algorithms. We evaluate our algorithm in Safety Gym environments and on a practical building energy optimization problem. In addition, we showed that in both experiments, our algorithms have the lowest number of constraint violations and achieve comparable rewards compared to baseline constrained RL algorithms.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; **Control methods**; **Planning and scheduling**; **Machine learning algorithms**.

KEYWORDS

Reinforcement Learning, Constrained Markov Decision Process, Cross-Entropy Methods, Differentiable Convex Optimization, Limited Multi-label Classification

ACM Reference Format:
Sam Mottahedi and Gregory S. Pavlak. 2022. Constrained Differentiable Cross-Entropy Method for Safe Model-based Reinforcement Learning. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3563357.3564055

*Department of Architectural Engineering.
†Institutes of Energy and the Environment

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564055

1 INTRODUCTION

In recent years, reinforcement learning (RL) has shown exceptional success in various automated control and decision-making tasks. The RL algorithm can automatically learn a policy that satisfies the specified objective. However, current RL algorithms often require millions of interactions with the environment, which results in an expensive training process and primarily limits their application to simulated domains [42, 43]. In addition, transferring policies learned in a simulation environment has proven to be challenging due to model uncertainties [13, 47] and mismatch between real and simulated observations.

In many real-world applications, safety considerations also prevent the agent from freely exploring the environment. For example, a self-driving agent cannot take any actions that could cause harm to pedestrians while learning to optimize its driving policies. The agent needs to be constrained to specific actions that do not violate the safety requirements. In general, it is usually non-trivial to transform constrained optimal control problems into unconstrained problems [50].

One common approach to addressing this issue is to enforce some operational constraints on the outputs of a machine learning algorithm. However, usually, these constraints are not enforced during the training process and can potentially negatively impact the overall performance of the system [10].

Other approaches have sought to develop constrained and policy gradient-based safe RL algorithms, however, current methods cannot guarantee strict feasibility of policies even when initialized with feasible initial policies [30]. This limitation precludes their use in safety-critical environments.

A third approach has generated RL algorithms that transform the reward optimization criteria into a combination of reward and constraint violation cost, however, such methods suffer when the task objective and safety objectives contradict each other [50]. In addition, learning the dynamics of the environment and black box cost function typically is very difficult, especially in high dimensional space [38].

[38] contends that safe reinforcement learning focused on a single scalar reward inherently conflates task performance and safety requirements. Other safe reinforcement learning approaches based on ergodicity or the requirement to remain in safe states may be helpful in some applications, such as robotics, but are not relevant in other tasks since they can rule out both irreversible actions that result in optimal and safe decisions, in addition to unsafe actions.

The work in this paper is inspired by recent results applying the differentiable cross-entropy method (DCEM) [6], and we propose

a new safe reinforcement learning algorithm we name the Constrained Model Predictive Differentiable Cross-Entropy Method (MPC-CDCEM) that builds upon the success of DCEM. In each iteration, the algorithm samples from the distribution of policies and selects a set of trajectories with the best objective values that satisfy constraint values. If there are not enough feasible trajectories, the algorithm uses trajectories with the best constraint satisfaction performance. Compared to similar proposed solutions that use the traditional cross-entropy method [50], using the differentiable cross-entropy method enables an end-to-end learning process for both optimizing the objective and learning system dynamics. The differentiable policy class parametrized by the model-based components is a solution to the objective mismatch problem in model-based control [27], which arises when the objective being optimized is different from a target, often uncorrelated metric that we wish to optimize. In the context of model-based reinforcement learning, the model that achieves better performance in one-step ahead prediction of system dynamics is not necessary better for control. Another benefit of a differentiable policy is that it allows us to learn a low dimensional latent action space. Learning lower dimensional latent space of reasonable candidates enables the policy to leverage spatial and temporal structure in the solution space of optimal action sequence and ignore irrelevant action sequences.

The contributions of this paper are as follows. First, we present a model-based constrained RL algorithm in continuous state and action spaces. We formulate the problem under the Constrained Markov Decision Process [4] framework with minimal additional assumptions on system dynamics and constraint function. The differentiable cross-entropy method induces a differential control policy that addresses the objective mismatch problem in model-based control problems. We show that our approach can achieve state-of-the-art performance in terms of constraint violation number and accumulated expected return on Safety Control Gym [51]. We also explore a microgrid energy management system to reduce energy consumption while ensuring thermal comfort satisfaction for occupants and equipment safety by preventing excessive cycling in chillers.

2 RELATED WORK

Our approach relies on recent developments in differentiable cross-entropy methods and is thematically similar to several recent works [30, 50]. Here we discuss these topics and refer the interested reader to [22] for a more comprehensive review of safe RL topics.

[22] considers two main approaches to safe RL. The first is based on modifying the optimality criterion to introduce the concept of risk, and the second modifies the exploration process to avoid actions that can lead to undesirable or catastrophic situations. Regarding the first approach, optimization-based methods can be further categorized as worst-case criterion [36, 45], risk-sensitive criterion [7, 8, 25], constrained criterion [4, 26, 33], and other optimization criteria such as r-squared value-at-risk (Var) [31], or density of return [34].

Regarding the second approach, in general, there are two main ways of modifying the exploration process. Prior knowledge can be incorporated into the exploration process [1, 21, 40, 46], and risk measures can be added to determine the probability of selecting an action during the exploration process [23, 28].

In this work, we focus on the constrained Markov decision process formulation for safe RL [4]. [48] proposed a projection-based constrained policy gradient method that relies on projected gradients to ensure feasibility. [2] proposed a model-free constrained optimization method based on trust-region methods. However, these methods suffer from errors in gradient and Hessian matrix estimation, which may lead to underperformance [50]. [50] showed that safe reinforcement learning algorithms based on policy gradient methods cannot guarantee strict feasibility even when initialized with feasible initial policies and usually cannot find even a single feasible policy until their convergence. [3, 44] proposed Lagrangian methods that use adaptive penalty coefficients to ensure constraint feasibility, which requires target constraint violations to be set in advance.

Several papers proposed a safe RL algorithm that uses a model-based learning framework. Model-based approaches often produce more sample-efficient control solutions while ensuring constraint feasibility [19, 39]. [17] proposed a method that combines model-free control with a model-based safety check to ensure action feasibility. [11, 12] proposed a Lyapunov-based approach that provides an effective way to guarantee global safety during training via a set of local, linear constraints. [16] extended PILCO [18] model based algorithm to enable active exploration using a metric for out-of-sample Gaussian Process that supports conditional-value-at-risk constraints.

The cross-entropy method (CEM) is a zeroth-order optimizer, which works by generating a sequence of samples from the objective function [41]. In recent works, CEM has shown state-of-the-art performance for solving a control optimization problem with neural network transition dynamics [14, 24]. Recently, [6] proposed a method to approximate the derivative through an unconstrained, non-convex, and continuous optimization process. The differentiable cross-entropy method allows us to embed action sequences in a lower-dimensional space. It induces a differentiable control policy that solves the objective mismatch problem in model-based control. [30, 50] proposed constrained cross-entropy methods that only select elite trajectories that satisfy constraint satisfaction criteria.

In recent years, there has been an increasing interest in the application of reinforcement learning to building energy management systems and microgrids [9, 35, 49, 52]. [20] has developed a sample efficient model-based reinforcement learning algorithm for HVAC control system that can achieve high control performance. [29] developed a safe batched reinforcement learning algorithm with a Kullback-Leibler (KL) regularization for HVAC control.

In this paper, we propose a constrained differentiable cross-entropy method (CDCEM) that effectively solves large safety-critical optimization problems in lower-dimensional latent space. In contrast to previous safe model-based algorithms, our algorithm induces a differentiable policy that can address objective mismatch problem by using the gradient information from a policy function and fine-tune controller components such as transition model or the cost model. In addition, backpropagating across all sampled trajectories is memory intensive and intractable in most practical problems; hence, this is only possible in lower-dimensional embedded action space [6].

3 PRELIMINARIES

3.1 Constrained Markov Decision Process

A Markov decision process (MDP) is defined as $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \mu)$ where $\mathcal{S}$ is set of states, $\mathcal{A}$ is a set of actions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$ is the transition function and $\mu \in \mathcal{D}(\mathcal{S})$ is an initial state distribution. Let $\Pi : \mathcal{S} \to \mathcal{D}(\mathcal{S})$ be set of all stationary policies.

The objective of reinforcement learning is to select a policy that maximizes the discounted expected return

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=1}^{H} \gamma^t R(s_t, a_t, s_{t+1})]$$

where $\gamma \in [0, 1)$ is the discount factor. Given a finite horizon H, a H-step trajectory is sequence of H state-action pairs. τ represents a trajectory ($\tau = s_1, a_1, \ldots, s_H, a_H$) and $\tau \sim \pi$ is distribution over trajectories.

A constrained Markov decision process (CMDP) [4] is an MDP with constraints that restrict the set of allowable policies over that MDP. The set of cost functions $C_i : S \times A \times S \to \mathbb{R}$ mapping transition tuple to real valued cost and limits $d_1, \ldots, d_n$. The expected discounted return $J_{C_i}(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=1}^{H} \gamma^t C_i(s_t, a_t, s_{t+1})]$ with respect to cost function C_i. The set of feasible stationary policies for CMDP is then:

$$\Pi_C = \pi \in \Pi : \forall i, J_{C_i}(\pi) \leq d_i$$

and the solution to CMDP is:

$$\pi^* = \underset{\pi\in\Pi_C}{\operatorname{argmax}} J(\pi)$$

3.2 Differentiable Constrained Cross-Entropy Method

The cross-entropy method (CEM) [41] is a zeroth-order optimization approach in the form of $\hat{x} := \operatorname{argmin}_x f_\theta(x)$. CEM is an iterative solver which uses a sequence sampling distributions $g_\phi \in \mathbb{R}^n$. In each iteration N candidate points are sampled from the domain $[X_{t,i}]_{i=1}^{N} \sim g_{\phi_t}(\cdot)$, evaluated using $v_{t,i} := f_\theta(X_{t,i})$ and k elite candidates are selected to fit the new sampling distribution by solving the maximum-likelihood problem:

$$\phi_{t+1} := \underset{\phi}{\operatorname{argmax}} \sum_i \mathbb{1}\{v_{t,i} \leq \pi(v_t)_k\} \log g_\phi(X_{t,i}) \tag{1}$$

The top-k operation in Equation (1) makes the $\hat{x}$ non-differentiable with respect to θ. The top-k operation can be made differentiable using a Multi-Label Projection (LML) layer [5].

4 APPROACH

4.1 Problem Formulation

Here we use Model Predictive Control (MPC) for our model-based RL approach for controlling discrete-time dynamical systems with continuous action-space, which allows our agent to adapt its plan based on new observations.

Let $\mathcal{A}^H$ be the space of control sequences over controller horizon length H. The goal is to learn a latent action space $\mathcal{Z}$ with parameterized decoder $f_\theta^{dec} : \mathcal{Z} \to \mathcal{A}^H$. For a special case of Constrained Markov Decision Process we aim to repeatedly solve the following optimization problem:

$$\begin{aligned} \hat{z} := J_\theta(z; s_{init}) := \underset{z\in\mathcal{Z}}{\operatorname{argmax}} \sum_{t=1}^{H} J_\theta(a_t; s_t) \\ \text{subject to } s_1 = s_{init} \\ s_{t+1} = f^{trans}(s_t, a_t) \\ a_{1:H} = f_\theta^{dec}(z) \\ c(s_{t+1}) = 0 \end{aligned} \tag{2}$$

where s_{init} is the initial system state governed by deterministic system transition dynamics f^{trans} and $c(s_t)$ is a constraint violation cost function. The goal is to find a valid trajectory $s_{1:H}, a_{1:H}$ that optimizes the cost J_θ while adhering to the $c(s_t)$ constraint. In a receding horizon control setting [32] we only use the first action a_1 in the real system.

Here we adopt the model-based RL Probabilistic Ensembles with Trajectory Sampling (PETS) [14] that uses an ensemble of models with trajectory sampling (TS) to estimate the epistemic uncertainty of the input data. Using an ensemble of B neural networks parametrized with θ_b, we train the models by minimizing the mean squared error (MSE) of loss function $\mathcal{L}(\theta) = \mathbb{E}_{(s_t,a_t,s_{t+1}\in\mathcal{D}_b)} \| s_{t+1} - f_{\theta_b}(s_t, a_t) \|$. Algorithm 1 describe the training pipeline for our MPC controller. The constraint violation cost function $c(s_t)$, and the reward function $r(s_t)$ can be learned from data using any classification model, or using a known cost function.

Algorithm 1 Model-based MPC with CDCEM

Require: Initial collected trajectories $\mathcal{D}$, dynamics models, reward model, action sequence decoder, CDCEM parameters; Initialize dataset $\mathcal{D}$ with S random seed episodes;
while Not converged **do**
  for $t = 1, \ldots, T$ **do**
    $a_t \leftarrow$ CDCEM-solve($h_{s_{t-1}}$)
    $\{r_t, c_t, s_{t+1}\} \leftarrow env.step(a_t)$
    Add $\{r_t, s_t, a_t\}$ to $\mathcal{D}$
    if t mod update-interval = 0 **then**
      sample trajectories $\tau = [r_\tau, s_\tau, a_\tau]_{\tau=1}^{H} \sim \mathcal{D}$ from the dataset.
      Compute the loss: $\mathcal{L}(\tau, \hat{s}_\tau)$
      $\theta_{trans} \leftarrow$ grad-update($\nabla_\theta \mathcal{L}(\tau, \hat{s}_\tau)$)
      $\hat{z}_\tau \leftarrow \operatorname{argmax}_{z\in\mathcal{Z}} J_\theta(z; \hat{s}_\tau)$
      $\theta_{dec} \leftarrow$ grad-update($\nabla_\theta \sum_\tau J_\theta(\hat{z}_\tau)$)
    end if
  end for
end while

4.2 Constrained Differentiable Cross-Entropy Algorithm

In order to solve the constrained optimization problem in Equation (2) we use constrained differentiable cross-entropy method (CDCEM) described in Algorithm 2. Here we use Multi-Label Projection (LML) layer [5] described in Equation (3) which allows us to

implement differentiable top-k operation to select top trajectories based on the task cost function and feasibility cost.

$$\begin{aligned} \Pi_{\mathcal{L}_k(\frac{x}{\kappa})} := \operatorname{argmin} & - x^T y - \kappa U_b(y) \\ \text{subject to: } & 1^T y = k, \\ & 0 < y < 1, \end{aligned} \tag{3}$$

where U is binary cross-entropy function and κ is a hyperparameter that will induce vanilla CEM when $\kappa \to 0$. The derivative of Equation (3) can be computed by implicitly differentiating the Karush–Kuhn–Tucker (KKT) optimality conditions [5].

We can combine the two top-k operations with the weighted sum of reward and constraint cost for each using a linear opinion pool [15]. This provides a belief aggregation method that combines the decision based on cost and reward objective which in the simplest case involves taking the weighted linear average of opinions:

$$\mathcal{I}_{combined,j} = \alpha \mathcal{I}_{1,j} + (1-\alpha)\mathcal{I}_{2,j} \tag{4}$$

The α denotes the weight associated with the reward objective and $1-\alpha$ is the weight associated with the cost objective respectively. The weighting parameters α can be set as a hyperparameter or estimated during training.

5 EXPERIMENTS

5.1 Experiment 1: Point Goal Environment

5.1.1 Problem Description. First, we evaluate our proposed safe reinforcement learning algorithm in the OpenAI Safety Gym [38]. We use Safety Gym because (1) it uses an auxiliary cost function to enforce safety requirements, and (2) state-of-the-art reinforcement learning algorithms with benchmarked performance are available in all environments. The Point-Goal Task (in Figure 1) requires the robot to navigate to the designated green point with two actuators for thrust and angle while avoiding hazards and vase. The agent is penalized for moving to a hazard point or touching the movable vases. The reward for reaching the goal and the penalty for touching vases or moving to a hazard point are distinct.

The robot will receive a reward ($r_t = 1$) when it reaches the goal and a cost ($c_t(s_t) = 1$) when it violates the safety requirement. Here we use the available official baseline methods provided in the Safety Gym Environment, which are Constrained Policy Optimization (CPO) [2] as a constrained reinforcement learning baseline and cross-entropy based Model-Predictive Control (MPC-CEM) as a model-based unconstrained baseline. We follow the metrics proposed in the Safety Gym paper [38] which are episodic reward and episodic cost, and the number of samples required to reach convergence as a proxy for sample efficiency.

5.1.2 Implementation Details. We use the same hyperparameters provided in the Safety Gym official benchmark for the CPO and the same hyperparameters for both model-based (MPC-CDCEM and MPC-CEM). We evaluate each algorithm with three different seeds. For the dynamics models, we use a neural network with three hidden layers with 64 neurons, ReLU activation, 512 batch size, and the Adam optimizer with $1e^{-1}$ learning rate. We train the model for 50 epochs. We use a smaller neural network with two hidden layers and 128 neurons, ReLU activation, and Adam optimizer with a learning rate of $1e^{-3}$ to predict the constraint violation given. For MPC-CDCEM, we use a neural network as a decoder to map embedded action from the latent planning horizon to a larger planning horizon. For the decoder, we use a neural network with two hidden layers and 256 neurons, Swish [37] activation, and the Adam optimizer with $1e^{-4}$ learning rate. The fused cost function parameter α is set to 0.5.

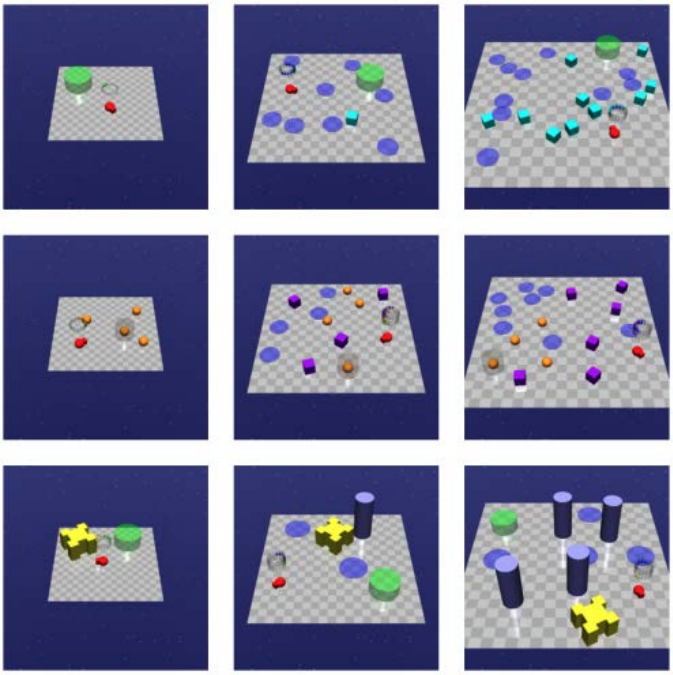


Figure 1: Point-Goal: Safety Gym environments for experiment 1.

5.1.3 Results. Figure 2 and Figure 3 show the accumulated reward and episodic cost violation which is defined as the total cost violation during an episode for the safety-gym point-goal task. The CPO algorithm's learning curve and violation cost are shown with a horizontal line since the model-free algorithm requires an order of magnitude more interaction (50 times) with the environment. It can be seen that our proposed algorithm converges to a slightly lower reward, but receives a significantly lower violation cost.

Table 1 compares the number of constraint violations during the first 5×10^3 iterations. The average over the three seed cases is reported along with the standard deviation (STD). It can be seen that the number of violations is significantly higher in the model-free case. Compared to the two model-based approaches, Table 1 demonstrates that the MPC-CDCEM incurs a lower cost while exploring the environment safely.

We further evaluate the effect of hyperparameter α in the MPC-CDCEM fused cost function on constraint violation. Figure 4 shows that constraint violation decreases sharply when α increases from 0 to 0.4. While a further increase in α reduces the number of violations, it negatively influences the obtained reward.

Algorithm 2 Constrained DCEM (CDCEM) $(r, c, g_\phi; \kappa, N, k, M)$

for $j = 1$ to M **do**
  $[X_{j,i}]_{i=1}^{N} \sim g_{\phi_j}(\cdot)$ ▷ Sample N points from the domain. Differentiate with reparameterization.
  $v_{j,i}^{reward} = r(X_{j,i})$ ▷ Evaluate the reward objective function at those points.
  $v_{j,i}^{safety} = c(X_{j,i})$ ▷ Evaluate the constraint objective function at those point.
  $\mathcal{I}_{1,j} = \Pi_{\mathcal{L}_k}(\frac{v_{j,i}^{reward}}{\kappa})$ ▷ Compute the soft top-k projection for reward objective.
  $\mathcal{I}_{2,j} = \Pi_{\mathcal{L}_k}(\frac{v_{j,i}^{safety}}{\kappa})$ ▷ Compute the soft top-k projection for constraint feasibility.
  $\mathcal{I}_{combined,j} = \alpha\mathcal{I}_{1,j} + (1-\alpha)\mathcal{I}_{2,j}$ ▷ Compute the soft top-k combining prediction $\mathcal{I}_{1,j}$ and $\mathcal{I}_{2,j}$.
  Update ϕ_{j+1} by solving the problem in Equation (1).
end for
Return: $\mathbb{E}[g_{\phi_{M+1}}(\cdot)]$

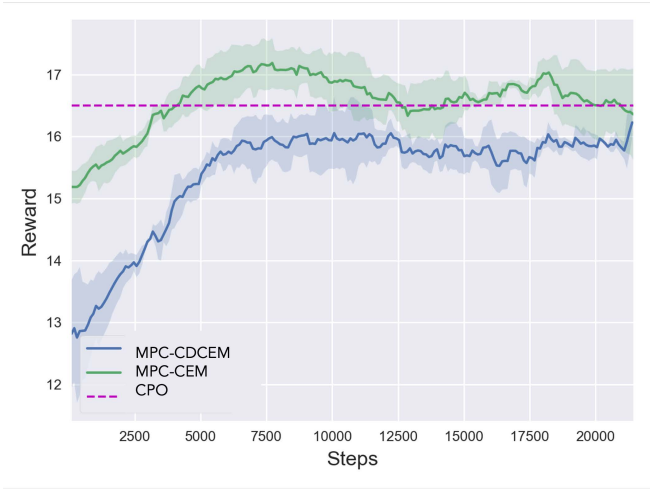


Figure 2: SafePoint-Goal task learning curves (α = 0.5)

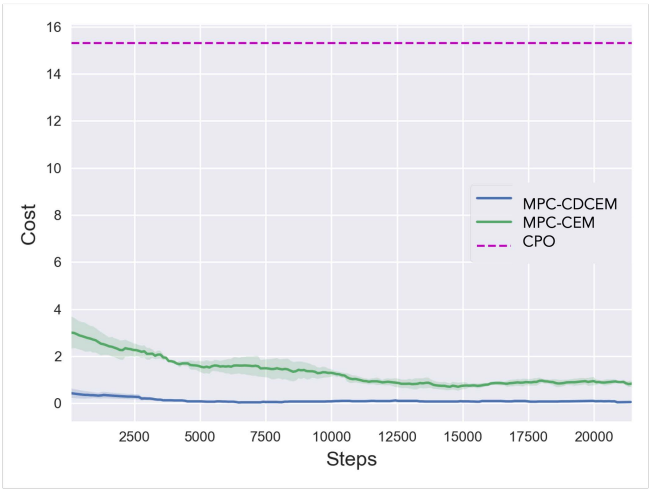


Figure 3: Constraint violation cost (α = 0.5)

5.2 Experiment 2: Microgrid Energy Management System

5.2.1 Problem Description. One of the most critical constraints for utilizing building thermal mass and energy flexibility in a microgrid is maintaining a satisfactory occupant comfort level while minimizing energy consumption. A common approach to ensure policy feasibility is to penalize the violations of thermal comfort, but this does not guarantee the occupant's comfort requirements. In addition, it is also necessary to ensure the control strategies do not violate physical operating constraints of the equipment or cause premature equipment degradation. For example it is often desirable to slowly ramp large fan motors to avoid large pressure fluctiations in the duct systems, and it is also often desired to limit the cycling of large equipment like chillers.

Table 1: Constraint violations in 5000 iteration (α = 0.5)

Algorithm	Constraint Violations	STD
MPC-CEM	56.21	3.52
MPC-CDCEM	29.75	1.21
CPO	812.4	12.2

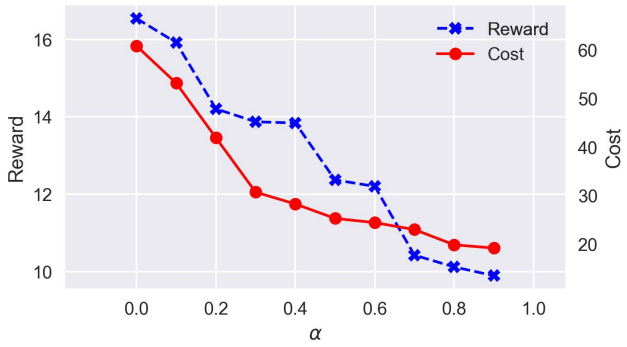


Figure 4: α in fused cost function

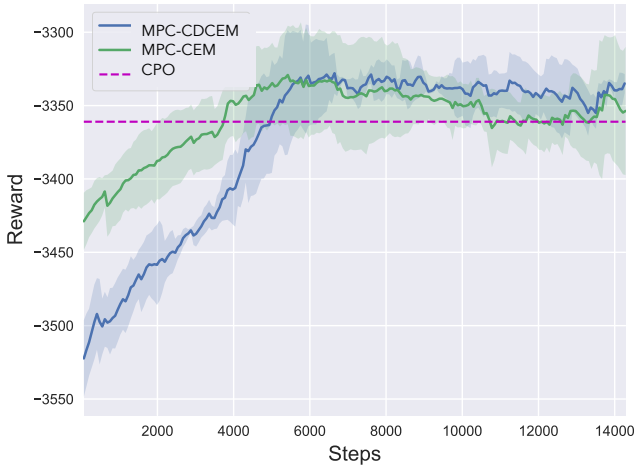
Here we evaluate our safe reinforcement learning algorithm on a building-level microgrid energy management test-bed. The simulation environment is implemented using the EnergyPlus model for a large office building, PV system, wind turbine, inverters, and a battery storage facility connected to the main grid. The additional electricity can be bought from the grid if the renewable energy and battery storage cannot meet the demand. The building model used here is a large commercial office building with a Chicago weather file. Cooling is provided to the building zones by chilled-water variable-air-volume (VAV) air-handlers and cooling-only terminal boxes. Zone heating is performed by electric resistance baseboard heaters. The central plant features two centrifugal chillers, two cooling towers, and water pumps. In this experiment, we control the zone cooling set-points to maintain the zone temperature to ensure occupants' comfort while minimizing the electricity consumption in the microgrid. The constraints here are to maintain a satisfactory comfort level as measured by the zone Predictive Mean Vote (PMV) index during the occupied hours and prevent excessive switching of large chilled water plant equipment. PMV index values range from −3 to +3, which describes the feeling from cold to hot, respectively. Based on ASHRAE 55 the agent will violate comfort constraints if PMV is outside the recommended limits (−0.5 and 0.5). The objective is described in Eq. (5).

$$\begin{aligned} \max J &= -\lambda_1 E_{hvac,t} - \lambda_2 \|u_t\|_1 \\ \text{s.t. } &|PMV_t| \leq 0.5, \quad \forall t \in \{\text{occupied hours}\} \\ &\text{chiller-cycles} \leq 2\text{per day} \end{aligned} \tag{5}$$

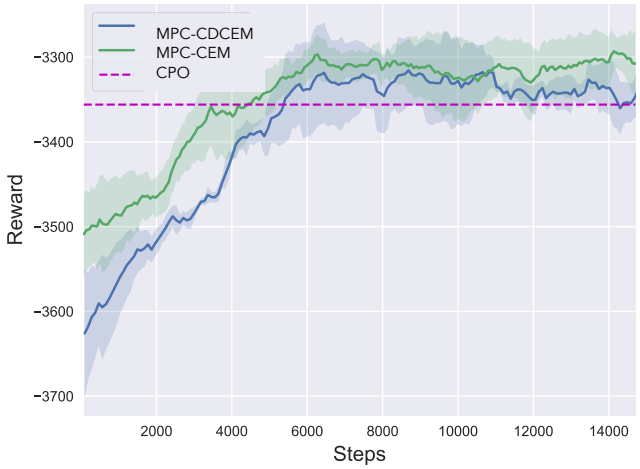
5.2.2 Implementation Details. We use the same hyperparameter for both baselines and MPC-CDCEM over the weather file from May 2018 to September 2018. The EnergyPlus model uses a 10-min control time step with a planning horizon of $H = 24$. The same hyperparameters are used for the dynamics model, constraint cost prediction model, and the decoder neural network, as mentioned in experiment 1. The decoder maps the latent horizon length of $H_l = 4$ to the task horizon $H = 24$. We train the induced MPC policy by iterating over-collected samples for 20 epochs with a batch size of 256. We follow the metrics proposed in the Safety Gym paper [38] which are episodic reward and episodic cost, and the number of samples required to reach convergence as a proxy for sample efficiency. We use the same hyperparameters provided in the Safety Gym official benchmark for the CPO and the same hyperparameters for both model-based MPC (MPC-CDCEM and MPC-CEM).The fused cost function parameter α is set to 0.5 in the thermal comfort experiment and 0.25 for for comfort and chiller cost top-k operation and 0.5 for the task objective.

5.2.3 Results. Figure 5 and 6 show the learning curve and reward and constraint violation for the occupancy comfort and chiller cycle constraint experiments. The figures show MPC-CDCEM quickly learns the underlying constraint function and converges to a reward that is inline or slightly lower than MPC-CEM and CPO. This observation is reasonable since the agent can ignore the constraint and maximize the task reward.

After training the agent, the trained agent directly operated in a simulation environment with the Washington D.C. weather sequence on the first week of June. Table 2 and Table 3 highlight the HVAC electric use, reward, and constraint violation during



(a) Thermal comfort experiment

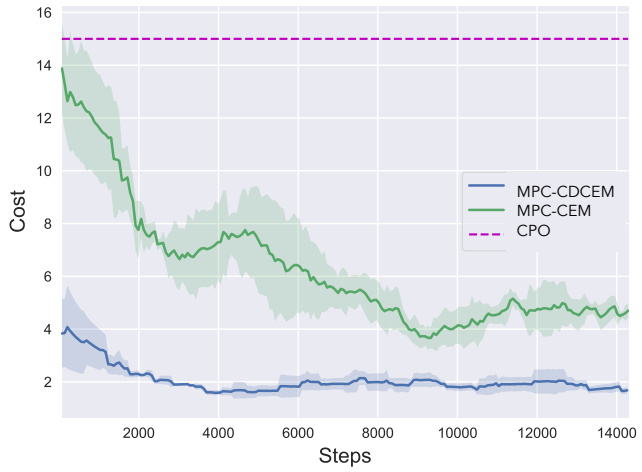


(b) Chiller cycle experiment

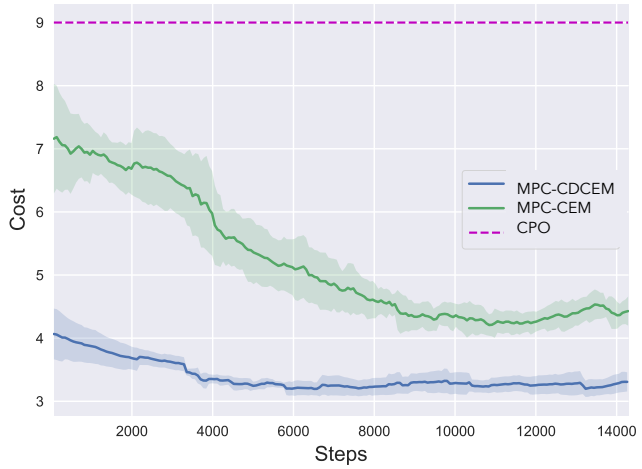
Figure 5: Learning curves for thermal comfort and chiller cycling constraint experiments.

a weekday in the testing period. The comfort and chiller cycle violations are defined as the number of timesteps with $|PMV| > 0.5$ and the number of chiller cycles during a weekday. Compared to other methods, the reward and constraint violations show that MPC-CDCEM achieves rewards comparable to MPC-CEM and CPO while constraint violations are always less than other methods. The proposed MPC-CDCEM agent (average over three random seeds) saves 12.3% energy compared to the default nighttime setup (NSU) and consumes approximately 1% more compared to MPC-CEM.

Figures 7 and 8 show the simulation results for a weekday during the testing period in order to compare the NSU, MPC-CDCEM, MPC-CEM, and CPO indoor thermal comfort and zone temperatures. The zone temperature represents the weighted average zone temperature based on zone area, and the PMV is the occupancy



(a) Thermal comfort experiment (α = 0.5)



(b) Chiller cycle experiment (α_{comf}, α_{chill} = 0.25, α_{rew} = 0.5)

Figure 6: Cost trend for thermal comfort and chiller cycling constraint experiments.

Table 2: Testing period result for thermal comfort experiment (α = 0.5.

Case	Energy Use [kWh]	Reward	Constraint Violation
NSU	31,320	-4,186	0
MPC-CEM	27,151	-3,811	3
MPC-CDCEM	27,429	-3,804	0
CPO	27,109	-3,834	4

weighted average PMV. It can be seen that MPC-CDCEM is trying to maintain the zone temperature at a temperature that does not violate the comfort constraint. MPC-CEM and CPO are more unstable than MPC-CDCEM, which resulted in violations of the

Table 3: Testing period result for chiller cycling experiment.

Case	Energy Use [kWh]	Reward	Constraint Violation: Chiller	Constraint Violation: Comfort
NSU	31,320	-4,186	6	0
MPC-CEM	27,285	-3,912	8	0
MPC-CDCEM	27,394	-3,888	4	0
CPO	27,147	-3,907	7	2

comfort constraints in the morning and afternoon. In the chiller cycle experiment, the MPC-CDCEM agent learns to maintain a lower temperature in the morning, possibly preventing excessive switching of chillers during the day. This confirms that the proposed algorithm observes the constraints during policy learning.

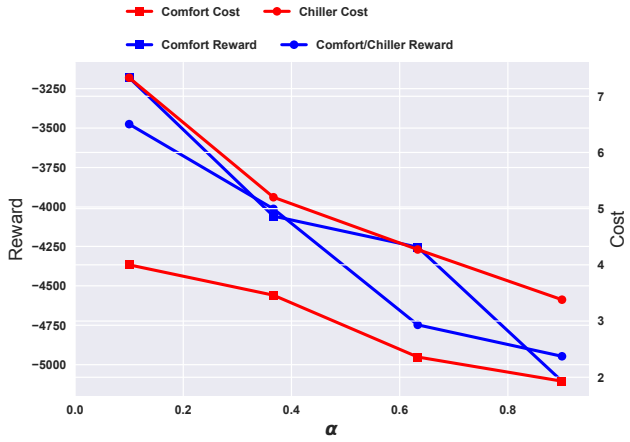


Figure 9: α in fused cost function.

We further evaluated the effect of fused cost function parameter α as show in Figure 9, which shows that a good balance between reward and cost constraint tends result in good compromise between cost and safety.

6 CONCLUSION

In this study, we present an effective constrained RL algorithm formulated under the Constrained Markov Decision Process framework with no additional assumption on the system dynamics. The proposed algorithm induces a differentiable control policy that addresses the objective mismatch problem and enables an end-to-end learning process while enforcing constraint feasibility. First, we evaluated our algorithm in the Safety Gym environment, which showed superior constraint satisfaction while maintaining task performance compared to other constrained RL algorithms. Next, we evaluated MPC-CDCEM in a microgrid environment to minimize energy consumption while ensuring occupants' thermal comfort and preventing excessive chiller cycles. In both cases, MPC-CDCEM achieved better constraint satisfaction while maintaining good reward performance compared to other baseline algorithms.

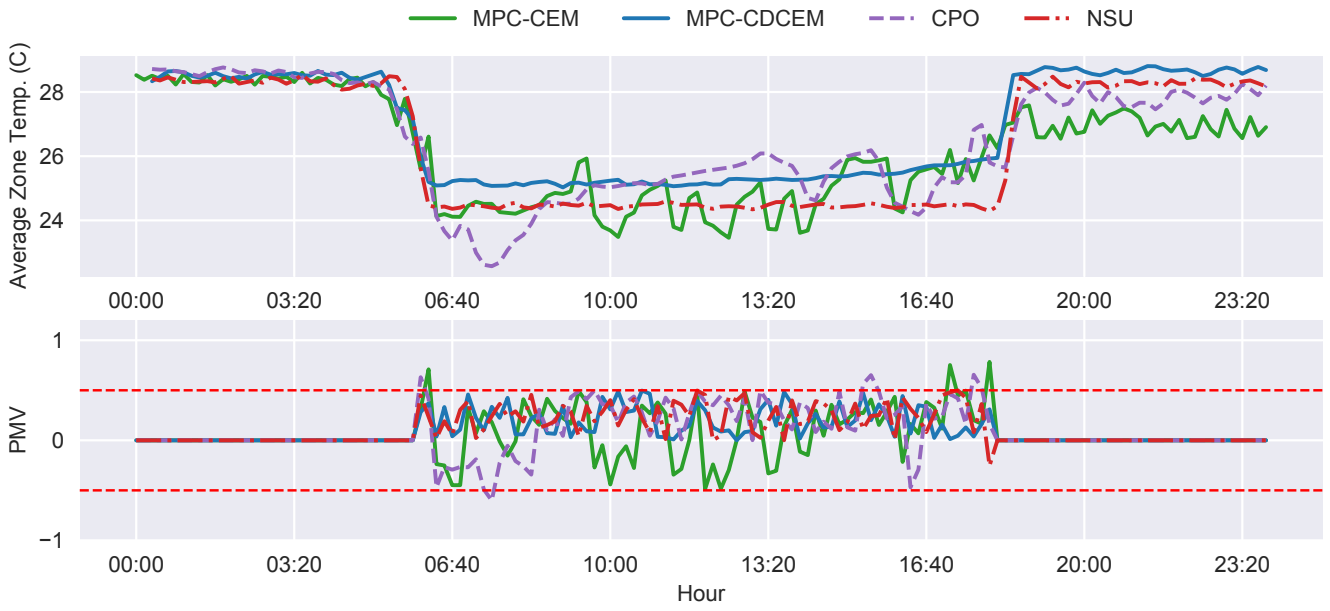


Figure 7: Performance evaluation for thermal comfort experiment.

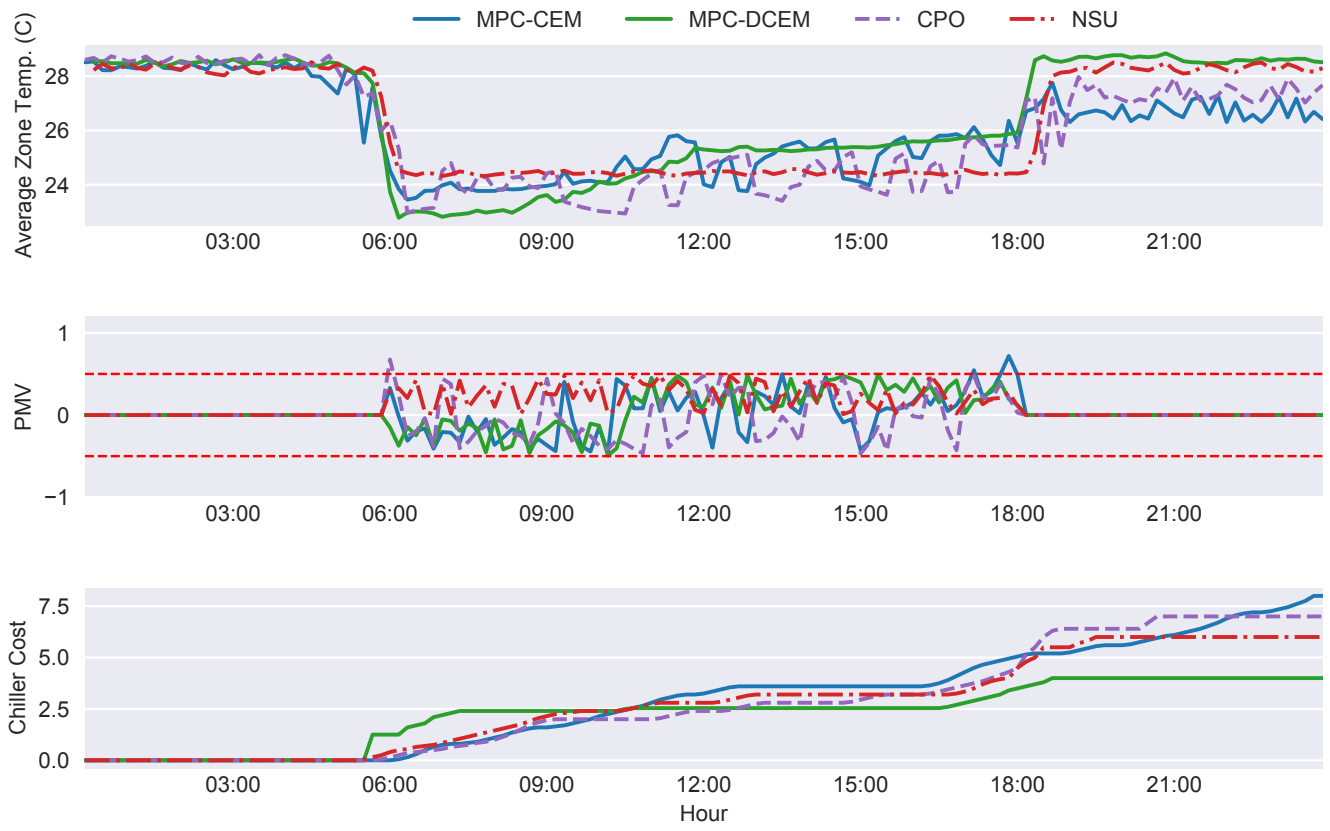


Figure 8: Performance evaluation for chiller cycle experiment.

REFERENCES

[1] Pieter Abbeel, Adam Coates, and Andrew Y Ng. 2010. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research* 29, 13 (2010), 1608–1639.

[2] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In *International Conference on Machine Learning*. PMLR, 22–31.

[3] Eitan Altman. 1998. Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical methods of operations research* 48, 3 (1998), 387–417.

[4] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC Press.

[5] Brandon Amos, Vladlen Koltun, and J Zico Kolter. 2019. The limited multi-label projection layer. *arXiv preprint arXiv:1906.08707* (2019).

[6] Brandon Amos and Denis Yarats. 2020. The differentiable cross-entropy method. In *International Conference on Machine Learning*. PMLR, 291–302.

[7] Arnab Basu, Tirthankar Bhattacharyya, and Vivek S Borkar. 2008. A learning algorithm for risk-sensitive cost. *Mathematics of operations research* 33, 4 (2008), 880–898.

[8] Vivek S Borkar. 2001. A sensitivity formula for risk-sensitive cost and the actor–critic algorithm. *Systems & Control Letters* 44, 5 (2001), 339–346.

[9] Bingqing Chen, Zicheng Cai, and Mario Bergés. 2019. Gnu-rl: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 316–325.

[10] Bingqing Chen, Priya Donti, Kyri Baker, J Zico Kolter, and Mario Berges. 2021. Enforcing Policy Feasibility Constraints through Differentiable Projection for Energy Optimization. *arXiv preprint arXiv:2105.08881* (2021).

[11] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2018. A lyapunov-based approach to safe reinforcement learning. *arXiv preprint arXiv:1805.07708* (2018).

[12] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2019. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031* (2019).

[13] Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. 2016. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518* (2016).

[14] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *arXiv preprint arXiv:1805.12114* (2018).

[15] Roger Cooke et al. 1991. *Experts in uncertainty: opinion and subjective probability in science*. Oxford University Press on Demand.

[16] Alexander I Cowen-Rivers, Daniel Palenicek, Vincent Moens, Mohammed Abdullah, Aivar Sootla, Jun Wang, and Haitham Ammar. 2020. Samba: Safe model-based & active reinforcement learning. *arXiv preprint arXiv:2006.09436* (2020).

[17] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. 2018. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757* (2018).

[18] Marc Deisenroth and Carl E Rasmussen. 2011. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*. Citeseer, 465–472.

[19] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. 2013. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence* 37, 2 (2013), 408–423.

[20] Xianzhong Ding, Wan Du, and Alberto E Cerpa. 2020. Mb2c: Model-based deep reinforcement learning for multi-zone building control. In *Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 50–59.

[21] Kurt Driessens and Sašo Džeroski. 2004. Integrating guidance into relational reinforcement learning. *Machine Learning* 57, 3 (2004), 271–304.

[22] Javier Garcıa and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1 (2015), 1437–1480.

[23] Clement Gehring and Doina Precup. 2013. Smart exploration in reinforcement learning using absolute temporal difference errors. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. 1037–1044.

[24] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*. PMLR, 2555–2565.

[25] Ronald A Howard and James E Matheson. 1972. Risk-sensitive Markov decision processes. *Management science* 18, 7 (1972), 356–369.

[26] Yoshinobu Kadota, Masami Kurano, and Masami Yasuda. 2006. Discounted Markov decision processes with utility constraints. *Computers & Mathematics with Applications* 51, 2 (2006), 279–284.

[27] Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. 2020. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523* (2020).

[28] Edith LM Law. 2005. Risk-directed exploration in reinforcement learning. (2005).

[29] Hsin-Yu Liu, Bharathan Balaji, Sicun Gao, Rajesh Gupta, and Dezhi Hong. 2022. Safe HVAC Control via Batch Reinforcement Learning. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 181–192.

[30] Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao. 2020. Constrained Model-based Reinforcement Learning with Robust Cross-Entropy Method. *arXiv preprint arXiv:2010.07968* (2020).

[31] Helmut Mausser. 1998. Beyond VaR: From measuring risk to managing risk. *ALGO research quarterly* 1, 2 (1998), 5–20.

[32] David Q Mayne and Hannah Michalska. 1988. Receding horizon control of nonlinear systems. In *Proceedings of the 27th IEEE Conference on Decision and Control*. IEEE, 464–465.

[33] Teodor Mihai Moldovan and Pieter Abbeel. 2012. Risk Aversion in Markov Decision Processes via Near Optimal Chernoff Bounds.. In *NIPS*. 3140–3148.

[34] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. 2010. Nonparametric return distribution approximation for reinforcement learning. In *ICML*.

[35] Adam Nagy, Hussain Kazmi, Farah Cheaib, and Johan Driesen. 2018. Deep reinforcement learning for optimal control of space heating. *arXiv preprint arXiv:1805.03777* (2018).

[36] Arnab Nilim and Laurent El Ghaoui. 2005. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research* 53, 5 (2005), 780–798.

[37] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).

[38] Alex Ray, Joshua Achiam, and Dario Amodei. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708* 7 (2019).

[39] Benjamin Recht. 2019. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), 253–279.

[40] Michael T Rosenstein, Andrew G Barto, Jennie Si, Andy Barto, Warren Powell, and Donald Wunsch. 2004. Supervised actor-critic reinforcement learning. *Learning and Approximate Dynamic Programming: Scaling Up to the Real World* (2004), 359–380.

[41] Reuven Y Rubinstein and Dirk P Kroese. 2013. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media.

[42] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[43] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* (2017).

[44] Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*. PMLR, 9133–9143.

[45] Aviv Tamar, Huan Xu, and Shie Mannor. 2013. Scaling up robust MDPs by reinforcement learning. *arXiv preprint arXiv:1306.6189* (2013).

[46] Jie Tang, Arjun Singh, Nimbus Goehausen, and Pieter Abbeel. 2010. Parameterized maneuver learning for autonomous helicopter flight. In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 1142–1148.

[47] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 23–30.

[48] Eiji Uchibe and Kenji Doya. 2007. Constrained reinforcement learning from intrinsic and extrinsic rewards. In *2007 IEEE 6th International Conference on Development and Learning*. IEEE, 163–168.

[49] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *Proceedings of the 54th annual design automation conference 2017*. 1–6.

[50] Min Wen and Ufuk Topcu. 2020. Constrained cross-entropy method for safe reinforcement learning. *IEEE Trans. Automat. Control* (2020).

[51] Zhaocong Yuan, Adam W Hall, Siqi Zhou, Lukas Brunke, Melissa Greeff, Jacopo Panerati, and Angela P Schoellig. 2021. safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning. *arXiv preprint arXiv:2109.06325* (2021).

[52] Zhiang Zhang and Khee Poh Lam. 2018. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments*. 148–157.

Data-driven Decarbonization of Residential Heating Systems

John Wamburu, Noman Bashir, David Irwin and Prashant Shenoy
University of Massachusetts Amherst
{jwamburu,nbashir,irwin,shenoy}@umass.edu

ABSTRACT

Heating buildings using fossil fuels such as natural gas, propane and oil makes up a significant proportion of the aggregate carbon emissions every year. Because of this, there is a strong interest in decarbonizing residential heating systems using new technologies such as electric heat pumps. In this paper, we conduct a data-driven optimization study to analyze the potential of replacing gas heating with electric heat pumps to reduce CO_2 emission in a city-wide distribution grid. We conduct an in-depth analysis of gas consumption in the city and the resulting carbon emissions. We then present a flexible multi-objective optimization (MOO) framework that optimizes carbon emission reduction while also maximizing other aspects of the energy transition such as carbon-efficiency, and minimizing energy inefficiency in buildings. Our results show that replacing gas with electric heat pumps has the potential to cut carbon emissions by up to 81%. We also show that optimizing for other aspects such as carbon-efficiency and energy inefficiency introduces tradeoffs with carbon emission reduction that must be considered during transition. Lastly, we present preliminary results that shed light into the expected load exerted on the electric grid by transitioning gas to electric heat pumps.

CCS CONCEPTS

• **Mathematics of computing → Linear programming**.

KEYWORDS

Decarbonization, Optimization, Electric Heat Pumps

ACM Reference Format:
John Wamburu, Noman Bashir, David Irwin and Prashant Shenoy. 2022. Data-driven Decarbonization of Residential Heating Systems. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3563357.3564058

1 INTRODUCTION

Residential energy usage contributes nearly 20% of all greenhouse gas emissions in the United States [12]. In 2019 alone, buildings contributed over 1850 million metric tons of greenhouse gases [1]. Heating and cooling account for roughly 38% of these emissions [23]. To avert the disastrous effects of climate change, the energy system has begun a major transition towards a carbon-free future. The building sector will play a major role in this transition.

To date, a significant fraction of buildings in colder climates, such as regions of North America and Europe, depend on natural gas, propane, or oil for residential heating in the winter. For example, 82% of Massachusetts households use non-electric sources of energy — such as utility gas, heating oil, or propane — for heating [7]. On the other hand, only 16% of households use electric heating. The low adoption of electric heating is attributed to the historical inefficiency of electric heat pumps in extremely cold climates. However, recent technological advancements have made it possible to operate electric heat pumps efficiently even at very low temperatures of -15°C [31]. This has made modern heat pumps viable candidates for replacing fossil fuel based heating even in the extreme climates of North America or Northern Europe.

Electric heat pumps offer two key decarbonization advantages over fossil fuel based heating, such as utility gas. First, they are more energy-efficient, which means they use less energy than gas furnaces to generate the same amount of heat energy [26]. Second, their reliance on electricity means that as the electric grid transitions towards greener and renewable sources for energy production, the carbon footprint of electric heat pumps will also decrease. In contrast, the carbon footprint of fossil fuel based heating will remain constant as the energy efficiency of gas furnaces is reaching its limits [11]. As a result, replacing gas furnaces with energy-efficient electric heat pumps has great potential to not only reduce a building's energy usage, but also reduce its overall carbon footprint.

A push for transition to electric heat pumps can come from either the utility or the end consumers. Although consumers do not have a direct incentive to reduce their carbon emissions, they do have a strong financial incentive to reduce their energy consumption, and ultimately their utility bills. However, the capital cost of such interventions is often a major bottleneck. To incentivize the switch, transition strategies are typically accompanied by significant rebates and cost savings. Often, subsidies provide assistance to make a building more energy efficient as a whole. For instance, in some states in the U.S., heat pump rebates can be as high as $\$10,000$, and are accompanied with an additional 75-100% rebate for adding new insulation to the building envelope [24]. Despite these subsidies, consumers are still expected to make a major upfront investment, which presents a financial hurdle for many customers. Utilities, on their own, also do not have any financial incentives for decarbonization, which can require upgrades in the electric grid, as well as retiring parts of the gas network infrastructure before its natural end of life. However, they are increasingly being required by government policy or regulations to reduce carbon emissions in line with commitments made at the UN's Paris Climate Agreement to limit the global temperature rise to less than 2°C [3].

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564058

Any transition strategy, be it from a utility's perspective or a consumer's, requires identifying a set of buildings to be retrofitted with heat pumps. The selection of buildings is non-trivial and traditionally depends on various factors, such as the total energy consumption and insulation levels. However, for decarbonization, two of the most important factors are the total carbon footprint and the carbon-efficiency of a building. The total carbon footprint quantifies the total amount of emissions from heating, irrespective of how much heat was generated and the efficiency of the process. Carbon efficiency, on the other hand, quantifies the amount of heat generated per unit of carbon emitted. The two metrics are related but distinct. For example, a building may have a large total carbon footprint, but be highly carbon-efficient. Therefore, while a carbon reduction strategy that targets the highest emitting buildings yields the greatest *initial* reduction in CO_2 emissions, it does not fully exploit additional opportunities for improvements, such as increasing building energy efficiency. In addition, there are additional questions that need to be answered. How does the choice of one metric impact the other? How does carbon-efficiency differ from energy-efficiency from a decarbonization standpoint? How does carbon emission reduction impact energy consumption (also a proxy for heating cost) for the end consumers? Finally, since the transition is not instantaneous but rather a gradual process, which buildings should be transitioned first, and which should come later? The answers to these questions are non-trivial and require an in-depth analysis of real energy consumption data.

In this paper, we conduct a data-driven optimization study to analyze the potential carbon emission reductions from replacing gas-based heating with electric heat pumps in a city-wide distribution grid. Our empirical study is based on analyzing real natural gas and electric data from 13,800 and 6,445 smart electric and gas meters respectively collected over a one year period. We conduct an in-depth analysis of the heating demand of buildings and quantify their carbon footprint. We quantify CO_2 emission reductions obtained when a carbon-optimal transition strategy is applied to the conversion from gas to electric heating. We then introduce additional goals such as CO_2 efficiency and improving building efficiency to take advantage of further energy improvements in addition to CO_2 reduction. In conducting our empirical analysis, this paper makes the following contributions.

Energy Consumption and Emission Analysis. We use a city-scale dataset to conduct an in-depth analysis of its gas consumption and the resulting CO_2 emissions. One of our key findings is that the median building produces $\approx$ 32 MT of CO_2 annually, with some buildings emitting $\geq$250MT CO_2, which is 7.8× the median. This analysis motivates transition strategies that target buildings with higher emissions to meet aggressive decarbonization goals.

Optimal CO_2 Reduction. We present a multi-objective optimization (MOO) framework that enables the flexible selection of a subset of homes for heat pump retrofits to achieve decarbonization goals. Our analysis of a transition and building selection strategy that achieves maximum possible initial CO_2 reductions suggests that it fails to take advantage of other aspects of energy transition such as improving energy and carbon efficiency in buildings. Consequently, we update our multi-objective optimization framework to consider additional objectives of energy efficiency and carbon efficiency.

Joint CO_2 and energy-efficiency optimization. In addition to a carbon emissions analysis, we analyze the energy inefficiency of buildings and its causes. We show energy efficiency can be improved by transitioning buildings from gas to electric heat pumps, to reduce emissions, while simultaneously improving energy efficiency via renovations, such as adding insulation to the building. We show the effect of prioritizing energy efficiency on energy demand and CO_2 emissions. Our analysis finds that older buildings are generally less efficient and should be prioritized in transition.

2 BACKGROUND

In this section, we present background on the energy transition, decarbonization of heating, and electric heat pumps.

2.1 Energy Transition

The U.S., along with most countries in the world, still relies on non-renewable, fossil fuel-based energy sources — such as coal, natural gas — for a majority of its energy needs. For example, fossil fuel-based energy resources fulfilled more than 79% of U.S. energy consumption in 2021 [1]. To curtail the effects of climate change, there is a push towards cleaner sources of energy. The energy transition can be achieved individually for each of the major sectors of energy consumption, such as transportation, buildings, and agriculture. However, prior studies have suggested that a more effective pathway is to transition our energy needs to electricity while intensifying efforts to clean the sources of electricity production [21]. This hypothesis is supported by recent estimates that suggest that the carbon intensity of electricity (in g·CO_2/kWh) in the U.S. decreased 30% between 2001 and 2017, largely due to the replacement of coal-fired power plants with natural gas and wind generation [30]. This trend is expected to continue as the use of renewable energy resources for electricity production increases. The electrification of buildings and transportation has received significant attention to accelerate the energy transitioning progress. In this paper, we quantify the impact of electrifying heating in the building sector via electric heat pumps on energy consumption and CO_2 emissions, an important step for energy transition.

2.2 Decarbonizing Heating

Heating using fossil fuels, such as natural gas, propane and oil, accounts for more than 47% of overall heating energy consumption in United States [10]. Natural gas and propane furnaces use a gas burner to heat air or water, which is then circulated to heat the building. The combustion of natural gas produces carbon dioxide as a byproduct, which is released into the atmosphere. Heating and cooling in the residential sector is responsible for more than 38% of all CO_2 emissions in United States every year [23]. The decarbonization of heating is an important step towards achieving overall carbon reduction goals. The decarbonization of heating, to various degrees, can be achieved in multiple ways by transitioning to geothermal heating, hybrid heating, and/or electric heat pumps. Heating through geothermal energy is an emerging technology, but may not be suitable for all locations [33]. Hybrid heat pumps combine electric heating with a secondary fuel, such as a propane tank. While these options may be cost-effective solutions in the short term, they are not a long-term solution if society is to transition to a carbon-free future. The use of energy-efficient electric heat

Table 1: Summary of Key Data Characteristics

No. of electric meters	13,800
No. of gas meters	6,445
Granularity of gas data	1 hour
Granularity of electric data	5 minutes
Duration	Jan 2020 - Dec 2020

pumps has been proposed as an ideal pathway to decarbonization of the future grid [5, 6, 16, 19, 34]. As the electric grid transitions towards a carbon-free future, discussed in Section 2.1, the heating sector will need to organically transition to a carbon-free future.

2.3 Electric Heat Pumps

Electric heat pumps are a new and energy-efficient alternative to gas furnace heating during cold seasons, as well as space cooling during summer seasons. During winter seasons, heat pumps pull warm air from outside and concentrate it into your home space, making the inside warm. Conversely, during summer seasons, a heat pump moves heat from within a building to the outside atmosphere which cools the inside of the building. Since the main principle behind heat pump operation is heat transfer instead of heat generation, heat pumps are more energy efficient than fossil fuel based burners.

The most popular type of heat pump available in the market today is an air-source heat pump [26], which transfers heat between the inside of a building and the outside air. Because these heat pumps rely on air heat transfer, as the outside temperature decreases, their heating capacity degrades. In the past, such heat pumps required a backup energy source to be used during extremely low temperatures, such as a gas furnace or electric heating [14]. However, recent advances in heat pump technology have made them efficient even at low temperatures, which makes them an ideal replacement for gas heating even in cold climates [9, 31]. In addition to increased energy efficiency, heat pumps also have other advantages over natural gas. Since they use electricity, as more electricity is sourced from renewable sources, their carbon footprint is lower than that of natural gas. Moreover, due to their reduced energy usage, heat pumps can reduce the cost of heating a building by up to 60%. This makes them an attractive source of heating from a carbon, energy efficiency, and cost perspective.

3 PROBLEM AND METHODOLOGY

In this section, we present the problem statement and key research questions we address in the paper. We also describe the datasets and experimental methodology we use to answer these questions.

3.1 Problem Statement

Given a set of residential buildings in a city or town, the primary goal of our work is to quantify the impact of replacing gas heating with electric heat pumps on carbon emission reductions, and the optimal order in which homes should be transitioned. Another goal is to understand the impact of introducing additional goals such as carbon-efficiency and energy inefficiency in buildings as priorities for such a transition, and the tradeoffs such goals have on emissions reduction. Specifically, we seek to answer the following questions.

(1) What is the distribution of heating energy consumption, and how much gas is required to meet these heating requirements? What are the daily and seasonal variations in gas consumption? How much CO_2 is emitted from this gas consumption, both for individual residential buildings and in the aggregate?
(2) What is the impact of replacing gas heating with electric heat pumps on energy consumption and CO_2 emissions? What is the optimal order in which buildings should be transitioned from gas to electric heat pumps in order to minimize CO_2 emission?
(3) How is this ordering impacted when additional goals such as carbon/energy inefficiency of buildings are introduced? How is CO_2 reduction impacted, and what are the tradeoffs?

3.2 Description of Datasets

The answers to these questions vary based on region and largely depend on seasonal factors such as the severity of winter weather, which in turn influences gas demand for heating. Other factors such as type and purpose of building e.g. industries, factories may also affect energy patterns. In this paper, we focus on residential data collected from a small city in the Northern region of United States. Since the gas and electric system design in this city is typical of many regions across the world, and residential usage is invariant across regions, we believe that our insights are widely applicable.

Gas and Electric Usage Data. Our dataset consists of electric and gas consumption data recorded by 13,800 electric and 6,445 gas meters. The data also includes a mapping of electric and gas meters installed at each building. To compute the aggregate load profile of a building, we sum up the load from the electric and gas meters installed in the building. Electricity demand data is recorded at 5 minute granularity and spans >5 years. Gas consumption data is recorded at hourly granularity, and spans the same duration. For the purpose of our study, we limit our analysis to the full calendar year 2020, which is the latest year whose complete data was available. Table 1 shows a summary of the characteristics for this dataset.

Building Property Data. In addition to load data, we collect property data for all buildings present in our dataset using public real-estate records. This includes the size of the building, type of building, e.g., single vs multi-family, etc. We use this data to augment out analysis, e.g., to generate a building's energy profile, we normalize the load by the building's size to enable comparative analysis across different buildings. We gather and parse this data from publicly available property information recorded as part of tax records.

Weather Data. Since our analysis involves measuring the impact of weather on energy usage, we gather weather data for the city from the Dark Sky API [1]. We collect multiple data points such as temperature, humidity from the API. We gather this data at hourly granularity to match our hourly gas load data.

4 ENERGY USAGE AND CARBON ANALYSIS

To understand the impact of transitioning buildings from gas to electric heat pump heating, we begin with an analysis of the current load on the gas system and the resulting CO_2 emission. Specifically, we study the daily, seasonal and annual variations in gas energy usage across the whole system.

[1] https://darksky.net/dev

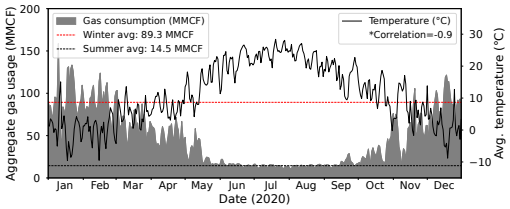


Figure 1: Aggregate gas demand during the year 2020.

4.1 Energy Demand Analysis

Figure 1 depicts the aggregate gas demand for the city under-consideration over the course of an year. There are two peak periods — between Jan-Feb and Nov-Dec months. These peaks coincide with the most severe winter months. The average daily gas demand during winter months is 89.3 MMCF, which is 6× the daily average during summer months (14.5 MMCF). The data also demonstrates a strong negative correlation (-0.9) between temperature and gas demand — as the temperature falls, gas demand rises due to increased residential heating in buildings. Figure 2(a) depicts the daily aggregate demand for gas across the system. The figure shows that on most days, aggregate demand is < 25 MMCF. This is mainly due to the use of non-heating appliance such as stoves. The figure also shows a spread of high usage days during which demand is highest. For instance, the peak day consumes > 150 MMCF, which is 3.5× the average usage. Since these high usage days are predominantly made up of heating consumption, replacing gas heating with electric heat pumps has great potential to curtail CO_2 emission.

In addition to analyzing the aggregate daily demand, we study the variation in gas demand by time of day, and the periods during which the daily peak demand occurs. Figure 2(b) depicts the average gas demand by time of day during winter and summer months. During winter, gas demand exhibits a bi-modal peak — a sharp peak between 8-9am, and a moderate peak between 5-8pm. This coincides with the morning and evening routines during which occupancy and activity in homes is highest. The peak hourly demand is 5.08 MMCF, while the average demand is 3.72, indicating a 1.4 peak-to-average ratio. Lastly, gas demand during summer months does not show significant variation over the course of the day. This is because gas usage during summer is predominantly made up of appliance usage which is fairly constant throughout the year.

4.2 Carbon Emission Analysis

The combustion of natural gas produces carbon dioxide as a byproduct which is released into the atmosphere. When gas is used for heating, the amount of CO_2 emitted is driven by the amount of gas required to generate enough heat for a building. This is in turn driven by the temperature e.g. as the temperature decreases, more heat is required to raise the indoor temperature, as well as the building size i.e. larger spaces require more energy to heat. Further, building characteristics such as insulation affect how much gas is consumed e.g. buildings with poor insulation lose heat to the atmosphere faster than those with better building envelope, and therefore have higher gas demand.

To examine the CO_2 emission generated directly from gas heating, we compute the emission for each building by multiplying the total gas consumption for the year with the emission factor of gas. About 0.0551 MT of CO_2 is produced for each MCF of natural gas burned [2]. To estimate heating gas consumption, we subtract summer average from overall gas usage. Figure 2(c) depicts the distribution of CO_2 emitted by each building from heating gas combustion. The figure shows that the median building emits 32.4 MT of CO_2 every year. The figure also shows a long tail, with a small number of buildings emitting a lot more CO_2 compared to others. These buildings in particular form good candidates for transition to in order to reduce CO_2 emission from the heating. The highest emitting buildings contribute >250 MT CO_2 during the year which is 8.1× the median emission and 7× the average CO_2 emission.

5 MULTI-OBJECTIVE DECARBONIZATION

In this section, we present a data-driven multi-objective optimization (MOO) framework that enables flexible selection of a subset of homes for heat pump retrofits to achieve decarbonization goals. We start the optimization with an initial goal of maximizing CO_2 reductions and iteratively add additional objectives of maximizing carbon efficiency and targeting energy inefficient buildings. In doing so, our formulation enhances decarbonization to not only aim for the highest emitters, but also target smaller buildings that have a smaller CO_2 footprint but are CO_2 or energy inefficient, with the aim of achieving a balanced transition. While the optimization can be extended to other objectives, in this work, we focus on CO_2 reduction, CO_2 efficiency and energy efficiency.

5.1 Optimizing for Carbon Emissions Reduction

Let $H = \{h_1, h_2 ... h_n\}$ denote the set of buildings, each indexed by i. Let C_i^g denote the total CO_2 emission from the cumulative gas consumption for building i required for heating during the year. Let C_i^e denote the total CO_2 emission from the cumulative electric consumption for building i required for heating when using an electric heat pump. Let α_i represent the transition-to-electricity status for the building i and S represent the denote the target number of buildings to transition to electric heat pump heating.

Given that, our objective is to select S buildings from the set H which when transitioned to electric heat pumps result in the lowest aggregate CO_2 emission possible across buildings. This objective can formally be described as follows.

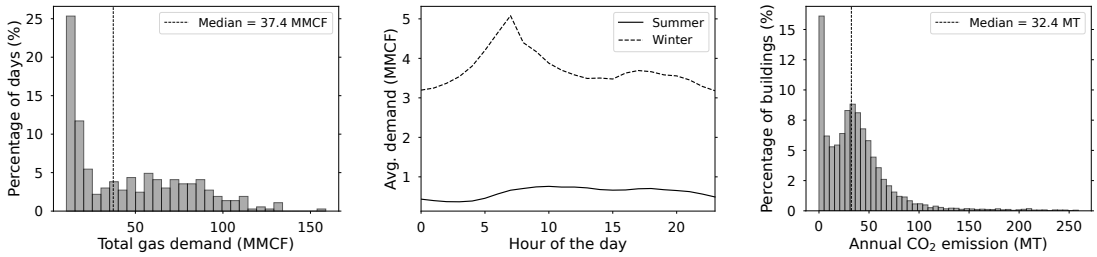
$$\begin{aligned} \min \quad & \sum_{i=1}^{n} (1-\alpha_i) \cdot C_i^g + \alpha_i \cdot C_i^e \\ \text{s.t.,} \quad & \text{Equations (2) - (4)} \\ \text{vars.,} \quad & C_i^g, C_i^e, \alpha_i, S \quad \forall i \end{aligned} \tag{1}$$

Our first constraint relates to to the level of transition. Let α_i denote a binary variable which indicates the state of transition for each building i such that $\alpha_i \in \{0, 1\}$. When set, the building is transitioned to electric heat pump heating, and when not set, the building remains on gas. Further, let S denote the target number of buildings to transition to electric heat pump heating. To ensure that only S buildings are transitioned, the sum of all values of α_i must equal S, as stated below.

$$\sum_{i=1}^{n} \alpha_i = S \tag{2}$$

Our final set of constraints simply ensure that a building cannot have negative carbon emissions from either the gas consumption or the electric demand.

$$C_i^g \geq 0 \quad \forall i \tag{3}$$



(a) Aggregate Daily Gas Demand (b) Avg. Gas Demand by Hour (c) CO_2 Emissions

Figure 2: Probability distributions for aggregate daily gas demand (a), average gas demand over the course of the day (b), and CO_2 emissions from buildings throughout the year.

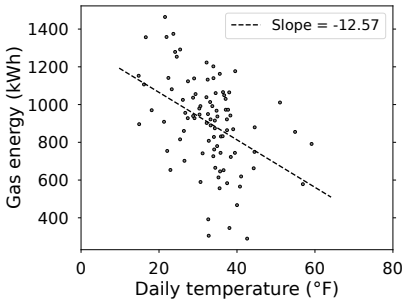


Figure 3: Relationship between the energy generated from gas consumption and temperature.

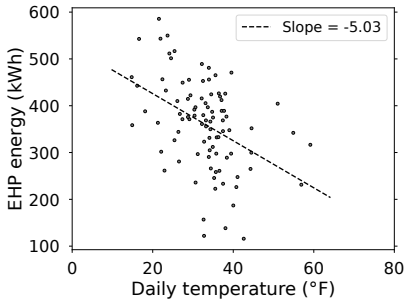


Figure 4: Relationship between energy demand of an electric heat pump and the temperature.

$$C_i^e \geq 0 \quad \forall i \tag{4}$$

The total CO_2 emission from gas consumption, C_i^g, over the course of an year for a building i is a multiple of the total heating gas demand D_i^g and the carbon intensity of gas I_g.

$$C_i^g = D_i^g \times I_g \tag{5}$$

The total CO_2 emission from electric demand, C_i^e, over the course of an year for a building i is a multiple of the total electricity demand D_i^e and the carbon intensity of the electric grid I_e.

$$C_i^e = D_i^e \times I_e \tag{6}$$

It should be noted that a simple ordering of homes based on their total carbon emissions can achieve the singular goal of selecting a set of buildings that maximizes carbon emission reductions after transition. However, we present this as a flexible multi-objective optimization framework so that additional objectives, discussed in subsequent sections, can be integrated into the same framework.

5.2 Optimizing for Carbon-efficiency

Optimizing for total carbon emission reduction targets buildings with highest carbon footprint. However, the large footprint may be a result of large residential area or large number of residents and the building itself may be making a highly efficient use of its gas demand. With CO_2 reduction as the sole goal, only larger buildings will be selected, and many smaller highly inefficient buildings will be left out. To capture this effect, we define the notion of *carbon-efficiency*. We define *carbon-efficiency* as the amount of CO_2 emitted when one unit area of a building is raised by one unit of temperature. We further elaborate the notion of *carbon-efficiency* next.

The notion of *carbon-efficiency* is based on the observation that electric heat pumps consume lower energy compared to gas to heat the same building from a lower temperature T_{low} to a higher temperature T_{high}. Figure 3 depicts the relationship between energy generated from gas consumption and temperature for a building during winter months. The figure shows an inverse relationship between energy and temperature. As the temperature decreases, the energy required to heat the building increases. The rate of change (captured in the slope of the fit line) indicates the amount of energy required to raise the building's temperature by one unit of temperature. This is directly proportional to the CO_2 produced for each unit temperature. To measure how well the fit line fits the energy and temperature data, we compute coefficient of determination, R^2 (0.19). Since we consider a coarse granularity of energy and temperature data i.e. average across a whole day, the R^2 value is adequate. We do this for both energy and gas usage data.

Figure 4 depicts the relationship between the energy demand of an electric heat pump and the temperature for the same building. Similar to gas energy consumption, there is an inverse relationship between electric energy and temperature. However, the slope is significantly less steep than that of gas. This is because electric heat pumps consume lower electrical energy to generate the same amount of heat energy. Since carbon emissions are directly proportional to energy consumption, the CO_2 produced for each unit temperature is lower for electric heat pumps. Since buildings have different sizes, energy consumption alone is not enough to compare usage between buildings of varying size. Before computing the energy slope, we normalize both gas and electric heat pump energy by size. We then compute carbon emissions per unit size.

Note that maximizing carbon efficiency introduces a tradeoff between reduction and efficiency. Since the most carbon efficient buildings are not necessarily the highest emitters in absolute scale, a portion of CO_2 reduction must be foregone to maximize efficiency. However, since gas furnaces are inherently inefficient, maximizing

carbon efficiency places a tighter bound on wasted CO_2 emission, and leads to better energy utilization. We extend the multi-objective optimization framework defined in Equation 1 to jointly maximize the *carbon-efficiency* and minimize the total carbon emissions. In doing so, we introduce a new set of variables that are defined next.

First of all, we use the absolute value of the carbon emission slopes for both gas and electricity as a substitute for *carbon-efficiency*. This formulation of the problem allows us to keep the overall objective as minimization of carbon emission reductions and slopes of the emission curves (representing carbon-efficiency). Given that, let λ_i^g be the absolute slope of gas CO_2 emissions for the building i. Let λ_i^e be the absolute slope of electric CO_2 emission for the building i. Our joint optimization of minimizing carbon emissions and maximizing carbon-efficiency can be stated as follows.

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} (1-\alpha_i)\cdot C_i^g + \alpha_i \cdot C_i^e \\ \min \quad & \left[\sum_{i=1}^{n} (1-\alpha_i)\cdot \lambda_i^g + \alpha_i \cdot \lambda_i^e\right] \cdot \frac{1}{n} \\ \text{s.t.,} \quad & \text{Equations (2) - (4)} \\ \text{vars.,} \quad & C_i^g, C_i^e, \alpha_i, \lambda_i^g, \lambda_i^e, S \quad \forall i \end{aligned} \tag{7}$$

As stated before, to maximize CO_2 efficiency, we minimize the average absolute slope of CO_2 emissions curve across all buildings.

5.3 Targeting Energy Inefficient Buildings

In addition to *carbon-efficiency*, building decarbonization strategies may also want to target energy inefficient buildings. Energy efficiency in transition is important for two main reasons. First, higher efficiency translates to a lower carbon footprint. Second, since nearly half of a building's energy usage results from heating and cooling alone, improving efficiency of heating is one of the most effective ways for reducing a building's energy bill.

The sources of energy inefficiencies include poor insulation, high temperature set points for heating and cooling, and inefficient appliances. In this section, we extend our optimization formulation to target buildings that have one or more energy inefficiencies. To do so, we extend our analysis to not only consider gas energy usage only, but also electric usage. We learn a building energy model and use it to identify energy inefficiencies which we target in decarbonization.

Let $U = \{h_1, h_2 ... h_p\}$ denote the set of buildings with heating inefficiency i.e. high heating slope, each indexed by k. Let $V = \{h_1, h_2 ... h_q\}$ denote the set of buildings with cooling inefficiency i.e. high cooling slope, each indexed by l. Further, let $W = \{h_1, h_2 ... h_r\}$ denote the set of all other buildings i.e. all buildings with any other inefficiency except heating and cooling, as well as those without any inefficiency, each indexed by m.

Let $C_k^{g,u}$, $C_l^{g,v}$ and $C_m^{g,w}$ be the total carbon emissions from gas consumption in heating inefficient, cooling inefficient, and the remaining buildings, respectively. Further, let $C_k^{e,u}$, $C_l^{e,v}$ and $C_m^{e,w}$ be the total carbon emissions from electricity usage in heating inefficient, cooling inefficient and remaining buildings, respectively. Let ζ_k, β_l and γ_m be the binary variables that indicate the transition status of heating inefficient, cooling inefficient, the remaining buildings, respectively. All of the binary variables can only take a value of either 0 or 1, which means that $\zeta_k, \beta_l, \gamma_m \in \{0, 1\}$ for all k, l, and m. To transition only S buildings, the sum of all set variables from all building groups must be equal to S.

$$\sum_{k=1}^{p} \zeta_k + \sum_{l=1}^{q} \beta_l + \sum_{m=1}^{r} \gamma_m = S \tag{8}$$

Lastly, since buildings cannot have negative energy usage and therefore negative emission, we ensure that emission from buildings in all groups is greater than or equal to zero.

$$C_k^{g,u}, C_l^{g,u}, C_m^{g,v}, C_k^{e,v}, C_l^{e,v}, C_m^{e,w} \geq 0 \quad \forall k, l, m \tag{9}$$

With these constraints in place, our objective is to select S buildings from the sets U, V and W such that when the S buildings are transitioned to electric heat pumps, carbon emissions are minimized, while the portion of S buildings selected from the heating and cooling inefficient groups is maximized. This multi-objective optimization problem can be formally stated as follows.

$$\begin{aligned} \min \quad & f_u(u) + f_v(v) + f_w(w) \\ \min \quad & \sum_{k=1}^{p} (-1 \cdot \zeta_k) + \sum_{l=1}^{q} (-1 \cdot \beta_l) \\ \text{s.t.,} \quad & \text{Equations (8) - (9)} \\ \text{vars.,} \quad & C_k^{g,u}, C_l^{g,v}, C_m^{g,w}, C_k^{e,u}, C_l^{e,v}, C_m^{e,w}, \zeta_k, \beta_l, \gamma_m, S \quad \forall k, l, m \end{aligned} \tag{10}$$

The composite functions f_u, f_v, and f_w are defined as follows.

$$f_v(u) = \sum_{k=1}^{p} (1-\zeta_k)\cdot C_k^{g,u} + \zeta_k \cdot C_k^{e,u} \tag{11}$$

$$f_u(v) = \sum_{l=1}^{q} (1-\beta_l)\cdot C_l^{g,v} + \beta_l \cdot C_l^{e,v} \tag{12}$$

$$f_w(w) = \sum_{m=1}^{r} (1-\gamma_m)\cdot C_m^{g,w} + \gamma_m \cdot C_m^{e,w} \tag{13}$$

Note that to maximize the number of buildings selected from the heating and cooling inefficient groups, we minimize the negation of all set binary variables from the two sets.

6 EVALUATION

In this section, we present the results for various decarbonization strategies presented in Section 5 and evaluate their efficacy in reducing carbon emissions and increasing energy efficiency. To do so, we introduce varying levels of transition across the system — where the transition rate represents the percentage of buildings converted from gas to electric heat pumps.

6.1 Experimental Setup

The gas and electricity consumption data from the buildings (described in Section 3.2) provides building-level metering of the gas and electricity demand. We first disaggregate gas and electric demand data into two components: first, used for heating purposes, and second, used by all the other appliances such as stoves. To do so, we compute the average gas usage during the summer, and subtract it from the year-round data to get the heating component of gas usage. This removes usage from other appliances such as stoves, and ensures we estimate CO_2 reduction from heating only. We also account for energy loss due to the inherent inefficiency of gas furnaces. To do so, we use an efficiency level of 87.5%, which lies between the typical efficiency of a standard and a high efficiency

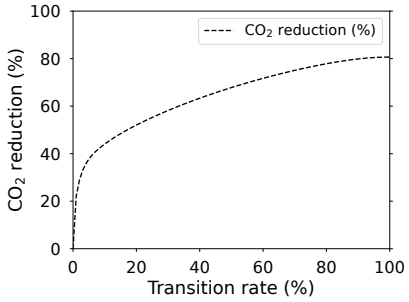


Figure 5: Reduction in carbon emission at varying levels of transition to electric heat pumps.

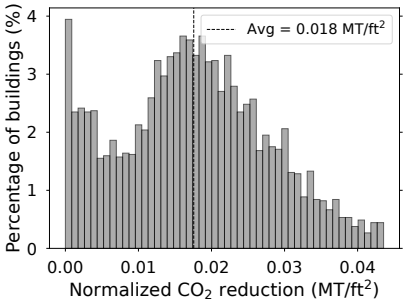


Figure 6: Distribution of normalized carbon emissions reduction across all buildings.

furnace. To compute total carbon emissions from gas usage, we use a gas carbon intensity value of 0.0551 MT/MCF [2].

To compute the corresponding electric heat pump emissions, we compute the total heat energy generated by the volume of gas consumed and use the Heating Seasonal Performance Factor (HSPF) of electric heat pumps to compute the electric energy required to generate the equivalent amount of heat energy. For all experiments, we assume a HSPF rating of 8.5, which is typical of many efficient heat pump models. Finally, to compute the carbon emissions from electric heat pump usage, we use a carbon intensity value of 0.000386 MT CO_2/kWh, corresponding to the average carbon intensity value for the United States electric grid [30].

Finally, to perform evaluation, our experimental setup makes a few assumptions. First, we use a coarse grained emission factor of electricity, as well as efficiency levels of gas furnaces. Second, to learn building energy models, we assume that high granularity energy and temperature data are available. While availability of the former varies from region to region and depends on the pervasiveness of smart meters, the latter is widely available.

6.2 Optimizing Carbon Emissions Reduction

We first analyze the impact on carbon emissions after transitioning buildings from gas to electric heat pumps using a strategy that optimizes carbon emission reductions. At each transition level, the buildings that lead to the highest reduction in carbon emission are selected for transition. We run this optimization on our gas consumption data and compute the resulting carbon emission at each transition level. Figure 5 presents the results for this analysis and there are two interesting observations. First, carbon emissions reduce at an exponential rate at the lower levels of transition i.e. 1-10%. This is because since we are only optimizing for carbon emissions reduction, the biggest emitters are selected first, which leads to a disproportionately high carbon emissions reduction at the start. As transition rate increases, carbon emission reductions enter another phase characterized by a linear growth (with low slope) from 10-100%, where the rest of buildings with moderate emissions are transitioned. Second, results also shows that at 100% transition, electric heat pumps have the potential to cut carbon emissions by up to 81%. This is a noteworthy observation, and demonstrates the viability of heat pumps to replace natural gas for heating and at the same time, helping make significant strides towards decarbonizing the building sector and achieving climate goals.

We next analyze the carbon emission reductions per unit area. We compute the total carbon emission reductions for each building, and normalize the difference with the size of the building. Figure 6 depicts the distribution of carbon emission reductions per unit area across all buildings. The figure shows that normalized CO_2 reduction is normally distributed with the average building seeing an annual reduction of 0.018 MT/ft^2. Given that the median house size of single family home in United States is 2273ft^2 [8], each home has a potential to reduce 40.9 MT each year.

6.3 Maximizing Carbon-efficiency

Next, we analyze the impact of optimizing for carbon-efficiency on carbon emissions reduction. The goal here is to quantify the tradeoff between carbon emissions reduction and efficiency, i.e. how much carbon emissions can be eliminated while also ensuring that carbon emissions per unit area is minimized. We solve the optimization problem described in Section 5.2 and compare the aggregate carbon emissions after the transition with the carbon-optimal strategy results presented in Section 6.2. Figure 7 depicts the results for this analysis. The figure shows that carbon emissions reduction is lower than the optimal case for up to $\approx$ 85% transition, after which carbon emissions are similar to the optimal scenario. The magnitude of initial growth of CO_2 reduction is also lower. This is because some of the highest emitting buildings have high carbon efficiency. This indicates a tradeoff between absolute reduction and efficiency i.e. in joint optimization, some large emitters are foregone in favor of less-efficient buildings which have a lower absolute carbon footprint. The largest deviation occurs at 15% transition, where 71 GT of carbon emissions reduction is foregone in favor of maximizing efficiency. However, carbon-efficiency increases by 1.9×. Utility companies can therefore choose between efficiency and absolute reduction depending on the weight associated with each outcome. Since there is not a significant deviation in carbon emissions reduction, utility companies can maximize carbon-efficiency while sacrificing only a small amount of carbon emissions reduction compared to the optimal case.

6.4 Targeting Energy Inefficient Buildings

We next examine the tradeoff in carbon emissions reduction introduced by prioritizing inefficiencies in buildings. We begin by performing building segmentation based on their unique energy inefficiencies and the underlying faults that cause such inefficiencies. Our fault analysis is based on the technique proposed in [17]. We apply the proposed technique to our data. Table 2 shows the indicator characteristics identified for each building along with the possible faults that underlay such inefficiencies. The third column shows the optimization group that we place each building in based

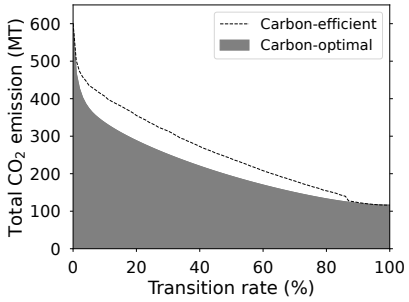


Figure 7: Carbon emissions at varying levels of electric heat pump transition while jointly optimizing for carbon emissions reduction and carbon-efficiency.

Table 2: Probable building faults alongside their underlying characteristics.

Indicator Characteristics	Probable Faults	Optimization Group
High heating slope	Inefficient heater, Poor building envelope	Inefficient heating
High cooling slope	Inefficient HVAC, Poor building envelope	Inefficient cooling
High heating temperature	High set point, Poor building envelope	Other
Low cooling temperature	Low set point, Poor building envelope	Other
High base load	Inefficient appliances	Other

on the identified characteristics. Specifically, we target homes that have heating and cooling inefficiencies since these would benefit most from transitioning from gas to electric heat pumps.

Figure 8 depicts the distribution of energy inefficiencies identified in buildings in our dataset. The figure shows that poor building envelope is the leading cause of energy inefficiency. This is true across buildings of all age groups. It also reveals that inefficient HVAC and heating units are the second and third most prevalent causes of energy inefficiencies of buildings, respectively. Since electric heat pumps are capable of operating as both heating and cooling units based on the season, this distribution of faults underpins the importance of targeting energy inefficient buildings in transition. The figure also shows that older buildings are more prone to being energy inefficient, while newer buildings show less prevalence probably due to improved building standards. This segmentation of buildings based on underlying energy inefficiency informed the basis of our targeted optimization, presented in Section 5.3. Targeting inefficient buildings offers multiple advantages over optimizing for carbon emissions alone. For example, transitioning to electric heat pumps typically comes with additional benefits such as building retrofits. This enables buildings to take advantage of these additional benefits during transition. Moreover, the amortized cost of transition may be reduced by performing multiple upgrades at once.

To quantify the tradeoff in carbon emissions reduction and targeting inefficiency buildings, we run the optimization described in Section 5.3 on our datasets and compare the resulting carbon emissions reduction with the carbon optimal scenario. Figure 9 depicts the results of this experiment, and presents some interesting observations. First, carbon emissions reduction show a gradual linear decrease from start to finish compared to the optimal case, and only converges at near full transition (≈ 98%). Since older buildings are more prone to energy inefficiency, this figure also indicates that

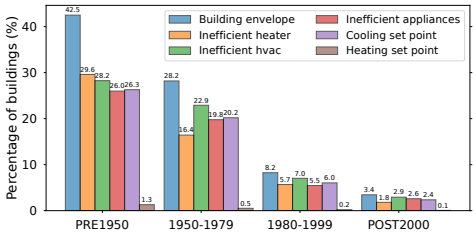


Figure 8: Distribution of energy inefficiencies across buildings by age group.

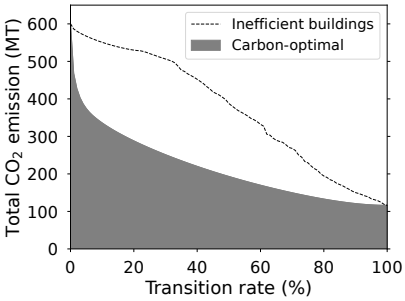


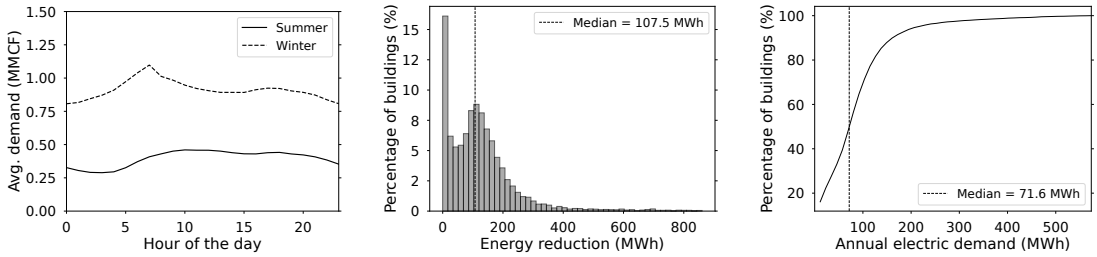
Figure 9: Carbon emissions at varying levels of electric heat pump transition while jointly optimizing for carbon emissions reduction and targeting energy inefficient buildings.

this strategy has the effect of selecting older buildings first. Similar to optimizing for carbon efficiency, targeting inefficient buildings introduces a tradeoff between absolute reduction and improving energy efficiency. We find that the highest emitters are not necessarily the most energy inefficient. Since the end goal in both cases is CO_2 reduction, utility companies can choose to forego one for the other depending on the weight associated with each outcome.

Finally, we evaluate the impact of transitioning to electric heat pumps on the daily gas demand. Figure 10(a) depicts the average hourly demand of gas during winter and summer months after 100% transition to electric heat pumps. Similar to the observations made in Figure 2(b), we find that gas demand exhibits a bi-modal peak — between 8-9am and 5-8pm. The figure also makes two interesting observations. First, the average peak demand reduces by 78% compared to the case before transition. Second, the extremity of the peak is also reduced significantly. Before transition, the peak demand was 1.4× the average hourly demand. Post transition, the peak-to-average ratio is 1.2, indicating a 14.3% reduction compared to the value before transition. Lastly, the figure shows no significant change in daily usage pattern of gas during summer months since consumption is mainly made up of appliance usage which does not change with the introduction of heat pumps.

6.5 Impact on Energy Consumption Reduction

Figure 10(b) depicts the distribution of potential energy reduction for buildings in our dataset. It shows that electric heat pumps can reduce annual energy usage by 1,193 GWh, with a median reduction of 107.5 MWh. Most buildings reduce energy usage by up to 200 MWh, with a few large energy consumers seeing annual reductions of up to 800 MWh, which is 7.5× the median. Reducing energy consumption makes buildings more energy efficient, and this makes electric heat pumps an attractive replacement for gas heating.



(a) Hourly gas demand (b) Energy consumption reduction (c) Additional electricity demand

Figure 10: Distribution of (a) average gas load demand by time of day after transitioning to electric heat pumps, (b) energy consumption reduction across buildings after transition to electric heat pumps, and (c) additional electric demand exerted on the system after transitioning from gas to electric heat pumps (cumulative).

6.6 Impact on the Electric Grid

After converting heating from gas to electric heat pumps, it is important to understand the expected load exerted on the electric grid. To compute the expected electric demand, we estimate the amount of heat energy generated from a building's gas consumption, and compute the electric energy required to generate the equivalent amount of heat energy (details in §5). Figure 10(c) depicts the CDF of electric demand required by heat pumps across the entire system. It shows that the median buildings increases electric demand by $\approx$ 72 MWh annually. The figure also indicates that most buildings increase electric demand by up to $\approx$ 200 MWh and only a few buildings having an additional annual demand of $>$ 200 MWh. Finally, the median annual gas energy is 179.1 MWh, which indicates a 60% reduction in absolute energy consumption. Further analysis is needed to study the impact of the extra load on the electric grid. However, these preliminary results show how the grid is expected to change as the penetration of heat pumps increases in buildings.

6.7 Implications of Results

In this section, we present a summary of the implications our results have on energy policy.

First, we have shown that making CO_2 footprint reduction the sole objective of decarbonization has several drawbacks. For instance, when CO_2 footprint is the sole objective of the optimization, the largest homes, which tend to belong to more affluent homeowners, are picked. This is because they tend to be the highest CO_2 emitters. However, since heat pump conversions come with government subsidies, this approach directs most subsidies to higher income households, which may not represent the best outcome for government policy. By considering CO_2 and energy efficiency, inefficient smaller homes as well large emitters would be chosen for transition, which leads to a more balanced transition.

Second, our results show that older buildings tend to be more energy inefficient, and may benefit more from transition than newer ones. This is due to improved building standards over time, as well as wear and tear in the older building. Consequently, a transition approach should prioritize these buildings more in decarbonization.

Finally, we show that transitioning to electric heat pumps have significant potential to reduce CO_2 emission, up to 81%. Therefore, to combat climate change, energy policy must move with haste towards decarbonization pathways such as electric heat pumps.

7 RELATED WORK

In this section, we discuss prior work on the energy transition, decarbonizing heating in buildings and electric heat pumps.

The energy transition. Multiple studies on transition pathways to a clean energy future have been conducted. Most of these studies examine the economic, environmental and societal benefits of a successful energy transition. For instance, Santamarta et al [29] evaluated the potential of transitioning to geothermal energy showing that in addition to CO_2 emission reduction, 66% energy savings and 13% ROI can be realized. Heinisch et al [15] propose an optimization model that interconnects various sectors of the energy ecosystem i.e. the electric grid, heating requirements and transportation, and heat pumps. Gonzalez-Salazar et al [13] explore pathways to phasing out coal-fired heating stations in favor CO_2-free energy sources. These studies are performed at macro scale i.e. energy generation and CO_2 mitigation are performed from centralized point of view. Our work is complementary to this work as it evaluates the potential of distributed transition at high granularity.

Decarbonizing heating. There have been numerous studies on decarbonizing space heating in the building sector [4, 16, 22, 27, 32]. For instance, Padovani et al [27] quantified the economic and decarbonization implications of replacing propane heating with cleaner electric energy sources such as solar heat pumps. Waite & Modi [32] propose and analyze a dual transition approach. Instead of replacing all existing fossil fuel heating with electric heat pumps, they propose a mix of both energy sources that gradually phases out fossil fuels over time. Leibowicz et al develop an energy system optimization model for decarbonizing residential buildings that incorporates transitioning to greener energy sources, migrating to more energy-efficiency appliances and improving the thermal properties of buildings e.g. through insulation retrofits. Hopkins et al propose transitioning to electric heat pumps for heating buildings. Finally, Baldino et al [4] analyze the cost and decarbonization benefits of hydrogen and renewable electricity as a replacement for heating. Our work is complementary to this work, as we evaluate the impact of multiple building selection strategies on CO_2 reduction. Since transitioning involves shifting energy load from one system to another i.e. from gas to electric, our work also quantifies the impact of such transition on the electric grid.

Electric heat pumps. The viability of electric heat pumps in place of gas heating in residential buildings has been widely studied [19, 20, 25, 28, 32, 34]. While some studies focussed on the evaluating the

performance of heat pumps in extreme temperatures [25, 31], others have analyzed their potential to decarbonize heating at various geographical scales. For instance, Johnshon & Krishnamoorthy [18] analyzed the cost and economic implications of transiting to electric heat pumps, and how it varies across different regions in the entire United States. Zhang et al [34] studied the decarbonization benefits of electric heat pumps using a simulated energy system of an entire city. Other studies [25, 31] have analyzed the applicability of heat pumps especially in extremely low temperatures. Our work is complementary to this work as we evaluate the viability of heat pumps at high granularity using real world data.

8 CONCLUSIONS

In this paper, we conducted a data-driven optimization study to analyze the potential of replacing gas heating with electric heat pumps to reduce carbon emissions in a city-wide distribution grid. We performed an in-depth analysis of gas consumption in the city and showed that ≈ 17 BCF of gas is consumed directly resulting in ≈360 GT of CO_2 emission annually. We presented a flexible multi-objective optimization (MOO) framework that optimizes carbon emissions reduction while also maximizing other aspects of the energy transition such as carbon-efficiency and energy inefficiency in buildings. We showed that transitioning to electric heat pumps can cut carbon emissions by up to 81% and energy required for heating by up to 60%. We also showed that optimizing for other aspects such as carbon-efficiency and energy inefficiency introduces tradeoffs with carbon emissions reduction that must be considered in a transition strategy. Finally, we presented preliminary results that examine the expected additional load on the electric grid by transitioning gas to electric heat pumps. We showed that a median building will add an annual load of 71.6 MWh to the electric grid.

ACKNOWLEDGMENTS

We thank our shepherd and the anonymous reviewers for their insightful comments. This research is supported by NSF grants 2136199, 2021693, 2020888 and 2105494.

REFERENCES

[1] U.S. Energy Information Administration. 2022. Total Energy Monthly Data. https://www.eia.gov/totalenergy/data/monthly/. (Accessed on 02/07/2022).
[2] Environmental Protection Agency. 2022. Greenhouse Gases Equivalencies Calculator - Calculations and References. https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references. (Accessed on 07/29/2022).
[3] Paris Agreement. 2015. Paris agreement. In *Report of the Conference of the Parties to the United Nations Framework Convention on Climate Change (21st Session, 2015: Paris). Retrived December*, Vol. 4. HeinOnline, 2017.
[4] Chelsea Baldino, Jane O'Malley, Stephanie Searle, Yuanrong Zhou, and Adam Christensen. 2020. Hydrogen for Heating? Decarbonization Options for Households in the United Kingdom in 2050.
[5] Sherri Billimoria, Leia Guccione, Mike Henchen, and Leah Louis-Prescott. 2021. The Economics of Electrifying Buildings: How Electric Space and Water Heating Supports Decarbonization of Residential Buildings. In *World Scientific Encyclopedia of Climate Change: Case Studies of Climate Risk, Action, and Opportunity*.
[6] Anna M Brockway and Pierre Delforge. 2018. Emissions Reduction Potential from Electric Heat Pumps in California Homes. *The Electricity Journal* (2018).
[7] US Cenus Bureau. 2019. American Community Survey Single-Year Estimates. *American Community Survey* (2019).
[8] United States Census Bureau. 2022. Highlights of Annual 2021 Characteristics of New Housing. https://www.census.gov/construction/chars/highlights.html. (Accessed on 07/29/2022).
[9] Christopher Dymond and Northwest Energy Efficiency Alliance. 2018. Air Source Heat Pump's Transformative Potential.
[10] US EIA. 2015. Residential Energy Consumption Survey: Energy Consumption and Expenditures Tables.
[11] Bob Formisano. 2020. Understanding Gas Furnace Types and AFUE Efficiency Categories. https://www.thespruce.com/gas-furnace-types-and-afue-efficiencies-1824743. (Accessed on 07/29/2022).
[12] Benjamin Goldstein, Dimitrios Gounaridis, and Joshua P Newell. 2020. The Carbon Footprint of Household Energy Use in the United States. *Proceedings of the National Academy of Sciences* (2020).
[13] Miguel Gonzalez-Salazar, Thomas Langrock, Christoph Koch, Jana Spieß, Alexander Noack, Markus Witt, Michael Ritzau, and Armin Michels. 2020. Evaluation of Energy Transition Pathways to Phase Out Coal for District Heating in Berlin. *Energies* (2020).
[14] Andreas Gschwend, Tobias Menzi, Stephen Caskey, Eckhard A Groll, and Stefan S Bertsch. 2016. Energy Consumption of Cold Climate Heat Pumps in Different Climates–Comparison of Single-stage and Two-stage Systems. *International Journal of Refrigeration* (2016).
[15] Verena Heinisch, Lisa Göransson, Mikael Odenberg, and Filip Johnsson. 2019. Interconnection of the Electricity and Heating Sectors to Support the Energy Transition in Cities. *International Journal of Sustainable Energy Planning and Management* (2019).
[16] A Hopkins, K Takahashi, D Glick, and M Whited. 2018. Decarbonization of Heating Energy Use in California Buildings. *Synapse Energy Economics* (2018).
[17] Srinivasan Iyengar, Stephen Lee, David Irwin, Prashant Shenoy, and Benjamin Weil. 2018. Watthome: A Data-driven Approach for Energy Efficiency Analytics at City-scale. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
[18] Brandon Johnson and Sreenidhi Krishnamoorthy. 2021. Where are Today's Residential Heat Pump Technologies Cost-Effective? *ASHRAE Transactions* (2021).
[19] Noah Kaufman, David Sandalow, Clotilde Rossi Di Schio, and Jake Higdon. 2019. Decarbonizing Space Heating with Air Source Heat Pumps. *Center Glob. Energy Policy* (2019).
[20] K Kontu, S Rinne, and S Junnila. 2019. Introducing Modern Heat Pumps to Existing District Heating Systems–Global Lessons from Viable Decarbonizing of District Heating in Finland. *Energy* (2019).
[21] Stefan Lechtenböhmer, Lars J Nilsson, Max Åhman, and Clemens Schneider. 2016. Decarbonising the Energy Intensive Basic Materials Industry Through Electrification–Implications for Future EU Electricity Demand. *Energy* (2016).
[22] Benjamin D Leibowicz, Christopher M Lanham, Max T Brozynski, José R Vázquez-Canteli, Nicolás Castillo Castejón, and Zoltan Nagy. 2018. Optimal Decarbonization Pathways for Urban Residential Building Energy Services. *Applied energy* (2018).
[23] Jessica Leung. 2018. Decarbonizing US buildings. *Center for Climate and Energy Solutions* (2018).
[24] Neeco. 2022. Ductless Mini-split Heat Pump Rebates and Incentives. https://neeeco.com/ductless-mini-split-heat-pump-rebates-and-incentives/. (Accessed on 07/25/2022).
[25] Francesco Neirotti, Michel Noussan, and Marco Simonetti. 2020. Towards the Electrification of Buildings Heating-Real Heat Pumps Electricity Mixes Based on High Resolution Operational Profiles. *Energy* (2020).
[26] Office of Energy Efficiency & Renewable Energy. 2022. Air-Source Heat Pumps. https://www.energy.gov/energysaver/air-source-heat-pumps. (Accessed on 07/26/2022).
[27] Filippo Padovani, Nelson Sommerfeldt, Francesca Longobardi, and Joshua M Pearce. 2021. Decarbonizing Rural Residential Buildings in Cold Climates: A Techno-economic Analysis of Heating Electrification. *Energy and Buildings* (2021).
[28] Alejandro Pena-Bello, Philipp Schuetz, Matthias Berger, Jörg Worlitschek, Martin K Patel, and David Parra. 2021. Decarbonizing Heat With PV-coupled Heat Pumps Supported by Electricity and Heat Storage: Impacts and Trade-offs for Prosumers and the Grid. *Energy Conversion and Management* (2021).
[29] Juan C Santamarta, Alejandro García-Gil, María del Cristo Expósito, Elías Casañas, Noelia Cruz-Pérez, Jesica Rodríguez-Martín, Miguel Mejías-Moreno, Gregor Götzl, and Vasiliki Gemeni. 2021. The Clean Energy Transition of Heating and Cooling in Touristic Infrastructures Using Shallow Geothermal Energy in the Canary Islands. *Renewable Energy* (2021).
[30] Greg Schivley, Ines Azevedo, and Constantine Samaras. 2018. Assessing the Evolution of Power Sector Carbon Intensity in the United States. *Environmental Research Letters* 13, 064018 (June 2018).
[31] Ben Schoenbauer, Nicole Kessler, David Bohac, and M Kushler. 2016. Field Assessment of Cold Climate Air Source Heat Pumps. In *ACEEE Conference on Energy Efficiency in Buildings*.
[32] Michael Waite and Vijay Modi. 2020. Electricity Load Implications of Space Heating Decarbonization Pathways. *Joule* (2020).
[33] Hansani Weeratunge, Gregorius R. Aditya, Simon Dunstall, Julian d. Hoog, Guillermo Narsilio, and Saman Halgamuge. 2021. Feasibility and Performance Analysis of Hybrid Ground Source Heat Pump Systems in Fourteen Cities. *Energy* (2021).
[34] Hongyu Zhang, Li Zhou, Xiaodan Huang, and Xiliang Zhang. 2019. Decarbonizing a Large City's Heating System Using Heat Pumps: A Case Study of Beijing. *Energy* (2019).

"I do not know": Quantifying Uncertainty in Neural Network Based Approaches for Non-Intrusive Load Monitoring

Vibhuti Bansal*† bansal.vibhuti25@gmail.com Bharati Vidyapeeth's College of Engineering, Delhi, India

Rohit Khoiwal*† khoiwalrohit.16@gmail.com Rajasthan Technical University India

Hetvi Shastri*‡ hshastri@umass.edu University of Massachusetts Amherst USA

Haikoo Khandor haikoo.ashok@iitgn.ac.in IIT Gandhinagar India

Nipun Batra nipun.batra@iitgn.ac.in IIT Gandhinagar India

ABSTRACT

Non-intrusive load monitoring (NILM) refers to the task of disaggregating total household power consumption into the constituent appliances. In recent years, various neural network (NN) based approaches have emerged as state-of-the-art for NILM. In conventional settings, NN(s) provide point estimates for appliance power. In this paper, we explore the question - can we learn models that tell when they are unsure? Or, in other words, can we learn models that provide uncertainty estimates? We explore recent advances in uncertainty for NN(s), evaluate 14 model variants on the publicly available REDD dataset, and find that our models can accurately estimate uncertainty without compromising on traditional metrics. We also find that different appliances in their different states have varying performance of uncertainty. We also propose "recalibration" methods and find they can improve the uncertainty estimation.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

Neural networks, Non-Intrusive Load Monitoring, Bayesian analysis, Uncertainty, Calibration

ACM Reference Format:

Vibhuti Bansal, Rohit Khoiwal, Hetvi Shastri, Haikoo Khandor, and Nipun Batra. 2022. "I do not know": Quantifying Uncertainty in Neural Network Based Approaches for Non-Intrusive Load Monitoring. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3563357.3564063

*These authors contributed equally to this research.

†Work done as an intern at IITGn

‡Work done as an IITGn student

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564063

1 INTRODUCTION

Non-intrusive load monitoring [9], or NILM, is the technique of decomposing overall power consumption into constituent appliances. Prior studies [5] suggest that providing appliance-wise energy consumption can help users potentially reduce their energy consumption by up to 15%.

Since the seminal work on NILM by George Hart [9], a variety of algorithms have been proposed in the recent past, including, but not limited to, time-series models such as additive factorial hidden Markov models [15], discriminative sparse coding [14], graph signal processing [10]. In 2015, Kelly et al. [13] proposed the application of neural networks for NILM. Since then, several neural network-based approaches for NILM have been proposed [12, 20, 24].

Conventional NN approaches for NILM provide point estimates, i.e. they may say that the fridge power consumption at 10 AM is 150 Watts. These conventional NN approaches do not quantify uncertainty. In contrast, a method that can quantify uncertainty may say that the fridge power at 10 AM is normally distributed with a mean of 150 Watts and a standard deviation of 5 Watts. However,if the model predicted that the fridge power is normally distributed with a mean of 150 Watts but a high standard deviation of 50 Watts, it means our model is unsure. The application designer or decision maker can factor in the uncertainty in predictions before deciding.

Recent literature has looked into methods of quantifying uncertainty in prediction from neural networks [3, 7, 22]. Such methods have been employed in various applications, including but not limited to medical imaging[18, 23]. A conventional approach would apply Bayesian analysis by putting a prior distribution over all the weights of the NN and then computing the posterior over the weights and the predictive distribution. However, such an approach would be computationally intractable [3]. Thus, more recently, various approximate inference methods have been proposed for quantifying the uncertainty in NNs. These methods include heteroskedastic NNs where we modify the architecture for regression to include two output nodes (one for the mean and one for the variance) instead of one output node. Other methods create an ensemble of NNs and combine the predictions from the individual models to obtain predictive uncertainty.

Prior literature has proposed a metric called expected calibration error and reliability diagrams (or calibration curves) to quantify the "goodness" of the predicted uncertainty (often also called how well a model is calibrated). We explain a well-calibrated model with

an example. Suppose our model's output is a normal distribution's mean (μ) and standard deviation (σ). Now, a 95% credible interval (CI) would correspond to 2σ, and in a well-calibrated model, 95% of the data points (ground truth) would lie within the predicted $\mu \pm 2\sigma$.

In this paper, we implement a total of 14 such model variants over the state-of-the-art NNs for NILM and evaluate these on three appliances on the publicly available REDD dataset [16]. We also propose a "re-calibration" method to improve the uncertainty quantification from our models. We now summarise the main questions and their answers that we explore in this paper:

(1) Do NNs with uncertainty achieve comparable error on conventional metrics to the baselines?
   (a) We find that we can achieve comparable or better performance on conventional metrics while additionally incorporating the notion of uncertainty.
(2) Are certain appliances or appliance states more prone to poor calibration?
   (a) We find that sparsely used appliances (like a dishwasher) have poor calibration compared to regularly used appliances (like a fridge).
(3) Can recalibration improve model uncertainty?
   (a) We find that for most of our models, our proposed recalibration scheme can improve the quantification of model uncertainty.

The rest of the paper is structured as follows. First, we discuss the methods of incorporating uncertainty in neural networks and various Bayesian approximation methods in Section 2. We also discuss quantification of predictive uncertainty and recalibration method. In Section 3, we outline the Seq2Point [3.1] and Bi-LSTM with attention[3.2] architectures which are state-of-the-art architectures for NILM. We discuss the evaluation in Section 4. We analyse the results in Section 4.4. After that, in Section 5, we go over prospective directions for this work before concluding in Section 6.

2 UNCERTAINTY IN NEURAL NETWORKS

We now discuss different architectures and approximate inference technique to estimate uncertainty in neural networks. We summarise these techniques and architectures in Figure 1. We direct the reader to recent surveys for an in-depth discussion on the different methods for estimating uncertainty in neural networks. [7, 22]

2.1 Homoskedastic Neural Networks

Homoskedasticity in the context of machine learning means models which have the same variance distribution across all input data. The "regular" neural network models (shown in Figure 1 (a)) or the linear regression models assume homoskedasticity. Under the assumption of homoskedasticity, the output from a neural network is distributed as:

$$\hat{y} \sim \mathcal{N}(\mu(\boldsymbol{x}), \sigma^2)$$

where the variance σ^2 is assumed constant. Thus, the loss function of the model is given by minimising the negative log-likelihood (equivalent to maximising the likelihood) under the i.i.d. assumption is given as:

$$\text{Loss} = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}$$

where y_i and $\hat{y}_i$ denote the ground truth and prediction of the i^{th} (where $i \in \{1 \cdots N\}$) data point. The prediction of the i^{th} data point ($\boldsymbol{x_i}$) can be computed by running the forward pass of the neural network on $\boldsymbol{x_i}$, or, in short, we can write: $\mu_i = NN(\boldsymbol{x_i})$. We can see that minimising the negative log-likelihood naturally leads us to mean squared error as the cost function. While often not discussed, one can calculate the maximum likelihood estimate for the variance σ^2 term as follows:

$$\sigma^2_{MLE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - NN(\boldsymbol{x_i}))^2$$

The loss functions can be optimised using the well-known gradient descent-based approaches. Given the "simplistic" notion of uncertainty from homoskedastic models, in practice, these models are treated as models without uncertainty quantification. Following the usual practice, we do not study the uncertainty estimates from these models in this paper.

2.2 Heteroskedastic Regression

In contrast to the above discussed homoskedastic regression model, heteroskedastic regression model (shown in Figure 1 (b)) can learn different variance for different data points. Thus, in heteroskedastic model, in addition to estimating $\mu(\boldsymbol{x})$ as a function of the input, we also estimate/learn $\sigma(\boldsymbol{x})$ as a function of the input. Similar to homoskedastic regression, we can define the loss (to be minimised) as the negative log-likelihood. But, unlike the homoskedastic regression case, we cannot assume σ^2 to be constant. Thus, can write the loss as follows (ignoring constants):

$$\text{Loss} = \frac{\sum_{i=1}^{N} \left(-\frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} (y_i - \hat{y}_i)^2 \right)}{N} \quad (1)$$

Similar to homoskedastic regression, the loss functions can be optimised using the well-known gradient descent based approaches.

Both the models discussed thus far consider only the data (or aleatoric) uncertainty. They do not consider the uncertainty in estimating the parameters (or the epistemic) uncertainty. One way to obtain epistemic uncertainty is by creating an ensemble of models. We first briefly discuss how we can obtain the prediction and uncertainty from an ensemble.

2.3 Prediction from an ensemble of NNs

We consider an ensemble of two kinds of models: homoskedastic and heteroskedastic models. If we have an ensemble of N homoskedastic NNs and their prediction for an input (obtained via the forward pass) is given as: μ_i, then the predicted mean and standard deviation of the ensemble is calculated by:

$$\mu_{\text{ensemble}} = \frac{\sum_{i=1}^{N} \mu_i}{N} \quad (2)$$

$$\sigma_{\text{ensemble}} = \sqrt{\frac{\sum_{i=1}^{N} (\mu_i - \mu_{\text{ensemble}})^2}{N}} \quad (3)$$

In heteroskedastic regression, for an input datapoint, we predict two neurons that correspond to mean (μ_i) and sigma (σ_i) $\forall i \in \{1, \cdots N\}$, and the output is a Gaussian (or Normal) distribution. The ensemble of N such models will produce a mixture of Gaussian

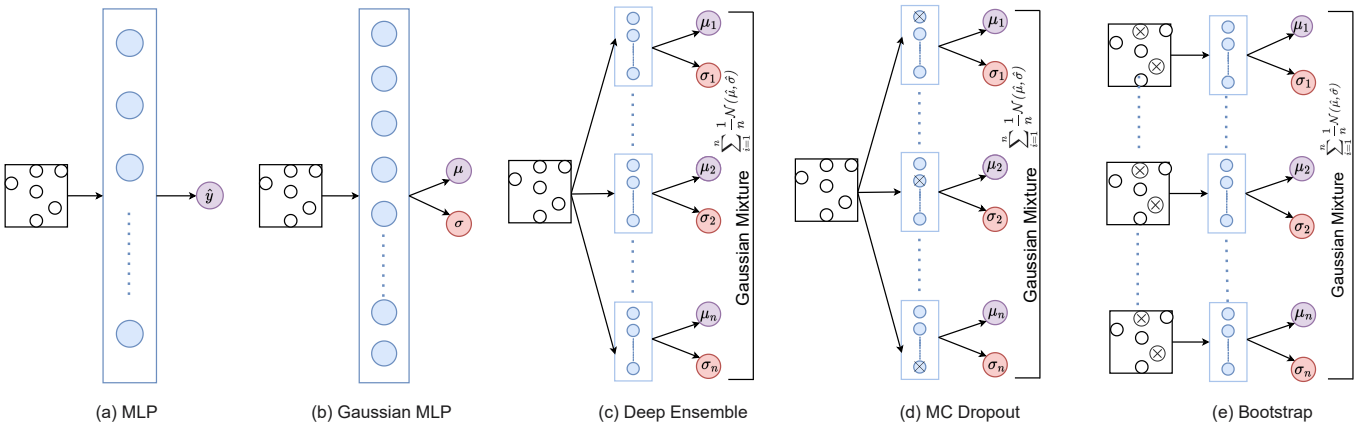


Figure 1: Various neural network model architectures for computing uncertainty in predictions

distributions [17]. Under the assumption that each distribution has an equal weight in the mixture, the resulting mean and sigma are determined. Calculating the predictive mean is same as equation 2. The standard deviation of the ensemble is calculated as:

$$\sigma_{\text{ensemble (Heteroskedastic)}} = \sqrt{\frac{\sum_{i=1}^{N}(\sigma_i^2 + \mu_i^2)}{N} - \mu_{\text{ensemble}}^2} \quad (4)$$

In this paper, we have implemented three ensemble methods, namely Monte Carlo (MC) Dropout, Deep Ensembles and Bootstrap with Homoskedastic and Heteroskedastic models. We now discuss these three methods.

2.4 MC Dropout

Monte Carlo Dropout [8] (called MC Dropout from now onwards) trains a neural network (either homoskedastic or heteroskedastic) as usual. However, at the time of prediction, it randomly drops out nodes from the network. The probability of a node being dropped (or retained) is given as per the Bernoulli distribution. We can note that the operation of MC Dropout is similar to the regular dropout considered in the context of reducing model overfitting [21]. However, the key difference is the application of dropout at test time. Every forward pass of the network can drop out different nodes (given each forward pass accepts a different random seed as an argument), and result in a different prediction for a given input, as shown in Figure 1(d). Importantly, the MC dropout method can be considered as an approximation of Bayesian deep gaussian processes [8]. Thus, the MC dropout method, though simple has strong theoretical properties.

2.5 Bootstrap

Bootstrap aggregating is a technique used to reduce the variance of a machine learning model [4]. It works by training multiple models on different subsets of the data as in Figure 1(e) and then averaging the predictions of all the models. This can help reduce overfitting and improve the overall performance of the model. Using the bootstrap method, unlike the MC dropout method we train N different models independently, each given a different subset of the dataset. Thus, the computation and memory requirement for Bootstrap based method to create an ensemble of NNs is expensive.

2.6 Deep Ensemble

The Deep ensemble method [17] is similar to the bootstrap method and trains N independent models as shown in Figure 1c. The key difference between the bootstrap and deep ensemble model is that each model in the ensemble learns over the entire dataset unlike the bootstrap method The models can be of different types (e.g., different neural network architectures), or they can be different instances of the same type of model (e.g., different random initialisations of the same architecture).

2.7 Quantifying predictive uncertainty

Having discussed various methods of estimating uncertainty using neural networks, we now discuss methods to quantify the "goodness" of the predicted uncertainty (often also called how well a model is calibrated). We explain a well-calibrated model with an example in Figure 2. We take a ground truth or true function $f(x) = x\sin(x)$ and learn a probabilistic model (learning the mean and the standard deviation) over the training data. Then, over the test data, we plot the 90% confidence interval. For the normal distribution, 90% CI refers to the $\mu \pm 1.64\sigma$ band. However, we can see that only 72.5% of the observations fall within the $\mu \pm 1.64\sigma$ band. From here on, we refer to the *chosen* CI as p and the empirically found fraction of points within the band corresponding to the CI of p as $\hat{p}$. Ideally, we would like $\hat{p}$ to be the same as p. In Figure 3, we show the reliability diagram (or calibration curve) for the above-mentioned example. From Figure 3, we can note that the relationship between $\hat{p}$ and p, uncalibrated (shown in blue), lies below the ideal ($\hat{p} = p$) line. It should be noted that to generate such a reliability diagram, we choose varying CIs (p) and then find the corresponding $\hat{p}$.

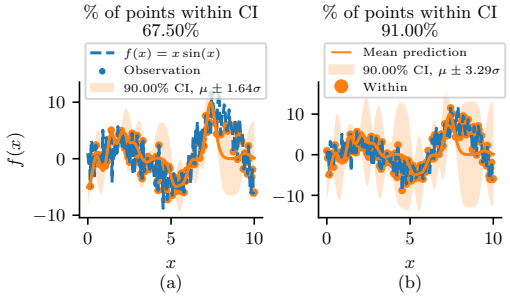


Figure 2: 33% Increase in number of data points lying in 90% Confidence Interval

The reliability curve can help visually understand quality of the model's calibration. We now discuss a quantitative measure called expected calibration error (ECE) to measure the model calibration. To compute the ECE, we choose a set of p values (say, 0.01, 0.02, $\cdots$, 0.99 as an example) and compute the corresponding $\hat{p}(p)$ as a function of p. Finally, we can compute the ECE as:

$$\text{ECE} = \frac{\sum_{p=1}^{P} |\hat{p}(p) - p|}{P}$$

Lower ECE value indicates a better calibration.

2.8 Model Recalibration

We now discuss our proposed method to improve model calibration or to reduce ECE. We continue working with our above running example from Figure 2 and Figure 3. We previously discussed: the *uncalibrated* (shown in blue) relationship between $\hat{p}$ and p lies below the ideal ($\hat{p} = p$) line. To improve this relationship, we want for example 90% CI to correspond to more than the 72.5% points from the uncalibrated model. Therefore, we learn a function g mapping $\hat{p}$ to p using Isotonic regression. Isotonic regression is well-suited for this specific function as it learns monotonically increasing non-parametric relationship. Finally, at test time, say, we want to get 90% of points within the 90% CI, we find $g(90\%)$, which in this example would map to a number higher than 90% (in our example this is: 92.7%) meaning that we need to increase our band in order to capture approximately 90% of the datapoints. Going from CI of 90% on original model to a CI of 92.7% on the original models means increasing the band from $\mu \pm 1.64\sigma$ to $\mu \pm 1.79\sigma$. This in turn, leads to a $\hat{p} = 78\%$, which is greater than the $\hat{p} = 72.5\%$ of the uncalibrated model (Figure 2). Further, we can repeat this procedure for the different values of p to obtain the improved reliability diagram for the calibrated model (shown in orange in Figure 3).

An important detail about the recalibration process is that we fit our NN on the training dataset, recalibrate (or learn the above-mentioned function g) on a previously unseen dataset called the calibration dataset. We split the dataset with 75% training and 25% calibration dataset. It should also be noted that our recalibration procedure only changes the model uncertainty (σ) without affecting the mean (μ) prediction.

3 NEURAL NETWORKS FOR NILM

We now discuss two state-of-the-art NN methods used for NILM.

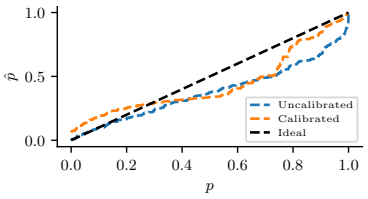


Figure 3: Reliability diagram helps quantify the quality of our uncertainty estimates

Appliance	Training House Number	Testing House Number
Fridge	1, 2, 3, 5	6
Dishwasher	1, 3	2
Microwave	1, 3	2

Table 1: The training and test settings for different appliances for our experiments

3.1 Seq2Point

Seq2Point (S2P) [24] maps a sequence of mains power to a point appliance power prediction. The NN architecture is composed of a sequence of 1d convolution filters and dropout. For more details, we refer the reader to prior research [2, 24].

3.2 BiLSTM with Attention Mechanism

The S2P model while considered the state-of-the-art model, was known to perform poorly on sparsely used appliances such as dishwasher and microwave. Recent work [19] have proposed using bi-directional LSTM models with attention for NILM and have shown these models to work well even for sparsely used appliances. We direct the reader to prior research for the detailed model architecture [20].

4 EVALUATION

We now provide the evaluation setup for answering the questions we raised in the introduction. Our work is fully reproducible. We provide the code in our repository[1].

4.1 Datasets

We have used the publicly available REDD dataset [16] for our research. The dataset consists of several appliances collected over several weeks from six different residences. We used information from three appliances for this study: the refrigerator, dishwasher, and microwave. Other appliances have far less data, and most families do not have access to it. Additionally, the dishwasher and microwave require human operation and are only occasionally utilised, whereas the fridge might be regarded a background appliance that operates without human interaction. Additionally, devices like the dishwasher frequently function in several modes (drying, heating, etc.) and demand varied amounts of power draw for these different states. Furthermore, we downsampled the data for both the mains and the appliances to a minute frequency using the pre-processing routines from NILMTK, as done in prior work [20].

[1] https://github.com/VibhutiBansal-11/NILM_Uncertainty

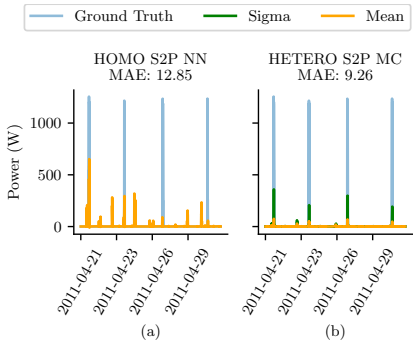


Figure 4: Models incorporating uncertainty such as Heteroskedastic S2P with MC dropout (shown in (b)) can achieve lower MAE than models without uncertainty such as Homoskedastic S2P NN (shown in (a)) for a sparsely used appliance such as the dishwasher. The S2P NN (a) shows high false positives (predicting the dishwasher to be ON when it is actually OFF) in comparison to the Heteroskedastic S2P MC (b).

4.2 Metrics

We use two different metrics to quantify our model performance. First, we use the conventional mean absolute error (MAE) metric defined as following: MAE $= \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$. Here, n is the number of samples, $\hat{y}_i$ is the predicted appliance reading, and y_i is the ground truth reading of an appliance. The MAE has been used across several prior NILM studies[6]. Further, we note that a lower MAE indicates better performance.

We used expected calibration error (ECE) discussed in Section 2.7 to quantify the uncertainty performance of our model.

4.3 Experimental Setup

We discuss the dataset split chosen for training and testing in Table 1. Our dataset split choice is based on prior literature [20], and on the basis of availability of the appliance data across these homes. In the interest of space, we link the hyperparameter space in a dedicated page in our Github[2]. Like our dataset split, our hyperparameter choices was inspired by previous literature [2, 20]. We used 4 X NVidia A100 GPUs for training our models, and JAX[3] and Flax[4] for creating our neural network models. All our models are compatible in the NILMTK ecosystem [1, 2].

4.4 Results and Analysis

We now present our results based on the questions we raised in the introduction section of this paper. The main result in Table 2 compares the MAE and the ECE across the different models and appliances.

[2]https://github.com/VibhutiBansal-11/NILM_Uncertainty/blob/master/hyperparameters.md

[3]https://jax.readthedocs.io

[4]https://flax.readthedocs.io/en/latest/overview.html

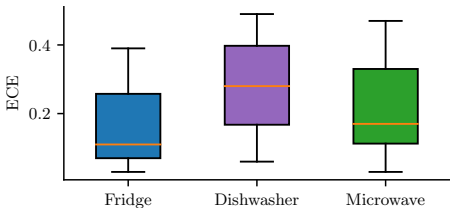


Figure 5: The expected calibration error (ECE) quantifying calibration performance for three appliances across different models presented in Table 2 is generally lower for the fridge compared to sparsely used appliances like the dishwasher and the microwave.

4.4.1 *Do NNs with uncertainty achieve comparable error on conventional metrics to the baselines?*

We can see from Table 2, that the MAE for the models that provide uncertainty such as S2P homoskedastic (MC, DE, BS), and heteroskedastic (NN, MC, DE, BS), is comparable or better than the baseline (Homoskedastic S2P/LSTM) models that do not incorporate uncertainty. As an example, the S2P homoskedastic model has a MAE of 26.16 for the fridge, whereas S2P homoskedastic with DE achieves a lower MAE of 24.73. Similarly, for the LSTM-based models, the MAE among model variants with uncertainty is comparable to the MAE of models without uncertainty. We believe that our finding that **we can achieve comparable or better performance on conventional metrics while additionally incorporating the notion of uncertainty** is an important finding for the community going forward.

Interestingly, we can significantly reduce the MAE for the dishwasher by using models incorporating uncertainty. For example, the S2P homoskedastic NN has a MAE of 12.85 compared to the improved MAE of 9.26 for the S2P heteroskedastic model with MC Dropout. We now explain this finding in Fig 4, where we can see that baselines S2P homoskedastic NN (MAE of 12.85) is better at predicting the peaks (ground truth) but it is also giving high false positives (wrongly predicting the dishwasher to be ON when it is actually OFF) which increases its MAE. While the model incorporating uncertainty (S2P heteroskedastic MC Dropout with MAE error of 9.26) does not predict the peaks well but has less false positives which results in lower MAE. Further, from Fig 4(b), we can also observe the uncertainty in the prediction (sigma, shown in green). is higher when the dishwasher changes state from OFF to ON. **Higher uncertainty when an appliance changes state is expected as the model is likely to be uncertain during the transition and will likely get more confident once it observes more samples from the changed state**.

4.4.2 *Are certain appliances or appliance states more prone to poor calibration?*

From Fig 5 and Table 2, we can observe that the ECE for the fridge is generally lower than that of sparsely used appliances (dishwasher and microwave). As discussed earlier, and in prior literature [20], NILM methods generally perform worse for sparsely used appliances in comparison to appliances such the fridge or air conditioner.

	Fridge			Dishwasher			Microwave		
	MAE	ECE	C.ECE	MAE	ECE	C.ECE	MAE	ECE	C.ECE
Model : S2P									
Homoskedastic									
NN	26.16	-	-	12.85	-	-	11.18	-	-
MC	26.22	0.25	0.23	12.97	0.48	0.42	11.17	0.47	0.45
DE	24.73	0.26	0.27	12.46	0.43	0.37	**11.16**	0.43	0.35
BS	**24.69**	0.24	0.26	11.49	**0.06**	0.21	11.25	0.30	0.18
Heteroskedastic									
NN	26.91	0.13	0.05	9.61	0.19	**0.06**	12.46	**0.03**	0.08
MC	26.38	**0.03**	0.19	**9.26**	0.16	0.12	12.56	0.05	0.18
DE	26.85	0.04	0.03	9.81	0.13	0.06	12.66	0.20	**0.06**
BS	26.68	0.07	0.11	10.21	0.43	*0.17*	13.27	0.22	0.35
Model: LSTM									
Homoskedastic									
NN	36.51	-	-	12.29	-	-	16.76	-	-
MC	36.57	0.33	*0.19*	12.29	0.49	0.39	16.77	0.43	*0.25*
DE	37.05	0.39	*0.25*	10.33	0.29	0.21	15.60	0.34	0.17
BS	36.66	0.37	*0.23*	9.99	0.14	0.18	14.80	0.12	0.15
Heteroskedastic									
NN	32.80	0.06	0.07	10.07	0.23	0.07	12.61	0.12	0.14
MC	32.83	0.07	0.04	10.11	0.27	0.05	12.61	0.14	0.12
DE	33.38	0.08	0.06	10.07	0.30	0.05	12.81	0.11	0.13
BS	33.16	0.09	**0.02**	11.12	0.29	0.07	13.06	0.04	0.11

Table 2: Mean absolute error (MAE), Expected Calibration Error pre calibration (ECE) and Expected Calibration Error post calibration (C. ECE) for three appliances across 16 model variants. The best performing model for each metric has been made bold and the value of C.ECE has been made italic where the improvement of error (C.ECE - ECE) is maximum

However, our findings suggest that not only is the performance in terms of MAE poor for sparsely used appliances, but, the models with uncertainty also have worse calibration for sparsely used appliances, in comparison to regularly used appliances Thus, these models present significant scope in improving both the MAE as well as calibration performance.

We now discuss the reliability diagram and the predicted power for the three appliances across a subset of the models. We choose the models and the plotted time window for illustrative purposes. However, it should be noted that the reported computed metrics are for the entire dataset as shown in Table 2. We first discuss the results for fridge. In Figure 6(a) we observe that the prediction for the Homoskedastic S2P model with bootstrap matches the ground truth well, resulting in a low MAE of 24.69. However, the ECE of 0.25 is high in comparison to other models (as seen from Figure 6(d)). From the reliability diagram (for now we direct the reader to only study the curve labelled *Total*) we can observe that the empirical fraction of points ($\hat{p}$) is below the ideal line ($\hat{p} = p$ line). This indicates that the learnt model is over-confident, i.e. the model thinks that it predicts the mean well and thus needs a low uncertainty band. However, if the uncertainty of the model were increased, especially during the fridge ON states (around 07:30 to 08:10 hours, 08:40 to 09:30 hours, and 10:10 to 10:40 hours), the calibration performance will improve (reduction in ECE).

From Figure 6(b), we can note that the prediction during the ON state is wrong. However, interestingly, the uncertainty in the prediction (sigma, shown in green) is high, especially during the time when the prediction is particularly bad. The high uncertainty helps in achieving a well-calibrated model. We confirm this in Figure 6(e), where we observe that the empirical fraction of points (for *Total* curve) ($\hat{p}$) is close to the ideal line ($\hat{p} = p$ line). From Figure 6(c), we can observe the predictions for Heteroskedastic LSTM model are comparable in terms of ECE to Heteroskedastic S2P model with bootstrap shown in Figure 6(b). However, interestingly, the corresponding calibration curves (Figure 6(e) and (f)) are substantially different when we consider the calibration curves separately for the ON and the OFF states. In Figure 6(e), the model is under-confident, i.e. it predicts a high value of uncertainty for both the ON and the

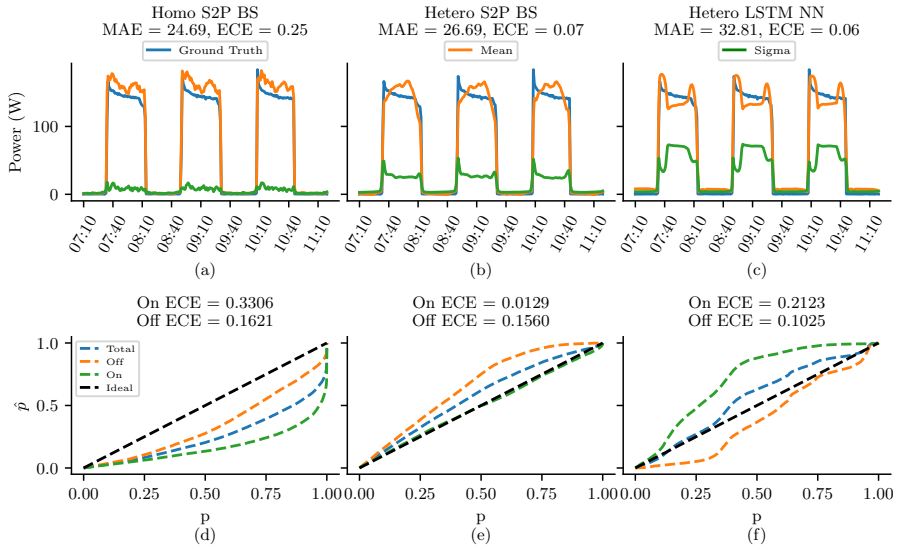


Figure 6: Predicted power and reliability diagrams for fridge across different models (a) Homoskedastic S2P model with bootstrap showing the best MAE (lowest) (b) Heteroskedastic S2P model with bootstrap showing a comparitively low MAE but low ECE (c) Heteroskedastic LSTM model showing a high MAE but low ECE

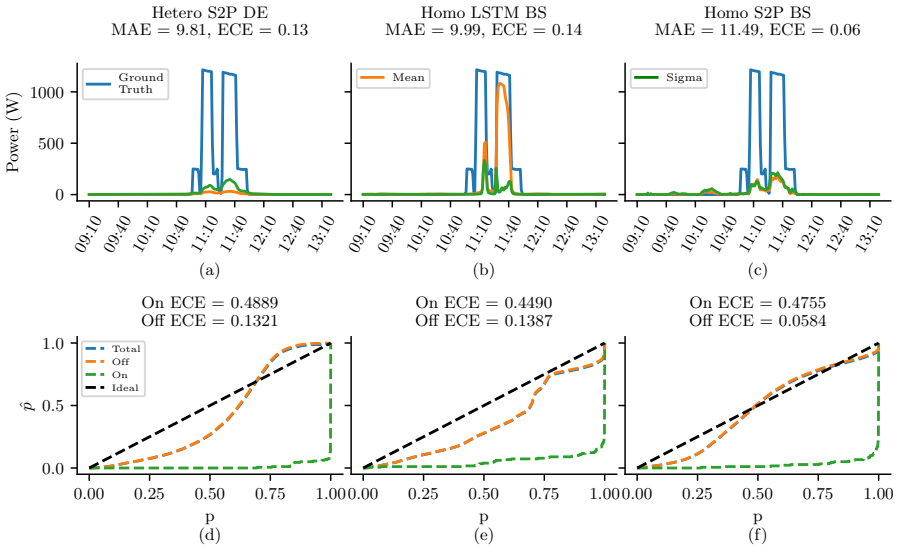


Figure 7: Predicted power and reliability diagrams for dish washer across different models (a) Heteroskedastic S2P model with deep ensemble showing the low MAE (b) Homoskedastic LSTM model with bootstrap showing a high MAE and high ECE but low ECE for ON-state (c) Homoskedastic S2P model with bootstrap showing a high MAE but best ECE (lowest)

OFF states. However, in Figure 6(f), the model is over-confident for the OFF state and under-confident for the ON state. These findings highlight that **different appliance states can have highly varying calibration curves**, and **we can achieve an overall low calibration error if the individual states calibration errors cancel out each other**.

We now discuss the calibration and predicted power for dishwasher. The homoskedastic S2P BS (Figure 7(c) and (f)) ECE value is low, corresponding to 0.06 but we are unable to obtain an uncertainty estimate that can capture the ON state ground truth within an appropriate confidence interval because the model is overconfident in this state. However, as the data is largely biased towards the OFF state, the final ECE values are low and the *Total* $\hat{p}$ is close to the $\hat{p} = p$ line. Similarly, for the heteroskedastic S2P DE (Figure 7(a) and (d)). In contrast, in Figure 7(b) for the homoskedastic LSTM with bootstrap model, the ECE is comparable (0.14) to homoskedastic

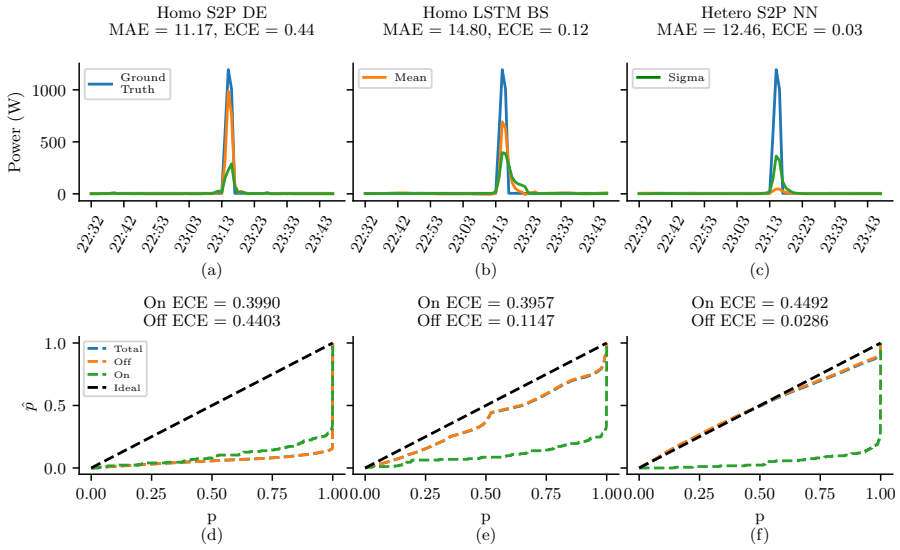


Figure 8: Predicted power and reliability diagrams for microwave across different models (a) Homoskedastic S2P model with deep ensemble showing the best MAE (lowest) (b) homoskedastic LSTM model with bootstrap showing a high MAE but low ECE (trade-off) (c) Heteroskedastic S2P model showing a high MAE but best ECE (lowest)

S2P BS (Figure 7(c)). However, the model is doing a better job at predicting the second peak (around 11:10 to 11:40 hours). Furthermore, the uncertainty (sigma, shown in green) increases on the right side of the second peak (around 11:40 hours), where the predicted power deviates from the ground truth. We confirm from Figure 7(e), that indeed for homoskedastic LSTM model with bootstrap, the ECE for the ON state is better than the other compared models. This leads us to our next learning: **A good ECE may hide the imbalance between the different states, and for a thorough analysis, it is recommended to consider the state-wise ECE.**

Our findings on similar experiments done on the microwave (Figure 8) are comparable to the findings for fridge and dishwasher. Importantly, as expected, **estimating the uncertainty accurately (low ECE) for the ON state for sparse appliances like dishwasher and microwave is non-trivial**. Overall, from the above experiments, we can conclude that **there is an important trade-off between the two considered metrics: MAE and ECE, and different applications may call for a nuanced choice of metric for evaluating performance.**

4.4.3 *Can recalibration improve model uncertainty?*

We applied our isotonic regression-based recalibration approach previously discussed in Section 2.7. From Table 2, we note that the ECE post calibration (C.ECE) improves (reduces) for most models in comparison to the ECE before calibration. We now dive deeper into some specific illustrative examples to show the effect of recalibration on the three appliances. We use the 95% confidence interval (CI) for our experiments. First, in Figure 9(a) we can observe that S2P homoskedastic MC model originally had 65% of points ($\hat{p} = 0.65$) in 95% confidence interval ($p = 0.95$) which is visibly improved after recalibration to 80% ($\hat{p} = 0.80$) in Figure 9(b). We can further confirm from Figure 9(c) that the reliability diagram improves post-calibration. Importantly, we can note from Figure 9(b) that a much higher proportion of the observation during the ON state (07:30 to 08:10) now fall within the CI, in comparison to Figure 9(a).

Similarly, for the dishwasher (Figure 10) and microwave (Figure 11) there is an improvement in model uncertainty as quantified by the reduction of the gap between p and $\hat{p}$. We may also note from Figure 10 and Figure 11, that the uncertainty quantification improves for both the ON and the OFF states. However, quantifying uncertainty for the ON state even post calibration has a significant scope for improvement.

We now analyse why ECE can increase post calibration for some models and appliances as shown in Table 2. This trend can be attributed to the contrasting nature of confidence of the calibration set and test set. We can see from Figure 12(a) that 90% of points lie in 95% CI before calibration for the fridge for heteroskedastic S2P MC dropout model. Instead of increasing from 90 to 95% (ideally) post calibration, we observe only 84% points which is worse and hence increases the test ECE as seen in Figure 13(b). The calibration set curve before calibration was under confident (above the ideal line), as seen in Figure 13. Thus, to match the ideal curve, post calibration, the $\hat{p}$ would be reduced for the same value of p. On the contrary, the model before calibration was overconfident on the test set, where $\hat{p}$ was below the ideal curve. Thus, recalibration further pushes down $\hat{p}$ making the ECE worse. We can thus conclude that **good recalibration requires similar characteristics between the calibration and the test set.**

5 LIMITATIONS, DISCUSSION AND FUTURE WORK

(1) In the future we plan to study performance of the 14 model variants on more datasets and appliances.
(2) In this paper we assumed normal distribution (as is the standard in the machine learning community). In the future, we

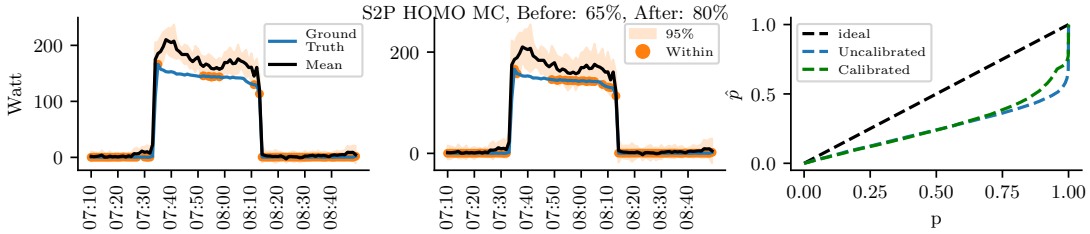


Figure 9: Effect of recalibration on fridge for homoskedastic S2P MC dropout model: (a) Corresponding to the 95% confidence interval, our uncalibrated model has only 65% of the observed data points; (b) the calibrated model in contrast has a higher fraction of 80% points; (c) the reliability diagram showing the improvement in uncertainty quantification post calibration.

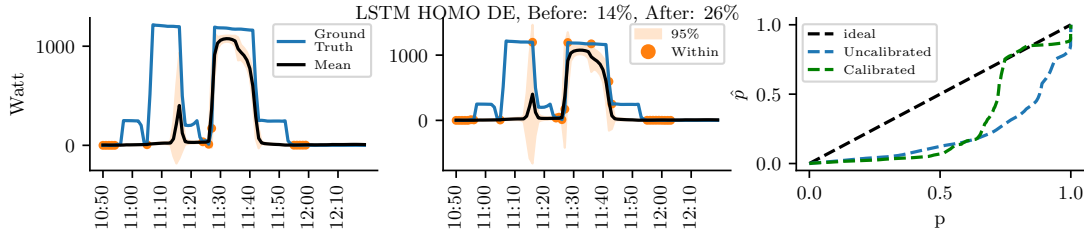


Figure 10: Effect of recalibration on dishwasher for homoskedastic LSTM Deep Ensemble Model: (a) Corresponding to the 95% confidence interval, our uncalibrated model has only 14% of the observed data points; (b) the calibrated model in contrast has a higher fraction of 26% points; (c) the reliability diagram showing the improvement in uncertainty quantification post calibration

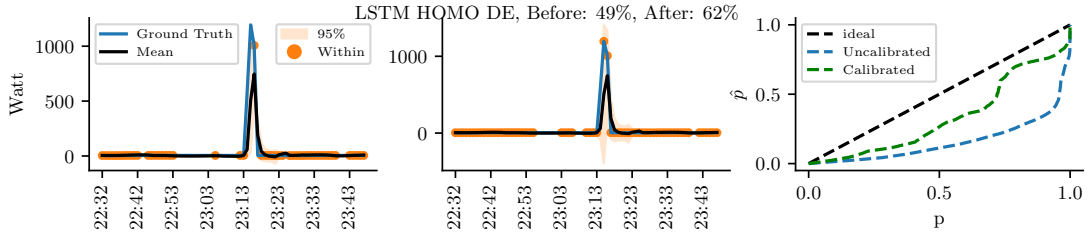


Figure 11: Effect of recalibration on microwave for homoskedastic LSTM deep ensemble model: (a) Corresponding to the 95% confidence interval, our uncalibrated model has only 49% of the observed data points; (b) the calibrated model in contrast has a higher fraction of 62% points; (c) the reliability diagram showing the improvement in uncertainty quantification post calibration

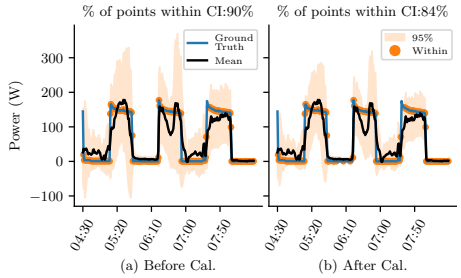


Figure 12: Effect of recalibration for fridge for heteroskedastic S2P model with MC Dropout - (a) Before Calibration 90% points in 95% CI (b) After Calibration 84% points in 95% CI

plan to study likelihoods such as log-normal distribution to strictly enforce non-negativity.

(3) In this paper we presented several approximate inference methods like Deep Ensemble, MC Dropout and Bootstrapping to obtain uncertainty. While we also evaluated on Markov Chain Monte Carlo (MCMC) based methods (where methods like NUTS [11]) on NILM data, we did not report the results as the experiments are significantly more time consuming. However, in the future, we plan to compare our methods proposed in this paper to MCMC based methods too.

(4) We discussed that the different characteristics of the calibration and test set can result in recalibration making the uncertainty performance worse. In the future, we plan to study techniques to understand the suitability of a given

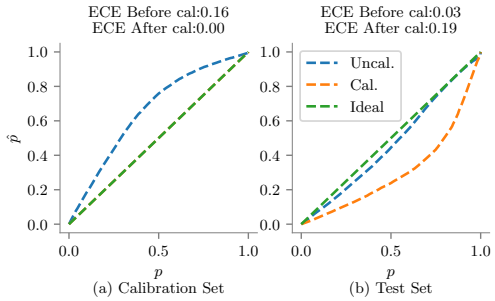


Figure 13: Test ECE increases by 0.13 after calibration for fridge for heteroskedastic S2P model with MC Dropout (a) Model is underconfident on calibration set, it becomes ideally confident with 0 ECE after recalibration (b) Model was overconfident before calibration and confidence after recalibration increases even more due to the nature of calibration set making ECE worse

calibration set to improve the uncertainty performance on an unseen test dataset.

(5) In NILM applications, the data distribution can drift over time owing to reasons such as: i) changes in weather conditions; ii) appliance wear and tear; iii) change in operational usage; etc. OOD examples are unlikely to contain the same patterns as training distribution examples. This may limit the generalization ability. Thus, in the future, we plan to study the uncertainty quantification for the out of distribution (O.O.D.) setting.

6 CONCLUSIONS

NN methods have proven to be the state-of-the-art models for NILM. In this paper, we have shown how to adapt existing architectures to provide predictive uncertainty. We took a NILM specific flavour to our work and discuss our findings in the NILM context. As an example, we showed that calibration needs to be studied separately for different states of an appliance. Finally, we have highlighted the shortcomings of existing approaches to quantify uncertainty. We hope this paper opens discussions in the NILM and the BuildSys community around predictive uncertainty.

7 ACKNOWLEDGEMENTS

We would like to thank Cisco Research for sponsoring this research. For this project, Vibhuti Bansal was supported via the Google exploreCS Research 2022 fellowship.

REFERENCES

[1] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*. 265–276.

[2] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. Towards reproducible state-of-the-art energy disaggregation. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 193–202.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Network. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1613–1622. https://proceedings.mlr.press/v37/blundell15.html

[4] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[5] Sarah Darby et al. 2006. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays* 486, 2006 (2006), 26.

[6] Anthony Faustine, Lucas Pereira, Hafsa Bousbiat, and Shridhar Kulkarni. 2020. UNet-NILM: A Deep Neural Network for Multi-Tasks Appliances State Detection and Power Estimation in NILM. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring* (Virtual Event, Japan) *(NILM'20)*. Association for Computing Machinery, New York, NY, USA, 84–88. https://doi.org/10.1145/3427771.3427859

[7] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1050–1059. https://proceedings.mlr.press/v48/gal16.html

[8] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning *(ICML'16)*. JMLR.org, 1050–1059.

[9] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.

[10] Kanghang He, Lina Stankovic, Jing Liao, and Vladimir Stankovic. 2016. Non-intrusive load disaggregation using graph signal processing. *IEEE Transactions on Smart Grid* 9, 3 (2016), 1739–1747.

[11] Matthew D. Hoffman and Andrew Gelman. 2011. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. https://doi.org/10.48550/ARXIV.1111.4246

[12] Patrick Huber, Alberto Calatroni, Andreas Rumsch, and Andrew Paice. 2021. Review on deep neural networks applied to low-frequency nilm. *Energies* 14, 9 (2021), 2390.

[13] Jack Kelly and William Knottenbelt. 2015. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. 55–64.

[14] J Zico Kolter, Siddharth Batra, and Andrew Y Ng. 2010. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*. 1153–1161.

[15] J Zico Kolter and Tommi Jaakkola. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial intelligence and statistics*. 1472–1482.

[16] J. Zico Kolter and Matthew J. Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. In *IN SUSTKDD*.

[17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).

[18] Max-Heinrich Laves, Sontje Ihler, Jacob F. Fast, Lüder A. Kahrs, and Tobias Ortmaier. 2020. Well-Calibrated Regression Uncertainty in Medical Imaging with Deep Learning. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning (Proceedings of Machine Learning Research, Vol. 121)*, Tal Arbel, Ismail Ben Ayed, Marleen de Bruijne, Maxime Descoteaux, Herve Lombaert, and Christopher Pal (Eds.). PMLR, 393–412. https://proceedings.mlr.press/v121/laves20a.html

[19] Veronica Piccialli and Antonio M. Sudoso. 2021. Improving Non-Intrusive Load Disaggregation through an Attention-Based Deep Neural Network. *Energies* 14, 4 (Feb 2021), 847. https://doi.org/10.3390/en14040847

[20] Hetvi Shastri and Nipun Batra. 2021. Neural Network Approaches and Dataset Parser for NILM Toolkit. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (Coimbra, Portugal) *(BuildSys '21)*. Association for Computing Machinery, New York, NY, USA, 172–175. https://doi.org/10.1145/3486611.3486652

[21] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15 (2014), 1929–1958.

[22] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty Estimation Using a Single Deep Deterministic Neural Network. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 9690–9700. https://proceedings.mlr.press/v119/van-amersfoort20a.html

[23] Yanwu Yang, Xutao Guo, Yiwei Pan, Pengcheng Shi, Haiyan Lv, and Ting Ma. 2021. Uncertainty Quantification in Medical Image Segmentation with Multi-decoder U-Net. https://doi.org/10.48550/ARXIV.2109.07045

[24] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-second AAAI conference on artificial intelligence*.

PACMAN – Physics-Aware Control MANager for HVAC

Srinarayana Nagarathinam, Yashovardhan S. Chati, Malini Pooni Venkat, Arunchandar Vasan
TCS Research
IIT-Madras Research Park, Chennai, 600113, India

ABSTRACT

Indoor user thermal comfort mainly depends upon air temperature, humidity, and wall temperature. Commonly used proportional-integral-derivative (PID) controllers use air temperature alone as feedback to control building Heating, Ventilation, and Air-Conditioning (HVAC) equipment. This may result in sub-optimal energy consumption or user comfort. Model Predictive Controllers (MPC) improve over PID control, but are limited by the accuracy of the underlying thermal model. Black-box (data-driven) and grey-box (data+domain-driven) thermal models may handle non-linear building thermal dynamics, but require extensive data and may not adhere to physical constraints.

We complement existing works with *PACMAN* – an MPC approach that uses Physics-Informed Neural Network (PINN)-based thermal models to estimate humidity and wall temperature in addition to air temperature. We first adapt PINNs to building HVAC control by overcoming the limitation of the original PINN formulation. Specifically, we build a PINN thermal model that can handle time-dependent exogenous (ambient temperature) and control (setpoint) inputs through a time resetting strategy. We then demonstrate the use of a PINN thermal model in MPC with a receding horizon.

We evaluate *PACMAN* using a simulated environment along two dimensions: 1) thermal model accuracy; and 2) control efficacy. As a baseline for the thermal model, we use a data-driven LSTM model. As baselines for the control, we use an as-is PID controller with fixed and seasonal temperature setpoints; oracle MPC without any thermal model errors; and MPC with an LSTM model. We find that the PINN thermal models improve errors over LSTM by an order of magnitude; and generalize better to out-of-distribution data. *PACMAN* reduces the annual energy consumption by 16% and the percentage of yearly discomfort hours by 26% points over as-is PID control with a fixed setpoint.

CCS CONCEPTS

• **Computing methodologies** → **Simulation evaluation**; • **Applied computing** → **Physics**.

KEYWORDS

physics-informed, neural network, receding horizon, building, HVAC, control, simulation

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564052

ACM Reference Format:
Srinarayana Nagarathinam, Yashovardhan S. Chati, Malini Pooni Venkat, Arunchandar Vasan. 2022. *PACMAN* – Physics-Aware Control MANager for HVAC. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3563357.3564052

1 INTRODUCTION

Buildings contribute significantly to energy consumption and carbon footprints. Because HVAC alone accounts for 40% of building energy consumption [4], energy management for HVAC systems has received significant research attention. Most HVAC deployments currently use PID control that uses only an error between the realized and target (setpoint) air temperatures to operate AHUs and chillers. PID control is widespread due to the ease of implementation in firmware. Therefore, the main focus in HVAC energy management has been to save energy through intelligent control that improves over PID-based as-is control even while meeting occupant comfort requirements [1].

Unmeasured thermal parameters: Intelligence in HVAC control improves over PID control by using additional sensing parameters (e.g., occupancy) [7, 13] and/or through optimisation in control (e.g., model-predictive [11] or reinforcement learning based [29, 37]). Since additional sensory deployments or extensive measurements may not be readily possible, the main focus generally has been on better methods for optimisation in control. However, in practice, some parameters that do not have sensory readings can nevertheless be critical in determining system performance irrespective of the control strategy used. Specifically, consider the surface temperature of the walls of a building. In most deployments, sensors typically exist only for the room air temperature (T_a), room humidity (W_a), supply and return temperatures, but not the wall temperature (T_w). Several research works estimate wall temperature using a functional approximation involving outdoor ambient (T_∞) and indoor air (T_a) temperatures [14, 28] or use T_a as a proxy for T_w in thermal comfort calculations [32]. However, as we show later, using a wrong value for T_w (e.g., with 5% error) can cause percentage discomfort to increase (by 80% points) and energy consumption to increase/decrease (by up to 13%). A correct estimation of such unmeasured critical parameters in the absence of hardware sensing can improve the performance of *any* intelligent control strategy.

Better thermal models for MPC: MPC methods [11] improve over PID control, but are highly dependent on the accuracy of the thermal models used to predict the parameters of interest, namely, T_a, T_w, and W_a. One approach to estimating these parameters could be to use purely data-driven models. However, such models are data-intensive and need extensive training and calibration. Moreover, they are not guaranteed to adhere to physical laws governing system dynamics. In this work, we complement existing approaches

by using PINNs-based models for estimating the thermal parameters. PINNs [34] encode the governing differential equations of the underlying system in the backpropagated gradient of the neural network using automatic differentiation. They thus capture the underlying physics of the model. The governing equations require only readily available data in the form of building material constants such as U-value and thermal capacitance. Therefore, PINNs bypass the need for extensive measurements to learn pure data-driven models of thermal parameters. PINNs also serve as a means to soft sense the unmeasured thermal parameter T_w by modelling the governing equation of T_w. Further, once trained, PINNs are much faster than numerically solving differential equations for modelling the dynamics of the system, particularly if the physics is governed by the Navier-Stokes equations, at every control step as the system state update would simply be a forward pass through a neural network for the next time step.

Gaps in existing works in using PINNs for control: Using PINNs to control building systems is non-trivial for several reasons. **(1)** The basic formalism of PINNs [34] is not readily amenable to control problems. Specifically, exogenous inputs (e.g., ambient temperature) and control decisions (e.g., mass flow rates) need to be specified as *functional* inputs to the neural network, which may not be known a priori. The input feature space could be very high dimensional depending upon the duration (monthly, yearly, etc.) and the (min-max) range of the inputs. **(2)** The efficacy of MPC [21] using PINNs is mostly unexplored. **(3)** To the best of our knowledge, most works on physics-constrained neural networks for building thermal models do not account for evolution in humidity [18, 30], which is an important parameter in thermal comfort calculation.

Our approach: We overcome these challenges and complement existing works by adapting PINNs for building HVAC control while explicitly estimating all critical thermal parameters, including the unmeasured T_w and the often neglected W_a. We present *PACMAN* that employs a PINN-based controller in an MPC framework for HVAC energy minimisation even while meeting thermal comfort requirements. We extend the PINN formalism for control problems with time-varying exogeneous and control inputs through a time resetting strategy and a zero-order hold assumption. *PACMAN recommends temperature setpoints that are used as reference values in the underlying PID controller and therefore does not require altering the implementation in firmware.*

To summarize, our **specific contributions** include the following:

- We contribute to the body of PINN literature by adapting PINNs for modelling the building's thermal dynamics and control of HVAC. As a baseline for PINN, we use a purely data-driven Long-Short Term Memory (LSTM) model for estimating the thermal parameters (T_a, T_w, and W_a).
- We demonstrate the need for a high-accuracy thermal model for MPC using experiments. Specifically, we consider model errors between ±5% in steps of 1% in T_a, T_w, and W_a. The metrics evaluated are the annual HVAC energy consumption and the percentage of HVAC operating hours that occupants' comfort is unmet (acceptable comfort being PPD $\leq$ 10%).
- We implement PINN in a simulated environment using real-world building data, including the associated HVAC design and building material properties. The simulator acts as the ground truth where the governing equations are solved using an ODE integrator (fourth-order Runge-Kutta method RK4). The PID controller is tuned using the Ziegler-Nichols technique [12].
- We study the sensitivity of PINN to the time reset length τ. We compare *PACMAN* 's performance with the following baselines: (BL1) as-is control that uses a fixed temperature setpoint of 23 ± 1^{o}C throughout the HVAC operation, (BL2) control that uses seasonal temperature setpoints of 22 ± 1^{o}C during summer months and 23 ± 1^{o}C for the rest of the year, (BL3) Oracle MPC with receding horizon control that assumes perfect knowledge of the system (zero model error), controller, and exogenous disturbances, and (BL4) MPC with the LSTM model.

Our **key findings** are as follows:

- MPC method is equally sensitive to model errors in T_a and T_w and less sensitive to those in W_a. Model errors affect discomfort by up to 80% points and annual energy consumption by up to 13%. This shows T_w, while unused, is equally important.
- *PACMAN* 's PINN model performs better than the baseline LSTM model. Furthermore, LSTM gives unphysical results for W_a while PINN respects physics at all times. When trained on a limited range of inputs, PINN generalized better than LSTM for out-of-distribution data. LSTM gave larger errors (23, 19, 29)% compared to PINN (3, 2, 3)%.
- In the control problem, *PACMAN* performs better than BL1 and BL2. Over BL1, *PACMAN* saves 16% energy and improves percentage comfort by 26% points. Over BL2 *PACMAN* saves 19% energy and improves percentage comfort by 24% points. Compared with BL3 ('oracle' MPC), *PACMAN* gives nearly the same unmet comfort hours for 6% more energy. Compared with BL4 (MPC + LSTM), *PACMAN* consumes 24% less energy and improves comfort by 2% points.
- Finally, we find that the PINN thermal model is sensitive to the time reset length τ. The training runtime varies from 7.3 hours at $\tau = 15$ minutes to 1.2 hours at $\tau = 60$ minutes. Given a simulation time step of 15 minutes, the thermal model error increases with increasing τ, from (0.4, 0.2, 0.7)% at $\tau = 15$ minutes to (7.8, 2.4, 4.3)% at $\tau = 60$ minutes for the parameters (T_a, T_w, W_a).

Paper outline: Section 2 presents related work. The optimal control problem is presented in Section 3. The solution strategy is discussed in Section 4, along with the algorithms. Section 5 presents the experimental design. The findings are presented in Section 6, followed by limitations and conclusions in Section 7 and Section 8, respectively.

2 RELATED WORK

Studies on optimal control of HVAC in buildings can be broadly grouped into thermal model development; model-assisted and model-free based optimal control; and physics-constrained data-driven approaches.

Thermal models: Several studies have looked at thermal model development, and these can be broadly classified into white-box, black-box, and grey-box models. **(1)** *White-box* models require a complete understanding of the system behaviour and are assumed to be deterministic. Tuning the models to account for uncertainties is non-trivial and time-consuming but also integrating with the building management system for real-time control can be challenging and expensive [8]. **(2)** *Black-box models* use statistical [24]

and neural-network [22] based techniques to capture the underlying non-linear relationships. These techniques require no domain knowledge but a lot of data and do not generalise well for unseen data in the training set [6]. Furthermore, because black-box models do not account for physics, they may give unphysical results leading to unreliable predictive control, as we show in Section 6.1.3.
(3) *Grey-box* models use prior domain knowledge and data to calibrate system dynamics [10]. One widely used grey-box model is the resistance-capacitance (RC) representation [16]. Depending on the complexity, RC forms a system of coupled differential equations that are challenging to solve for real-time optimal control problems [9].
Model-based and model-free : MPC is a commonly used technique to improve building's energy savings and comfort. Cost-saving of 20% was demonstrated in real-world case studies over a naive rule-based control [35]. Black-box-based optimal temperature was considered in [20]. Grey-box based RC model with deterministic and stochastic exogenous variables was used for optimal control in [11] and [33], respectively. The efficacy of MPC is sensitive to model errors, as we show in Section 6.1.1. Recently, model-free agent-based approaches such as deep reinforcement learning are gaining importance due to the challenges involved in developing accurate models for MPC. The agent interacts directly with the environment and, over time, learns an approximate control policy [37]. However, such agent-based methods would require significant exploration/interaction with real-world systems, making them impractical due to various operational constraints [29].
Physics-constrained ML: Methods encoding the prior domain knowledge as part of the neural network architecture were shown in [17, 27] and using custom loss function in [25]. A technique to directly solve the governing differential equation through automatic differentiation of neural networks was considered in [34]. A significant drawback of this method is that accounting for time-dependent exogenous and control inputs is non-trivial since the functional form of input features is required a priori. In [18], multi neural network architecture was proposed to account for exogenous input variables. However, [18] lacked a proper baseline for performance comparison. [30] proposed a recurrent neural network (RNN) architecture together with a physics-constrained formulation. While their approach was 50% more accurate than a purely data-driven method, the accuracy relied heavily on the availability of historical data, which may not always be available, particularly for wall temperature. A recent study, [21], looked at mitigating the issues with the original PINN formulation for control problems by considering the historical logs of the exogenous and control inputs as part of the input feature space and using an encoder-decoder neural architecture. However, [21] did not study the efficacy of the controller using the PINN model. A comprehensive review of informed machine learning can be found in [36].

We propose *PACMAN* - a physics-aware optimal HVAC control manager that complements existing works. We extend the idea of a time resetting strategy and a zero-order hold assumption, first proposed in [31], to a more complex system of coupled ODEs involving both a time-varying ambient temperature and a time-varying control input. In the time resetting strategy, we split the entire time-widow into multiple time slots, each of length τ; and we reset the time to zero at the start of each time slot. Next, we apply the zero-order hold assumption where the ambient temperature and the control input are constant within each time slot. We refer to Section 4.1 for more details. We use the standard lumped model representation for state evolution, and unlike previous studies on PINN, we also account for humidity. Although less important than T_a and T_w, high error in W_a can lead to increase in discomfort hours, as we will show in Section 6.1.1. We also study the sensitivity of PINN model errors and training runtime to the time reset length. Finally, we evaluate the efficacy of MPC with PINN.

Table 1: Notation used.

Symbol	Meaning (units)
Thermal model and optimisation	
t, δt	Time instant and control time step (s)
$\mathcal{H}$	Prediction horizon (s)
T, W	Temperature (°C) and Humidity ratio (gkg^{-1})
MRT	Mean Radiant Temperature (°C)
$\mathbf{T}_{SP}$	Temperature setpoint vector (°C)
C	Thermal capacitance ($J{\cdot}K^{-1}$)
C_P	Specific heat capacity of air ($J{\cdot}kg^{-1}{\cdot}K^{-1}$)
ρ	Air density ($kg{\cdot}m^{-3}$)
V	Room volume (m^3)
E_{HVAC}	Energy consumed by HVAC (W·s)
$\dot{Q}_i, \dot{Q}_L$	Building internal and total cooling load (W)
h	Specific enthalpy ($J{\cdot}kg^{-1}$)
R_i, R_o	Indoor and outdoor thermal resistance ($W^{-1}{\cdot}K$)
$\dot{m}$	Mass flowrate of AHU ($kg{\cdot}s^{-1}$)
$\dot{m}_g$	Building internal moisture generation ($kg{\cdot}s^{-1}$)
$\kappa_p, \kappa_i, \kappa_d$	PID gain constants
e	PID error (°C)
PPD	Predicted Percentage Dissatisfied (%)
v_a	Air speed ($m{\cdot}s^{-1}$)
MET	Metabolic Rate (met)
Clo	Clothing insulation (Clo)
Subscripts and Superscripts	
a, w, ∞, sa	Air, wall, outdoor ambient, and supply air
$\hat{x}$	Approximation of x
t	At time instant t
0	At time instant 0
PINN	
$\mathcal{M}$	Neural network model
τ	Time reset length (s)
θ	Parameters/weights of neural network
α	Learning rate
$\mathcal{L}$	Loss function
$\mathcal{R}$	Residual of Equations 3–5
N_p	# collocation points for ODE residual losses
N_0	# points for initial value losses
EPOCHS	# of training epochs

3 PROBLEM FORMULATION

The main aim of building HVAC equipment is to keep the user's comfort within acceptable limits. The standard metric used to evaluate the user thermal comfort is Fanger's Predicted Percentage Dissatisfied (*PPD*) [19], and $PPD \leq 10\%$ is considered as an acceptable thermal condition [1]. The three important thermal parameters that determine *PPD* are the indoor air temperature T_a, mean radiant temperature MRT, and humidity ratio W_a. The other factors, such

as v_a, MET, and Clo, are usually assumed to be constants. MRT is determined by T_w [3]. While many combinations of T_a, T_w, and W_a may lead to $PPD \leq 10\%$ criteria, they may lead to different energy consumptions. We are interested in a combination that minimises energy. The control knob typically available in the building is the temperature setpoint T_{SP}. Hence, there is a scope to dynamically adjust T_{SP} so as to provide just enough comfort at minimum energy. Notations are summarized in Table 1.

3.1 Optimal control problem

We formally define the optimal HVAC control problem as:

$$\min_{\mathbf{T_{SP}}} \quad \sum_{t}^{t+\mathcal{H}} E_{\text{HVAC}} \tag{1}$$

Here $\mathcal{H}$ is the optimisation horizon; E_{HVAC}, the HVAC energy consumption; and $\mathbf{T_{SP}}$, the temperature setpoint vector over the optimisation horizon. $E_{\text{HVAC}} = \dot{Q}_L/\text{COP}$, where $\dot{Q}_L$ is the cooling load and COP the coefficient of performance of the HVAC chiller. $\dot{Q}_L = \dot{m} \cdot (h_a - h_{sa})$ where h_a and h_{sa} are the specific enthalpies of the room air and supply air, respectively. The specific enthalpy h is a function of air temperature and humidity [2]. We consider compressor energy consumption as the objective function as it accounts for 80% of the total HVAC energy [15].

The constraints to the optimisation problem are as follows:

(1) **Thermal comfort satisfaction [1]:**

$$all(\mathbf{PPD}[t : t + \mathcal{H}]) \leq 10\%, \tag{2}$$

where $\mathbf{PPD}[t : t + \mathcal{H}]$ are the PPD values over the horizon.

(2) **Environment/plant model:** The evolutions of air temperature, humidity ratio, and wall temperature are governed by,

$$\frac{dT_a}{dt} = \frac{\dot{Q}_i}{C_a} + \frac{1}{C_a \cdot R_i}(T_w - T_a) + \frac{\dot{m}C_P}{C_a}(T_{\text{sa}} - T_a), \tag{3}$$

$$\frac{dW_a}{dt} = \frac{\dot{m}_g}{\rho V} + \frac{\dot{m}}{\rho V}(W_{\text{sa}} - W_a), \tag{4}$$

$$\frac{dT_w}{dt} = \frac{1}{C_w \cdot R_i}(T_a - T_w) + \frac{1}{C_w \cdot R_o}(T_\infty - T_w). \tag{5}$$

For simplicity, we omit the terms related to outdoor air infiltration and solar radiation. The internal heat load and moisture generation depend on the room occupancy, electrical fixtures, plants in the room, etc. We assume the internal heat load to be constant in this work, corresponding to a constant room occupancy. Note that these assumptions are not limitations of *PACMAN* and can be relaxed in a more general setting.

(3) **AHU mass flow rate evolution model:** The evolution of AHU mass flow rate ($\dot{m}$) is governed by the HVAC PID control logic and is of the form,

$$\dot{m}^t = \dot{m}^{t-\delta t} + \kappa_p \cdot e^t + \kappa_i \cdot \sum e^t \delta t + \kappa_d \cdot \frac{e^t - e^{t-\delta t}}{\delta t}, \tag{6}$$

where the error $e^t = T_a^t - T_{SP}^t$.

3.2 Optimal control framework

The closed-loop control is achieved by solving the optimisation problem together with the constraints (Equations 1–6) with a receding horizon technique at every control step using a prediction model of the system. A typical model-based controller interacts with the real-world environment at every control time step, as shown in Figure 1. The controller consists of a thermal model together with an optimiser. The output of the controller is the optimal temperature setpoints for the entire prediction horizon. A building management system (BMS) modulates the AHU fan by translating the setpoint of the very next step to mass flow rate through a PID control logic. The environment moves to a new state, and BMS monitors state variables. At the next control step, the new state variables are given as inputs to the model-based controller and the closed-loop control repeats. *PACMAN* uses a PINN-based thermal model to predict the future states.

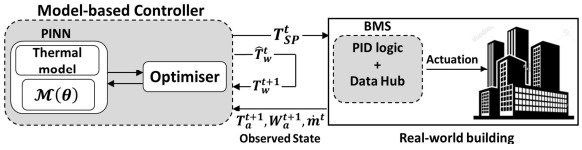


Figure 1: Optimal control framework.

3.3 PINN background

First proposed by [26] and recently made popular by [34], PINNs approximate the solution of the governing physics-based ordinary differential equations (Equations 3 – 5) by adding the ODE residuals to the loss function of a neural network. This approach is useful, particularly when labeled data is limited or unavailable (self-supervising mode). A schematic of the original PINN architecture [34] is shown as a computation graph in Figure 2. We explain the concept of PINN using the following initial-value problem: $\dot{\mathbf{x}}^t = \psi\left(\mathbf{x}^t, \mathbf{u}^t\right)$ given $\mathbf{x(0)} = \mathbf{x^0}$, where $t \in \mathbb{T}$, $\dot{\mathbf{x}}^t$ is the rate of change of $\mathbf{x}^t$, $\mathbf{x^0} \in \mathbb{X}$ (the system state), and $\mathbf{u} \in \mathbb{U}$ (set of control actions). Given the function $\mathbf{u}^t$ and the initial condition $\mathbf{x^0}$, the neural network of PINN takes t as the input. PINN approximates $\mathbf{x^t}$ as $\hat{\mathbf{x}}^\mathbf{t}(\theta)$, where θ is the vector of network weights. The physics-informed intelligence comes from the automatic differentiation of $\hat{\mathbf{x}}^\mathbf{t}$ with respect to the independent input variables (in this case, t). The neural network loss $\mathcal{L}(\theta)$ is the summation of labeled data loss, physics loss (residual of ODE), and initial condition loss, as shown in Figure 2. Interested readers may refer to [34] for more details. In this paper, we only consider the self-supervising mode of the PINN and do not consider any labeled data.

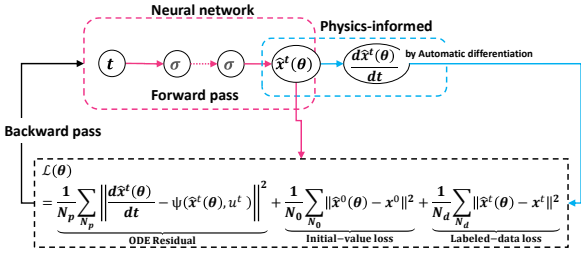


Figure 2: Original PINN computation graph.

3.4 Limitation of original PINN model

Consider a room exposed to a time-varying ambient temperature $T_\infty(t)$ and no cooling control with the following approximate governing equation for the room temperature: $\frac{dT_a}{dt} = K(T_\infty - T_a)$. Even

though $\frac{dT_a}{dt}$ at a point $t{=}t'$ depends upon only the current $T_\infty(t{=}t')$, the solution $T_a(t{=}t')$ at time t' would depend upon *the entire path* of $T_\infty(t)$ over all points t in $[0, t']$ and not just $T_\infty(t{=}t')$. In other words, to correctly solve the system, one needs to know the *function* $T_\infty(t)$ and not the scalar $T_\infty(t = t')$. Because T_a depends upon a time-varying function T_∞, its total derivative can be written as follows:

$$\frac{dT_a}{dt} = \frac{\partial T}{\partial T_\infty} \times \frac{dT_\infty}{dt} + \frac{\partial T_a}{\partial t}$$

In the original PINN formulation, T_∞ is assumed constant with time, so we have $\frac{dT_\infty}{dt}$=0 and so $\frac{dT_a}{dt} = \frac{\partial T_a}{\partial t}$. Note that the auto-differentiation would compute $\frac{\partial T_a(\theta)}{\partial t}$ and so, the *residual* error between $\frac{\partial T_a(\theta)}{\partial t}$ from the neural network; and $\frac{\partial T_a}{\partial t}$ from the governing equation can be minimised as function of θ. For HVAC control, we need to handle a non-zero $\frac{dT_\infty}{dt}$. If we do not do this, the output could be quite different than reality as illustrated in Figure 3 that shows: 1) the driving time-varying ambient temperature; 2) the correct time-varying room temperature; and 3) the incorrect PINN prediction that does not account for time-variation in the ambient temperature.

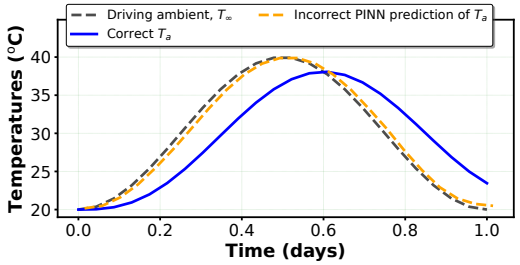


Figure 3: Original PINN model does not handle a time-varying $T_\infty(t)$.

Naive approaches: One approach to handling a time-varying $T_\infty(t)$ would be to specify the function $T_\infty(t)$ as a discretized vector input $\mathbf{T}_\infty$ to the PINN for discrete points t. However, this significantly increases the dimensionality of the neural network input space; and is not scalable with the number of points t. Another approach would be specify $T_\infty(t)$ as a closed analytical form function of t as part of the ODE residual. This may not be feasible in general. Even if it were possible for time-varying $T_\infty(t)$, for model-predictive control, time-varying control inputs (from the optimiser) also need to be handled in exactly the same way as time-varying $T_\infty(t)$ for the PINN to be adapted for HVAC control. Clearly, the time-varying optimal control cannot be specified as a closed analytical form to the PINN as input, because the control would itself be decided through optimisation on the PINNs output.

4 SOLUTION STRATEGY

4.1 Overcoming original PINN limitations

Figure 4 shows the schematic of our approach. We divide the time horizon into multiple time slots of equal length τ (time reset). Within each τ, we assume that the external inputs and the initial values are invariant (zero-order hold assumption). This eliminates the need to know a priori the exact functional variation of the external inputs with time. The original PINN, trained only over the period τ, can now be used with constant initial values and constant external inputs as additional input features. This is because the assumption of constant external inputs within τ results in the partial derivative of the thermal parameters estimated by automatic differentiation to be a correct estimate of the total derivative. Another advantage of the simplification is the reduction in sample space that is computationally tractable. Our PINN modification is a natural candidate for MPC with a receding horizon technique where the control input is constant over the control time step. An advantage of the PINN formulation over the numerical integrators available to solve the ODE (such as RK4) is that we can directly predict the output for any time $t' \in [k\tau, (k+1)\tau]$ with a forward pass without the need to predict at intermediate points within $[k\tau, t']$. As τ is reduced, the prediction given by the PINN approaches the true solution for the thermal parameters given by solving the ODEs for the actual functional inputs.

As the PINN is used with MPC in a receding horizon technique, the predicted outputs at t's such as $\hat{x}^{t+1}, ..., \hat{x}^{t+m}$, which become inputs at their respective following control step, are replaced with the actual observed values from the environment. This reduces error accumulation across control time steps. Only those parameters that are sensed in the BMS (such as T_a, W_a, T_∞, and $\dot{m}$) are directly fed back to the prediction model. T_w is usually not measured and thus needs to be approximated. For this purpose, we use $\hat{T}_w$ predicted by the model at the current time step t as a soft-sensed input T_w^{t+1} to the next control step, as shown in Figure 1.

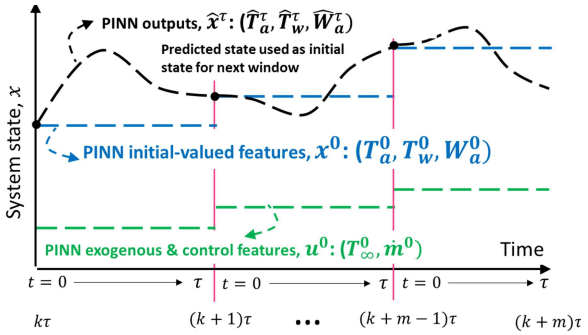


Figure 4: Schematic of our PINN implementation.

4.2 PINN thermal model training

Based on the time resetting and zero-order hold assumption, the following six input features are chosen for our PINN formulation:

(1) Time instant $t \in [0, \tau]$
(2) T_∞^t assumed constant over $[k\tau, (k+1)\tau]$ $\forall k$
(3) $\dot{m}^t$ assumed constant over $[k\tau, (k+1)\tau]$ $\forall k$
(4) The initial value T_a^0 at the beginning of each time slot
(5) The initial value T_w^0 at the beginning of each time slot
(6) The initial value W_a^0 at the beginning of each time slot

The three output variables of the PINN are $\hat{T}_a^t$, $\hat{T}_w^t$, and $\hat{W}_a^t$, for any time t given as the input.

We present the outline of our self-supervised (without labeled data) PINN training in Algorithm 1. First, we initialize the hyper-parameters: learning rate α, EPOCHS, N_p, N_0, and τ in Line 1 to Line 5. Then we sample N_p collocation examples of (t, T_∞, $\dot{m}$, T_a^0, T_w^0, W_a^0) and N_0 initial-value examples of (0, T_∞, $\dot{m}$, T_a^0, T_w^0,

Algorithm 1: PINN training.

Hyperparameters:

1 α // Learning rate of neural network
2 EPOCHS // Number of iterations

Inputs:

3 N_p // Number of collocation points for physics loss term
4 N_0 // Number of points for initial-value loss term
5 τ // Time reset length
6 $\mathcal{X}_p \leftarrow$ Sample$[(t, T_\infty, \dot{m}, T_a^0, T_w^0, W_a^0), N_p]$ // Draw samples for physics loss
7 $\mathcal{X}_0 \leftarrow$ Sample$[(0, T_\infty, \dot{m}, T_a^0, T_w^0, W_a^0), N_0]$ // Draw samples for initial value loss
8 $\mathcal{Y}_0 \leftarrow \mathcal{X}_0[:, (T_a^0, T_w^0, W_a^0)]$
9 Randomly initialize model $\mathcal{M}$ parameters θ
10 **for** $e = 1$ to EPOCHS **do**
11 $(\hat{T}_a(\theta), \hat{T}_w(\theta), \hat{W}_a(\theta)) \leftarrow$ Evaluate $\mathcal{M}(\mathcal{X}_p)$
12 $\hat{\mathcal{Y}}_0 \leftarrow$ Evaluate $\mathcal{M}(\mathcal{X}_0)$ // Forward pass
13 $(\frac{d\hat{T}_a(\theta)}{dt}, \frac{d\hat{T}_w(\theta)}{dt}, \frac{d\hat{W}_a(\theta)}{dt}) \leftarrow$ AutoDiff $(\hat{T}_a, \hat{T}_w, \hat{W}_a)$ // Automatic differentiation
14 $\mathcal{R}_{T_a}, \mathcal{R}_{W_a}, \mathcal{R}_{T_w} \leftarrow$ Obtain residuals using Lines 11 and 13 for Equations 3, 4, 5 as functions of θ
15 $\mathcal{L}_{\text{phy}} \leftarrow \frac{1}{N_p}||\mathcal{R}_{T_a}||^2 + \frac{1}{N_p}||\mathcal{R}_{T_w}||^2 + \frac{1}{N_p}||\mathcal{R}_{W_a}||^2$
16 $\mathcal{L}_{\text{init}} \leftarrow \frac{1}{N_0}||\hat{\mathcal{Y}}_0 - \mathcal{Y}_0||^2$
17 $\mathcal{L}(\theta) \leftarrow \mathcal{L}_{\text{phy}} + \mathcal{L}_{\text{init}}$
18 Evaluate gradients, $\nabla_\theta \mathcal{L}(\theta)$
19 Update parameters, $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}(\theta)$
20 **end for**
21 **Return** $\mathcal{M}(\theta)$

W_a^0) in Lines 6 – 8 for evaluating physics and initial-value losses, respectively. We use the Latin Hypercube Sampling (LHS) technique, which has better sampling coverage than standard random sampling [38], to sample the input features from their respective ranges. We initialize the network parameters/weights θ in Line 9. In the main iteration loop (Line 10), in Lines 11 and 12, we do a forward pass of the neural network to get an estimate of the thermal model parameters at $\mathcal{X}_P$ and $\mathcal{X}_0$. We evaluate the derivatives of the estimates of temperatures and humidity ratio with respect to t at Line 13 using automatic differentiation (as functions of θ). Next, we plug in the model estimates and their gradients along with the external and control inputs in Equations 3–5 to calculate the ODE residuals (difference between the left hand and right hand sides of the ODE); this is shown in Line 14. The physics loss $\mathcal{L}_{\text{phy}}$ is calculated as the sum of the individual mean squared ODE residuals (Line 15). The initial value loss $\mathcal{L}_{\text{init}}$ is calculated as the mean squared error between $\hat{\mathcal{Y}}_0$ (estimated in Line 12) and $\mathcal{Y}_0$ (sampled in Line 8). The total loss function $\mathcal{L}(\theta)$ is the sum of the physics loss and the initial-value loss. The gradients of the loss function with respect to the network parameters are evaluated in Line 18. Finally, the network parameters are updated using the standard gradient descent algorithm in Line 19. Though not shown here, loss contributions from each thermal parameter could be weighted differently in order to improve the performance of the trained PINN.

4.3 PINN in control

Algorithm 1 returns the trained PINN model $\mathcal{M}(\theta)$, which is used as the thermal model in the optimal control framework, as shown in Figure 1 and detailed in Section 3.2.

5 EXPERIMENTAL SETUP

We design experiments to evaluate: **(1)** the accuracy of PINN thermal model in isolation; and **(2)** the efficiency of PINN when used in *PACMAN* control. For the former, we use LSTM as baseline. For the latter, we use different controls as baselines.

5.1 Thermal model evaluation

Ground truth: We simulate the environment using a numerical integration of the thermal model (Equations 3–5). Specifically, we use the fourth-order Runge-Kutta method (RK4) [5], which is an accurate and widely used method for solving initial-value problems governed by first-order ODEs. The simulation time step is 15 minutes. The constants used in the thermal model (Equations 3–5) are summarised in Table 2. Note that $\dot{m}_g$ in Equation 4 varies dynamically since it is a function of the latent load (a constant) and latent heat of vaporisation (a function of T_a). We also simulate a PID control logic (Equation 6), the gain constants of which are tuned using the Ziegler-Nichols methods [12]. Furthermore, the PID control logic (Equation 6) usually operates within a temperature dead-band, that is, $\dot{m}$ is not adjusted if $T_a \in [T_{\text{SP}} \pm \text{dead-band}]$. We also implement an anti-windup logic where the integral error is not propagated if $T_a \in [T_{\text{SP}} \pm \text{dead-band}]$.

Table 2: Thermal model parameter list. All values are in SI units unless otherwise specified. Temperatures are in °C.

Parameter	Value	Parameter	Value
$\dot{Q}_i$	5000	C_a, C_w	50e6, 306e6
R_i, R_o	3.2e-5, 1.1e-5	ρ, C_P	1.2, 1005
$T_{\text{sa}}, W_{\text{sa}}$ (g·kg^{-1})	13, 8	Latent load	1250
$\dot{m}$	$\in [0., 3.]$	T_{SP}	$\in [15., 30.]$
$\kappa_p, \kappa_i, \kappa_d$	0.1, 0.0001, 0.	PID dead-band	±1

PINN specific details: Theoretically, $\tau \geq$ control time step. The time reset parameter τ is taken 15 minutes. Depending on the choice of τ, there will be a trade-off between computation speed and accuracy, as we show in Section 6.1.4. The input features, which are the time instant t, temperatures T_∞, T_a^0, T_w^0, humidity ratio W_a^0, and mass flow rate $\dot{m}$ are sampled from $[0, \tau]$, [14.5, 39.5] °C, [5., 16] g·kg^{-1}, and [0., 3.] kg·s^{-1}, respectively. The neural network loss $\mathcal{L}(\theta)$ is the sum of physics loss $\mathcal{L}_{\text{phy}}$ (ODE residuals) and initial value loss $\mathcal{L}_{\text{init}}$ (refer to Figure 2 and Algorithm 1). We have used *Tanh* activation function for all hidden layers as it works better than other activation functions such as *sigmoid* and *ReLU* for problems involving approximating non-linear governing ODEs/PDEs [23]. The number of iteration EPOCHS = $2M$.

LSTM specific details: The labeled temporal examples of T_∞, T_a, W_a, and $\dot{m}$ required for LSTM model are sampled from the environment observations. Note that, although T_w is usually not measured, to make a fair comparison with PINN, we also use T_w temporal examples as an input to LSTM. The sampling frequency is 15 minutes. We implement LSTM using the standard Tensorflow-Keras function

calls. Unlike PINN, LSTM uses only the mean-squared-error (MSE) between predicted and labeled values as a loss function.

Table 3 summarizes a few of the hyperparameters used in the training of both the PINN and LSTM models. We consider a typical annual ambient profile of a building in a tropical climatic region where $T_\infty \in [14.5, 39.5]^\circ$C.

Table 3: PINN and LSTM common hyperparameters.

Hyper-parameter	Value	Hyper-parameter	Value
Size of input layer	6	# hidden layers	2
# nodes per hidden layer	20	Size of output layer	3
Hidden layer activation	Tanh	Optimiser	Adam
Output layer activation	Linear	α	1e-4
Input normalisation	*Min-Max*, [-1, +1]		

Performance metric for thermal model evaluation: We evaluate the efficacy of thermal models using the average relative error between the predicted and the ground truth values for T_a, T_w, W_a.

5.2 Control evaluation

Recall that the decision variable for optimal control is the indoor air temperature setpoint vector $\mathbf{T}_{sp}$, which is used as a reference value in the PID control logic. In this work, we assume complete knowledge of the PID control logic. We consider the following methods to evaluate the control efficacy of *PACMAN* .

(1) **BL1 (As-is):** We assume a constant setpoint of $T_{SP} = 23 \pm 1^\circ$C throughout the HVAC operation. This is the most commonly observed control strategy implemented in buildings.
(2) **BL2 (Seasonal T_{SP}):** We consider a constant setpoint of $T_{SP} = 22 \pm 1^\circ$C during the summer months and $T_{SP} = 23 \pm 1^\circ$C during the rest of the year. This seasonal choice of setpoints is expected to increase occupants' comfort during the warmer months.
The following two baselines and *PACMAN* undertake model-based optimal setpoint determination in a receding horizon control framework as explained in Section 4. The control time step δt and the prediction horizon $\mathcal{H}$ have the same length of 15 minutes. These control techniques differ only in the model used for predicting the thermal parameters over the prediction horizon.
(3) **BL3 ('oracle' MPC)**: We assume complete knowledge of the system and external disturbances of the thermal parameters as seen by the environment. Though unrealistic, this baseline quantifies the best that can be achieved. The thermal model used is the ground truth RK4.
(4) **BL4 (MPC + LSTM)**: We use the LSTM-model (as described in Section 5.1) to predict the thermal parameters over the optimisation horizon. The PINN thermal model in Figure 1 is replaced by the LSTM model.
(5) ***PACMAN* :** We use PINN as thermal model in optimal control framework (Figure 1). The time reset length τ is the same length as the control step, that is, $\tau = 15$ minutes.

Performance metrics for control evaluation: We consider two metrics: **(1)** The annual HVAC energy consumed in MWh, **(2)** The percentage of HVAC operation time that the comfort is unmet, that is, $PPD > 10\%$. Lower energy and unmet comfort hours indicate a better control strategy. In the PPD calculation, we fix v_a, *MET*, and *Clo* at 0.1 m/s, 1.2 met, and 0.5 Clo, respectively.

6 RESULTS

6.1 Thermal model evaluation

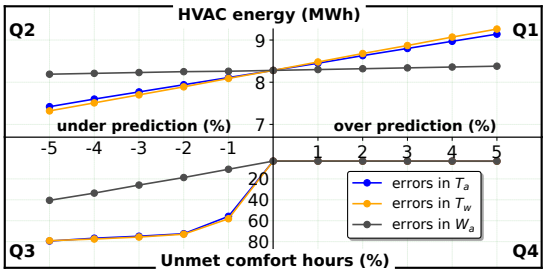
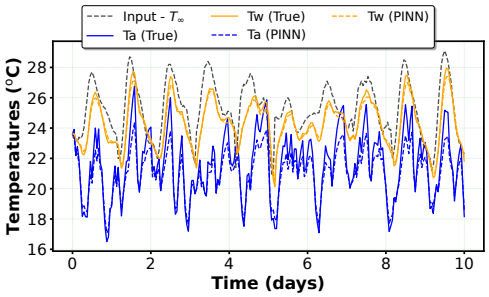


Figure 5: Sensitivity of optimal control solution to model errors.

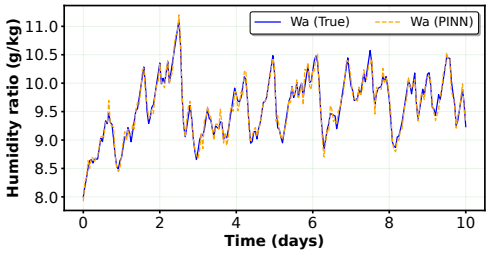
6.1.1 **Need for accuracy in thermal model.** Figure 5 shows the sensitivity of the optimal control solution to model errors. The X-axis represents the percentage of errors injected into the thermal model. The top Y-axis (Q1-Q2) is the annual HVAC energy consumed in MWh, and the bottom Y-axis (Q3-Q4) is the percentage unmet comfort hours. We consider the following model errors here: ±1, ±2, ±3, ±4, and ±5%. We use positive errors to evaluate the model's over-predictions and vice versa. Errors are introduced in one parameter at a time. Specifically, when an error is introduced in T_a, the model errors in T_w and W_a are kept at zero. We infer the following from Figure 5: **(1)** The optimal control solution (energy and unmet comfort hours) is more sensitive to errors in T_a and T_w compared with W_a. **(2)** Thermal parameters T_a and T_w are equally important. **(3)** The HVAC energy varies almost linearly with errors in T_a and T_w because the HVAC energy is modelled as a function of enthalpy, which is a linear function of T_a. Note that errors in T_w affect the enthalpy through T_a (T_a and T_w are connected via the governing equations). **(4)** An interesting trend is observed in the unmet comfort hours. When the model over-predicts thermal parameters, the unmet hours remain nearly unchanged. However, when the model under-predicts the parameters, particularly T_a and T_w, the unmet hours increase non-linearly. This behaviour is explained as follows. For cooling, a higher temperature setpoint leads to a lower HVAC energy and vice versa. The thermal comfort is bounded by $PMV \pm 0.5$, which is equivalent to $PPD \leq 10\%$. During over-prediction (under-prediction), the model picks T_{SP} closer to lower (higher) value of T_{SP}^{opt}. Because the optimiser is trying to minimise energy, it picks a higher T_{SP} among the acceptable values, and the chosen T_{SP} during over-prediction has more cushion to the thermal comfort boundaries compared with T_{SP} picked during under-prediction.

For model errors in the range of (-5% to +5%) in T_a and T_w, the percentage change in the energy from the 'oracle' model (zero errors) is -11% to +13% and the unmet comfort hours vary from 80% to 3%. For model errors in W_a in (-5% to +5%), the energy changes are marginal from -1.1% to +1.2% while the unmet hours vary from 40% to 3%. *These experiments demonstrate the need for an accurate thermal model.*

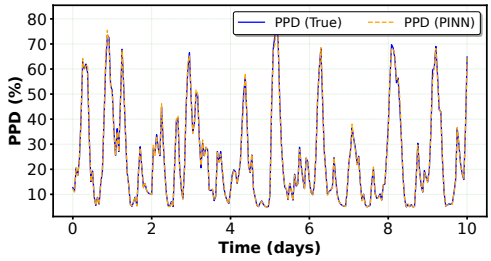
6.1.2 **Prediction accuracy of PINN vs. ground truth.** Figure 6 shows the thermal parameter profiles obtained by PINN and compares the solution with the ground truth. The X-axis represents the time in days, and the Y-axis represents the three primary thermal parameters, namely T_a, T_w (Figure 6a), W_a (Figure 6b), and a derived metric *PPD* (Figure 6c). We observe that PINN compares well with the ground truth data (the solid and dashed lines almost overlap) and gives 0.4%, 0.2%, and 0.7% average errors in T_a, T_w, and W_a over the annual profiles (we have zoomed to 10 days time window in Figure 6 to highlight the goodness of the predictions). Because PPD is a non-linear function of T_a, T_w, and W_a, a slightly higher error of 2.2% is noted in PPD.



(a) PINN temperature model vs true temperature



(b) PINN humidity model vs true humidity



(c) PPD predicted vs true PPD

Figure 6: Comparison of PINN solution with ground truth.

Figure 7 shows the reduction in ODE residuals for each of the three differential equations (ϕ_{T_a}, ϕ_{T_w}, and ϕ_{W_a}) along with the total loss ϕ_t. We observe a smooth decrease in the loss for all physical parameters of interest with training.

6.1.3 **PINN vs purely data-driven models.** The temperature and humidity ratio predictions with LSTM are shown in Figure 8. The temperature predictions are zoomed to 10 days, while the humidity ratio predictions are shown for the entire year to demonstrate unphysical solution. The LSTM model gives 5%, 3%, and 7% errors in T_a, T_w, and W_a, respectively, while an error of at most 1%

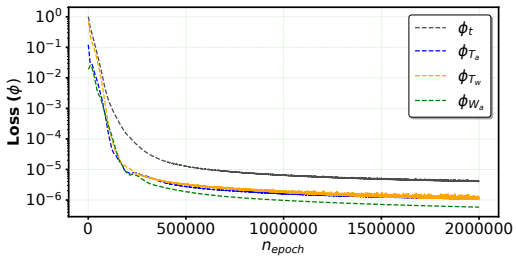
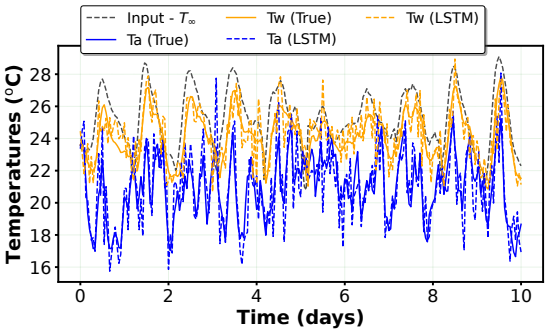
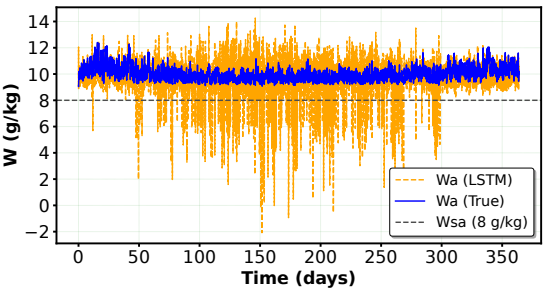


Figure 7: PINN convergence.

for T_a and T_w is needed to avoid severe discomfort. Furthermore, LSTM is a purely data-driven model and so may result in unphysical results, as shown in Figure 8b. The humidity ratio is observed to go below the supply value W_{sa} = 8 g/kg and is also seen to take negative values, which is unphysical. Because the humidity is added to the room by the occupants in our experiments, the resulting W_a cannot go below W_{sa} for the operating conditions we consider. The poor performance of LSTM was also noted for a building heating and cooling problem in [30]. *PINN has the required accuracy and does not violate any physical constraints on the parameters due to its underlying physics modelling.*



(a) LSTM temperature model vs true temperature



(b) LSTM humidity model vs true humidity

Figure 8: Comparison of LSTM solution with ground truth.

Generalizability: Any data-driven model would need to train over a wide range and a large number of sample values of the physical parameters to generalize well at unseen input values. However, since PINN is physics-constrained, it can learn the underlying physical process even with limited data and generalize better than a purely data-driven approach. We demonstrate generalizability by training both LSTM and PINN with summer months data, where the ambient temperature values are sampled from 28 – 40°C. Next, we use the trained models to predict the solution at out-of-distribution ambient

temperatures from 15 − 25°C. Figure 9 compares training and generalization (test) errors for LSTM and PINN. We observe that LSTM generalizes poorly over the out-of-distribution test dataset for all the thermal parameters, whereas *PINN generalizes well over the out-of-distribution test dataset* by learning the underlying physics. Although better than LSTM, PINN generalization errors on out-of-distribution samples are still higher than the desired errors (< 1%). The finding is consistent with [26], where PINN was shown to have higher generalization error when the samples deviated significantly from the samples used in training.

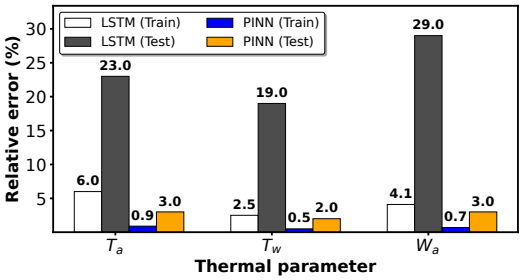


Figure 9: Generalization error: LSTM vs PINN.

6.1.4 **Sensitivity of PINN to τ.** Figure 10 studies the sensitivity of PINN to the time reset length τ. The X-axis represents τ. The primary Y-axis represents the PINN's CPU runtime during training and the secondary Y-axis is the thermal model error. We considered τ = 15, 30, 45, and 60 minutes. As expected, the runtime drops

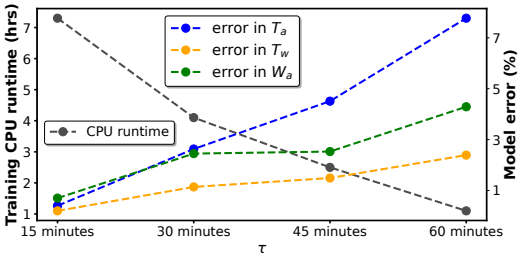


Figure 10: Sensitivity of CPU runtime and model errors to τ.

sharply from 7.3 hours at τ = 15 minutes to 1.2 hours at τ = 60 minutes. Because the inputs to the model are held constant over the time reset window, we expect that the smaller the τ is, the lower is the model error. In Figure 10, we observe that the errors for all the thermal parameters progressively increase with increasing τ, as expected.

6.2 Control evaluation of *PACMAN*

Figure 11 compares *PACMAN* 's control performance with that of the baselines. The numerals on top of the bar chart for BL1 indicate the raw energy consumption in MWh and the percentage of unmet comfort hours. For BL2 through *PACMAN*, the numerals indicated are percentage relative changes for energy and percentage point changes for the percentage unmet comfort hours compared to BL1. The as-is control (BL1) uses a fixed setpoint of $T_{SP} = 23 \pm 1$°C throughout the HVAC operation and consumes approximately 10.3 MWh of annual energy with 29% annual unmet comfort hours (that is, for 29% of the year the PPD > 10%). BL2 uses a seasonal

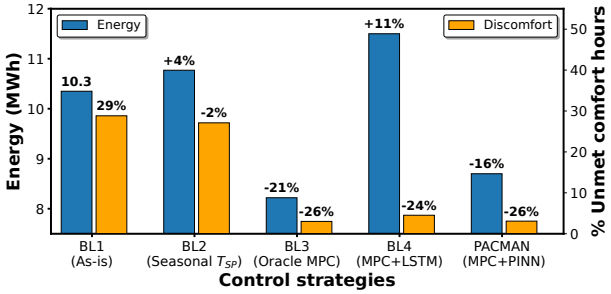


Figure 11: Comparison of *PACMAN* 's performance with baselines. The numerals over the orange bars (*from BL2 to PACMAN*) are % point changes for the percentage unmet comfort hours over BL1.

setpoint strategy, that is, $T_{SP} = 22 \pm 1$°C during summer months and $T_{SP} = 23 \pm 1$°C during the rest of the year. BL2 reduces the percentage unmet comfort hours by a marginal 2% points while consuming 4% more energy over BL1 due to using a lower setpoint during the summer months. The 'oracle' MPC (BL3) limits the maximum possible energy reduction while reducing unmet comfort hours. BL3 results in 21% energy savings and reduces percentage unmet comfort hours by 26% points over BL1. In BL4, the thermal model of MPC is represented by LSTM. Recall that LSTM may result in unphysical results particularly for W_a prediction. For the sake of control with LSTM, we lower bound W_a to an acceptable range. While BL4 reduces the percentage unmet comfort hours by 24% points compared to BL1, it consumes 11% more energy. Finally, *PACMAN* outperforms all other baselines except BL3 (the 'oracle' controller) in reducing energy and unmet comfort hours. *PACMAN* reduces energy by 16% and percentage unmet comfort hours by 26% points over BL1. *PACMAN* 's unmet comfort hours are comparable to BL3, albeit by consuming 6% more energy than BL3. This shows that PINNs can help to improve control optimisation by improved modelling.

7 LIMITATIONS

We highlight some limitations of our approach, which can guide future research. **(1)** All the thermal parameters and exogenous inputs are assumed to be deterministic. However, any real-world system will be affected by noise, and the effect of this noise on the controller's performance needs to be investigated. **(2)** We have assumed complete knowledge of the constants (such as thermal resistance and capacitance) in the governing thermal equations. In reality, these may be unknown and need to be inferred, for which, again, some labeled data are required. PINNs have been shown to work in this inverse problem of calibrating equation parameters using data [34]. We have also assumed a complete knowledge of the HVAC PID controller that maps the setpoint to the HVAC supply mass flow rate. In reality, the controller logic may be proprietary to the manufacturer and may need to be approximated. **(3)** As this paper focuses on the PINN model in a control framework, the prediction horizon is kept equal to the control time step for simplicity of optimisation. However, the horizon is usually longer than the control time step, making the optimisation more challenging. More powerful optimal setpoint determination methods based on Reinforcement Learning can be explored. Such techniques have been

shown to work in real-life stochastic settings too. **(4)** The lumped thermal model considered in this paper treats the thermal parameters to be homogenous in the entire room. However, in practice, the parameters may be non-uniform and this non-uniformity must be accounted for better thermal comfort quantification. Besides, the true advantage of PINN comes from solving the governing partial differential equations such as the Navier-Stokes equations to model the spatial variations in parameters of interest. We propose to look at this problem and the computational benefit that can be achieved in real-time control as part of our future work.

8 CONCLUSION

We proposed *PACMAN* to solve the optimal HVAC control problem in buildings that minimises energy while meeting the user comfort. We used a physics-aware neural network framework to model the thermal dynamics. We reformulated the original PINN to make it suitable for optimal control problems. Specifically, we used a time resetting strategy and a zero-order hold assumption that helped negate the functional specification requirement for the exogenous and control inputs and reduce the sampling space. We showed that physics-aware architecture helps better predict the thermal parameters than a purely data-driven model (LSTM), and the solution obeys physical laws. PINNs resulted in an error < 1% for all thermal parameters. Our evaluation showed the need for high thermal model accuracy in MPC. Our approach performed better than most baselines, next only to 'oracle' MPC, and saved 16% energy and reduced percentage unmet comfort hours by 26% points compared with the as-is control strategy. Finally, we showed that PINNs are sensitive to the time reset length; increasing the time window degraded the solution because the approximation of holding the input variables constant over a longer time window introduces higher model errors. Our future work includes evaluating the utility of PINNs in modelling spatial variations in thermal parameters and real-time control benefits over a full computational fluid dynamics-based approach.

REFERENCES

[1] ASHRAE standard 55: Thermal environmental conditions for human occupancy. ASHRAE, Atlanta, 2010.

[2] Psychrometrics (equations). Built Environment Research,Civil, Architectural and Environmental Engineering Illinois Institute of Technology, US, 2017.

[3] *ISO 7726: 2001 Ergonomics the thermal environment. Instruments for measuring physical quantities.* 2001.

[4] *Buildings Energy Data Book.* 2011.

[5] M. L. Abell and J. P. Braselton. 2 - first-order ordinary differential equations. In *Differential Equations with Mathematica (Fourth Edition)*, pages 45–131. Academic Press, Oxford, fourth edition edition, 2016.

[6] Z. Afroz, G. Shafiullah, T. Urmee, and G. Higgins. Modeling techniques used in building hvac control systems: A review. *Renewable and Sustainable Energy Reviews*, 83:64–84, 2018.

[7] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management for smart building automation. In *Proceedings of BuildSys*, pages 1–6. ACM, 2010.

[8] K. Arendt, M. Jradi, H. Shaker, and C. Veje. Comparative analysis of white-, gray- and black-box models for thermal simulation of indoor environment: Teaching building case study. In *Proceedings of IBPSA-USA*, pages 173–180, 2018.

[9] J. Arroyo, F. Spiessens, and L. Helsen. Identification of multi-zone grey-box building models for use in model predictive control. *Journal of Building Performance Simulation*, 13(4):472–486, 2020.

[10] P. Bacher and H. Madsen. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7):1511–1522, 2011.

[11] A. Beltran and A. E. Cerpa. Optimal HVAC building control with occupancy prediction. In *Proceedings. of BuildSys*, pages 168–171. ACM, 2014.

[12] V. Bobál, J. Macháček, and R. Prokop. Tuning of digital pid controllers based on ziegler - nichols method. *IFAC Proceedings Volumes*, 30(21):145–150, 1997.

[13] J. Brooks, S. Goyal, R. Subramany, Y. Lin, T. Middelkoop, L. Arpan, L. Carloni, and P. Barooah. An experimental investigation of occupancy-based energy-efficient control of commercial building indoor climate. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 5680–5685. IEEE, 2014.

[14] T. Chaudhuri, Y. C. Soh, S. Bose, L. Xie, and H. Li. On assuming mean radiant temperature equal to air temperature during pmv-based thermal comfort study in air-conditioned buildings. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7065–7070, 2016.

[15] E. Cheng. Dynamic chiller plant optimization. ATAL Building Services Engineering Ltd, 2018.

[16] R. D. Coninck, F. Magnusson, J. Åkesson, and L. Helsen. Toolbox for development and validation of grey-box building models for forecasting and control. *Journal of Building Performance Simulation*, 9(3):288–303, 2016.

[17] F. Djeumou, C. Neary, E. Goubault, S. Putot, and U. Topcu. Neural networks with physics-informed architectures and constraints for dynamical systems modeling. *arXiv*, 10.48550/ARXIV.2109.06407, 2021.

[18] J. Drgoňa, A. R. Tuor, V. Chandan, and D. L. Vrabie. Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy and Buildings*, 243:110992, 2021.

[19] P. Fanger. *Thermal Comfort: Analysis and Applications in Environmental Engineering.* McGraw Hill, 1970.

[20] P. Ferreira, A. Ruano, S. Silva, and E. Conceição. Neural networks based predictive control for thermal comfort and energy savings in public buildings. *Energy and Buildings*, 55:238 – 251, 2012.

[21] G. Gokhale, B. Claessens, and C. Develder. Physics informed neural networks for control oriented thermal modeling of buildings. *Applied Energy*, 314, 2022.

[22] H. Huang, L. Chen, and E. Hu. A neural network-based multi-zone modelling approach for predictive control system design in commercial buildings. *Energy and Buildings*, 97:86–97, 2015.

[23] A. Jagtap, K. Kawaguchi, and G. Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2019.

[24] A. Jain, F. Smarra, and R. Mangharam. Data predictive control using regression trees and ensemble learning. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4446–4451, 2017.

[25] A. Karpatne, W. Watkins, J. S. Read, and V. Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *ArXiv*, abs/1710.11431, 2017.

[26] I. Lagaris, A. Likas, and D. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[27] M. Lutter, C. Ritter, and J. Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *7th International Conference on Learning Representations (ICLR)*. ICLR, 2019.

[28] S. Nagarathinam, H. Doddi, A. Vasan, V. Sarangan, P. Venkata Ramakrishna, and A. Sivasubramaniam. Energy efficient thermal comfort in open-plan office buildings. *Energy and Buildings*, 139:476–486, 2017.

[29] S. Nagarathinam, V. Menon, A. Vasan, and A. Sivasubramaniam. MARCO - multi-agent reinforcement learning based control of building HVAC systems. In *Proceedings 11th ACM e-Energy*, page 57–67, 2020.

[30] L. D. Natale, B. Svetozarevic, P. Heer, and C. N. Jones. Physically consistent neural networks for building thermal modeling: theory and analysis. *CoRR*, abs/2112.03212, 2021.

[31] J. Nicodemus, J. Kneifl, J. Fehr, and B. Unger. Physics-informed neural networks-based model predictive control for multi-link manipulators. *arXiv*, 10.48550/ARXIV.2109.10793, 2021.

[32] B. Olesen and K. Parsons. Introduction to thermal comfort standards and to the proposed new version of en iso 7730. *Energy and Buildings*, 34(6):537–548, 2002.

[33] Parisio, Alessandra and Varagnolo, Damiano and Risberg, Daniel and Pattarello, Giorgio and Molinari, Marco and Johansson, Karl H. Randomized model predictive control for HVAC systems. In *Proceedings of BuildSys*, pages 1–8. ACM, 2013.

[34] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[35] D. Sturzenegger, D. Gyalistras, M. Morari, and R. S. Smith. Model predictive climate control of a swiss office building: Implementation, results, and cost–benefit analysis. *IEEE Transactions on Control Systems Technology*, 24(1):1–12, 2016.

[36] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker. Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[37] T. Wei, Y. Wang, and Q. Zhu. Deep reinforcement learning for building HVAC control. In *Proceedings 54th ACM DAC*, 2017.

[38] H. Xiao, W. Pei, W. Deng, L. Kong, H. Sun, and C. Tang. A comparative study of deep neural network and meta-model techniques in behavior learning of microgrids. *IEEE Access*, 8:30104–30118, 2020.

Accelerate Online Reinforcement Learning for Building HVAC Control with Heterogeneous Expert Guidances

Shichao Xu
Northwestern University
Evanston, USA
shichaoxu2023@u.northwestern.edu

Yangyang Fu
Texas A&M University
College Station, USA
yangyang.fu@tamu.edu

Yixuan Wang
Northwestern University
Evanston, USA
yixuanwang2024@u.northwestern.edu

Zhuoran Yang
Yale University
Evanston, USA
zhuoranyang.work@gmail.com

Zheng O'Neill
Texas A&M University
College Station, USA
zoneill@tamu.edu

Zhaoran Wang
Northwestern University
Evanston, USA
zhaoran.wang@northwestern.edu

Qi Zhu
Northwestern University
Evanston, USA
qzhu@northwestern.edu

ABSTRACT

Building heating, ventilation, and air conditioning (HVAC) systems account for nearly half of building energy consumption and 20% of total energy consumption in the US. Their operation is also crucial for ensuring the physical and mental health of building occupants. Compared with traditional model-based HVAC control methods, the recent model-free deep reinforcement learning (DRL) based methods have shown good performance while do not require the development of detailed and costly physical models. However, these model-free DRL approaches often suffer from long training time to reach a good performance, which is a major obstacle for their practical deployment. In this work, we present a systematic approach to accelerate online reinforcement learning for HVAC control by taking full advantage of the *knowledge from domain experts in various forms*. Specifically, the algorithm stages include learning expert functions from existing abstract physical models and from historical data via offline reinforcement learning, integrating the expert functions with rule-based guidelines, conducting training guided by the integrated expert function and performing policy initialization from distilled expert function. Experimental results demonstrate up to 8.8X speedup over previous DRL-based methods.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Computer systems organization** → **Embedded and cyber - physical systems**.

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564064

KEYWORDS

HVAC control, Reinforcement learning, Deep learning

1 INTRODUCTION

Buildings account for around 40% of the energy consumption in the United States, of which nearly half is by the heating, ventilation, and air conditioning (HVAC) systems [32]. In addition, the operation of HVAC systems significant affects the physical and mental health of building occupants, as people spend around 87% of their time indoors [20, 46], and even higher during the COVID-19 pandemic in recent years [16]. It is thus a critical task to develop effective HVAC control strategies that can maintain a comfortable indoor environment while reducing energy cost [31, 42, 45].

In the literature, there are extensive works of developing *model-based approaches* for HVAC control. For instance, [26] uses RC-networks to model the building thermal dynamics and applies the linear quadratic regulator (LQR) method for controlling the HVAC system. [37] designs a model predictive control (MPC) method to minimize the energy consumption and cost of the building HVAC system combined with a solar power unit. Some other works on model-based approaches can be found in [25, 27, 35, 44]. However, to achieve good performance, these model-based approaches require the development of detailed and accurate physical models, which are often difficult and costly in practice. Thus, there has been significant interest in developing *learning-based, model-free approaches* for HVAC control, in particular those based on deep reinforcement learning (DRL). For example, [41] utilizes the deep Q-learning method for controlling the indoor air flow rate and leverages the EnergyPlus platform [6] for simulation-based training. And various other techniques have also been applied for DRL-based building HVAC control, including Deep Deterministic Policy Gradient (DDPG) [11], Proximal Policy Optimization (PPO) [1], Asynchronous Advantage Actor-Critic (A3C) [51], etc.

However, a major difficulty in adopting DRL-based methods for building HVAC control is that it could take a long time to train the RL agent in practice during building operation. For instance, it may

take more than 100 months of training to reach convergence for the Q-learning based methods in [7, 41], and around 500 months of training for the DDPG algorithm in [12] to converge on a laboratory building model. In [48], DDPG is used for temperature control and energy management, and it takes around 2.4×10^4 months to reach the best performance. In [47], the training time is almost 4×10^4 months in their multi-zone building environment. Clearly, such long training time would make it impossible to adopt DRL in practice for building control. While developing a detailed simulation model (e.g., in EnergyPlus) and conducting the training via simulation may help avoid this issue, the development of the simulation model itself is difficult and costly (in terms of both time and expertise), just as in the model-based methods.

Thus, recently researchers have been trying to improve the training efficiency for DRL-based building HVAC control. In [46], a transfer learning approach is proposed to extract and transfer the building-agnostic knowledge from an existing DRL controller of a source building to a new DRL controller of a target building, and only re-train the building-specific components for the new DRL controller. The work in [23] also leverages transfer learning, but for heat pump control in microgrid. However, the effectiveness of the transfer learning-based methods strongly relies on the similarity between the existing building target building and the transferred building, and may not be feasible when they are not similar [46]. There are also a few studies on the application of offline reinforcement learning for building HVAC control, where historical data on existing controllers are leveraged to train new RL-based controllers. For instance, [36] conducts conservative Q-learning (CQL) to train controllers for maintaining the room temperature setpoint. The problem of such offline RL methods, however, is that the learned agents' performance strong depends on the quality of the historical data. They tend to perform poorly due to the distributional shift between the historical data and the learned policy, and may have limited improvement even with fine tuning via online training [30].

In this work, to address the above challenge in DRL training efficiency, we propose a **unified framework that leverages the knowledge from domain experts in various forms** to accelerate online RL for building HVAC control. This is motivated by the observation that in established domains such as building control, there is extensive domain expertise, represented in various forms such as 1) *abstract physical models* (e.g., RC-networks [24] or ARX models [49]) of building thermal dynamics – they are not accurate enough for enabling training DRL or designing model-based methods with good performance, but nevertheless contain valuable information of building dynamics, 2) *historical data* collected from existing controllers – they may not be able to train DRL controllers with good performance due to distribution shift, but also contain useful information on building behavior, and 3) expert rules that reflect basic policies. We believe that leveraging these domain expertise can help accelerate the online RL process. In particular, our framework first learns *expert functions* from existing abstract physical models and from historical data via offline RL, and then combines those with expert rules to generate an *integrated expert function*, which will then be used to drive online RL with prior-guided learning and policy initialization from expert function distillation. In experiments, our framework is able to significantly reduce the convergence time for DRL training by up to 8.8X, while maintaining similar performance (in terms temperature violation rate and energy cost).

To summarize, our work makes the following contributions:

- We propose a novel framework to accelerate online RL for building HVAC control with heterogeneous expert guidances, including abstract physical models, historical data, and expert rules. These various guidances are unified in our framework via the expert functions.
- We conducted a series of experiments for evaluating the effectiveness of our framework. The results demonstrate that our approach can effectively reduce the DRL training time while maintaining good performance.

The rest of the paper is organized as follows. Section 2 introduces the related literature. Section 3 presents our approach, and Section 4 presents the experimental results. Section 5 concludes the paper.

2 RELATED WORKS

Reinforcement Learning for HVAC Control: Building HVAC control is a critical and challenging problem as it significantly affects both building energy efficiency and occupants' physical and mental health. In traditional model-based approaches, detailed and accurate physical models are needed for control optimization, but are often difficult and costly to develop and slow to run. Such limitations have motivated the exploration of model-free approaches in recent years, particularly those based on deep reinforcement learning [41, 46, 51, 52]. These DRL-based HVAC control approaches leverage a variety of RL algorithms including DQN [41], A3C [51], DDPG [11], PPO [1], etc. For instance, Wei *et al.* [41] convert the building HVAC control into a Markov decision process (MDP) problem and leverage the DQN method to intelligently learn the operation strategy based on offline simulations. Gao *et al.* [11] adopt the neural network to predict occupants' thermal comfort for part of their reward function design, and then apply the standard DDPG algorithm to learn from their building simulation environment. Abrazeh *et al.* [1] develop a real-time digital twin with a PPO-based backstepping controller to maintain the relative humidity and temperature in buildings. However, a major obstacle in applying these DRL-based control algorithms is that they often require *dozens of months or more* for training to reach the desired performance [7, 11, 12]. Such long time is clearly not feasible for direct training during real building operation (i.e., sensing the real building environment and sending the actuation signals to HVAC equipment). It may be possible to avoid this by developing accurate and detailed building models and conducting training via simulations on tools such as EnergyPlus and Modelica [28]-based tools [7], however, this again requires the development of those detailed and costly physical models and somewhat defeats the original purpose of using model-free approaches. Thus, it is critical to improve the efficiency of online RL for HVAC control *without* the development of detailed physical models.

Transfer Learning for HVAC Control: One way to speed up RL is to transfer the learned policy between different buildings. For instance, [46] reduces the DRL training time by re-designing the learning objective and decomposing the neural network to a building-agnostic sub-network and a building-specific sub-network. The

building-agnostic sub-network can be directly transferred from an existing DRL controller of a source building, and only the building-specific sub-network needs to be (re)-trained on the target building. This can reduce the DRL training time from months/years to weeks. [23] utilizes the direct policy transfer between different houses with the same state/action space for heat pump control in microgrids. [50] applies the transfer learning to a PPO-based controller for smart home to reduce the training cost. The main limitations of these approaches is that the effectiveness of the transfer strongly relies on the similarity between the source and the target buildings. When the buildings are not similar or not operating in similar environment, the transfer may have poor performance [46].

Offline Reinforcement Learning: Another way to accelerate online RL is through *offline RL*, by leveraging historical data collected under existing control policies. Recent offline RL works focus on two aspects: offline policy optimization, and offline policy evaluation. The former aims to learn an optimal policy for maximizing a notion of cumulative reward, while the latter is intended to evaluate the accumulated reward (or the value function) of a given policy.

For offline policy optimization in particular, a major challenge is that the agent cannot directly explore the environment. And the error (called extrapolation error [10]) that is caused by selected actions not contained in the historical dataset could occur and propagate during the training. This is one of the reasons that limits the effectiveness of existing offline RL approaches for building HVAC control [36]. The approaches that address this challenge mainly utilize regularization or constraint-based methods to help the policy stay near to the existing actions in the historical dataset. For instance, the batch-constrained Q-learning (BCQ) approach [10] restricts its action space to make the learned behavior similar to the actions in the historical dataset. [17] penalizes divergence between the prior learned from the historical dataset and the Q-network policy using KL-control. [40] learns the policy by filtered behavioral cloning, which utilizes critic-regularized regression to filter out low-quality actions. And other related investigations can be found in [2, 4, 8, 9, 13, 22]. And from the prior experiments, we notice that not all offline RL algorithms can be chosen for building the expert function. The method like TD3+BC [9] may not always provide a good value estimation for the given states, as it only aims to make the learned policy closer to the behavior in the offline dataset and tend to overestimate the Q-value. So in this work, we use historical data as one of the expert guidance and conduct offline RL to build an expert function. We leverage the idea from [21] to estimate the value function from historical dataset because of its effectiveness, by directly setting regularization on the Q-function and generating the Q-value estimation in a conservative way to reduce overestimation.

3 OUR PROPOSED FRAMEWORK

3.1 System Model

We use the building model with the fan-coil system from [7], which is extended from a single-zone commercial building with manipulable internal thermal mass. The internal air is conditioned by an idealized fan coil unit (FCU) system, and the fan airflow rate is chosen from multiple discrete levels $\{f_1, f_2, \cdots, f_m\}$ (which can be viewed as m control actions; f_1 is to turn off the cooling system, and f_m is to run it at full speed.). There are two different working modes in this system: the occupied time (daily from 7 am to 7 pm), and the unoccupied time (rest of the day). The HVAC system will run in a low-power mode during the unoccupied time for the energy-saving purpose (with the cooling system almost turned off). And the setting of comfortable temperature bound is different in these two modes. The system conducts control with a period of Δt. Each training episode contains 2 days data, so there are $\frac{2880}{\Delta t}$ control steps in each episode. Other experiment-related settings can be found in Section 4.1. The system state contains the following elements:

- Current physical time t,
- Indoor air temperature T_t^{in},
- Outdoor air temperature T_t^{env},
- Solar irradiance intensity q_t^{sun},
- Power consumption during the current control interval P_t,
- Outdoor air temperature forecast in the next three control steps $\{T_{t+1}^{env}, T_{t+2}^{env}, T_{t+3}^{env}\}$, and
- Solar irradiance intensity forecast in the next three control steps $\{q_{t+1}^{sun}, q_{t+2}^{sun}, q_{t+3}^{sun}\}$.

One thing to note is that we add one additional variable in the implementation to the system state design, which is the remainder after dividing the current physical time t by $24 * 60 * 60$. This is to help the RL agent figure out the time position within one day (morning, noon, afternoon, etc.), and may help it reach better performance as observed in our preliminary experiments.

3.2 Our Online DRL Framework with Heterogeneous Expert Guidances

As stated in Section 1, to accelerate online DRL for HVAC control, we propose a unified framework that leverages heterogeneous expert guidances including abstract physical models, historical data, and expert rules. Figure 1 shows the overview of our framework design. Specifically, the framework includes the following major components:

- An expert function h_u learned from an expert model. The expert model could be an abstract physical model developed by domain experts (commonly exists in building domain), or in case such physical model is not available, a neural network with its parameters determined from historical data (but different from offline RL; more details later).
- Another expert function h_o learned via offline RL on historical data that was collected using existing controllers.
- An integrated expert function h by combining h_u and h_o as well as expert rules.
- Application of prior-guided learning and policy initialization from expert function distillation based on h.

The detailed flow of our approach is shown in Algorithm 1. Next, we will first explain the underlying DRL algorithm we use, and then introduce the details of each component in our approach to improve the DRL efficiency with heterogeneous expert guidances.

Underlying DRL algorithm: Similarly as in recent works [7, 41, 46], we utilize double Deep Q-learning (DDQN) [39] as the underlying DRL algorithm for our framework and also the baseline

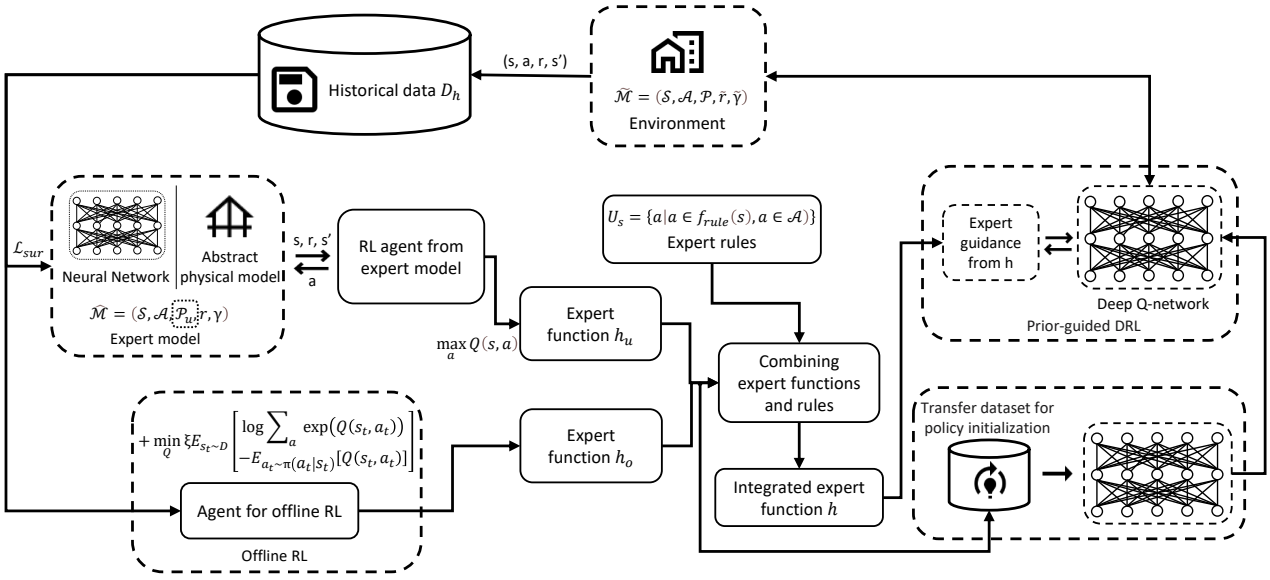


Figure 1: Overview of our online DRL framework with heterogeneous expert guidances. The framework includes the following major components: (1) An expert function h_u learned from an expert model, which can be an abstract physical model or a neural network with its parameters determined from historical data. (2) Another expert function h_o learned from offline RL based on historical data. (3) An integrated expert function h generated by combining h_u and h_o as well as expert rules. (4) Application of prior-guided learning and policy initialization from expert function distillation based on h.

Algorithm 1 Our Online DRL Framework with Heterogeneous Expert Guidances

n_{ep1}, n_{ep2}: number of training epochs
n_{max}: maximum training time of an epoch
n_{tar}: time interval to update target network
Randomly initialize Q-network Q
Learn expert function h_u from expert model using Algorithm 2
Learn expert function h_o from offline RL using Algorithm 3
Generate integrated expert function h from h_u, h_o and expert rules, following Equation (10)
Calculate initialization dataset D_{init}^{y} by h_u, h_o
Train Q-network Q by loss function $\mathcal{L}_{init}$
for $Epoch$ = 1 to n_{ep2} **do**
  Reset building environment Env
  for t = 0 to n_{max} **do**
    Select action a_t using epsilon-greedy
    $s_t, s_{t+1}, r_t \leftarrow Env.execute(a_t)$
    Update $\lambda \leftarrow \lambda_0 + (1 - \lambda_0) \tanh(\alpha_\lambda((Epoch - 1) * n_{max} + t))$
    $\tilde{r}_t = r_t + (1 - \lambda)\gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[h(s')]$, $\tilde{\gamma} = \lambda\gamma$
    Add transition $(s_t, s_{t+1}, a_t, \tilde{r}_t)$ to replay buffer
    Randomly sample a batch $B = (\mathcal{S}, \mathcal{S}', \mathcal{A}, \mathcal{R})$ from replay buffer
    Update Q-network Q with B and $\tilde{\gamma}$
    Update target network Q' with interval n_{tar}
  end for
end for

method for comparison in our experiments. We choose DDQN mainly for its convenience in leveraging the value function and the good performance it has shown for HVAC control in those recent works, but our expert-guidance approach can also be applied to improve the efficiency for many other DRL algorithms.

We assume that the next state of the building HVAC system only relies on the current system state, and thus HVAC control can be treated as a Markov decision process (MDP). As stated in Section 3.1, the state $s = (t, T_t^{in}, T_t^{env}, q_t^{sun}, P_t, T_{t+1}^{env}, T_{t+2}^{env}, T_{t+3}^{env}, q_{t+1}^{sun}, q_{t+2}^{sun}, q_{t+3}^{sun})$. The discrete action space $\mathcal{A}$ contains the normalized air flow rate (0 to 1) with $m - 1$ intervals. The reward is designed with consideration of indoor temperature violation and energy cost, as shown below:

$$r_t = \alpha \cdot \epsilon_t + \beta \cdot c_t, \tag{1}$$

where ϵ_t represents the temperature violation for the current time step, c_t is the energy cost for the current time step, and α, β are the scaling factors. More specifically, ϵ_t is defined as:

$$\epsilon_t = \max(T_i^{in} - T_{upper}, 0) + \max(T_{lower} - T_i^{in}, 0), \tag{2}$$

where T_{upper} is the upper bound of a given comfortable temperature range (which could be based on standards such as ASHRAE [34] or OSHA [33]) and T_{lower} is the lower bound. Moreover:

$$c_t = p_t P_t, \tag{3}$$

where p_t is the energy price at time t, and P_t is the power consumption during the current control interval at time t.

The goal of the DRL is to minimize total energy cost while maintaining indoor temperature within the comfortable temperature range. The loss function $\mathcal{L}_Q$ for updating the Q-network is:

$$\mathcal{L}_Q = \mathbb{E}_{(s_t, a_t, s_t') \sim D}\left[(r_t + \gamma \max_{a_{t+1}} Q'(s_{t+1}, a_{t+1}) - Q(s, a))^2 \right], \tag{4}$$

where $s_t, s_{t+1} \in \mathcal{S}$, $a_t \in \mathcal{A}$, Q is the Q network and Q' is the target Q network. Then, the components introduced in the rest of this section will generate expert functions to provide prior guidance and policy initialization for this underlying DRL algorithm.

Algorithm 2 Learning Expert Function from Expert Model

1: n_{ep1}: number of training epochs
2: n_{max}: maximum training time of an epoch
3: n_{tar}: time interval to update target network
4: Randomly initialize Q-network Q_u
5: Prepare input samples and corresponding labels $\{x^*, y^*\}$ from historical dataset D_h for training an expert model
6: Train expert model Env_u using dataset $\{x^*, y^*\}$ and loss function $\mathcal{L}_u$
7: **for** $Epoch$ = 1 to n_{ep1} **do**
8: Reset building environment Env
9: **for** t = 0 to n_{max} **do**
10: Select action a_t using epsilon-greedy
11: $s_t, s_{t+1}, r_t \leftarrow Env_u.execute(a_t)$
12: Add transition (s_t, s_{t+1}, a_t, r_t) to replay buffer RB_h
13: Randomly sample a batch $B = (\mathcal{S}, \mathcal{S}', \mathcal{A}, \mathcal{R})$ from RB_h
14: Update Q-network Q_u with B and $\tilde{\gamma}$
15: Update target network Q_u' with interval n_{tar}
16: **end for**
17: **end for**
18: Set expert function h_u using Q_u

Learning Expert Function h_u from Expert Model: An expert function h_u can be learned through an expert model. In many cases, such expert model already exists in the form of an abstract physical model for the building thermal dynamics, e.g., an ARX or RC-networks model. While these abstract models are typically not accurate enough to enable good performance for DRL or model-based methods, they can be effectively leveraged to generate an expert function.

If an abstract physical model is not available, we can build a neural network as the expert model, with its parameters decided from historical data collected under existing control policy, as shown in Algorithm 2 (Line 5 in Algorithm 1) and described in the following.

We denote the historical dataset as D_h, with n data samples. For each data sample $(x, y) \in D_h$, let input $x = \{t, T_t^{in}, T_t^{env}, q_t^{sun}, P_t, T_{t+1}^{env}, T_{t+2}^{env}, T_{t+3}^{env}, q_{t+1}^{sun}, q_{t+2}^{sun}, q_{t+3}^{sun}, a\}$ as defined in Section 3.1 and $a \in \mathcal{A}$, and let output label $y = \{T_{t+1}^{in}\}$. The neural network-based expert model consists of m_u fully-connected layers. All hidden layers are followed by a GELU activation function [14], and are sequentially connected (the detailed layer setting will be specified later in Table 1 of Section 4). As different variables may not be in the same order of magnitude (e.g., t can be 1000 times larger than T_t^{in}), we normalize the input x and the output label y. The preprocessed input and output can be written as $x^* = \frac{x - x_l}{x_h - x_l}$, $y^* = \frac{y - y_l}{y_h - y_l}$, where x_h and x_l are the upper bound and lower bound of the variable x, and y_h and y_l are the upper and lower bound of the variable y. We then train the expert model with a mean square error loss function

$$\mathcal{L}_u = \| y^* - y^*_{pred} \|^2, \tag{5}$$

where y^*_{pred} is the network prediction for the normalized y. When we apply this expert model after model training, we obtain the prediction of y by reversing the operation of previous-mentioned normalization step. Note that it may not be necessary to predict the entire system state. For example, the environment temperature T_t^{out} and solar irradiance q_t^{sun} may be obtained from weather forecast.

Once we have the expert model, either in the form of an abstract physical model or a neural network, the expert function h_u can be viewed as a prior guess of the optimal value function in the building HVAC control task and can be learned via DRL. More specifically, we define an MDP problem $\hat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \mathcal{P}_u, r, \gamma)$ where the definitions of state $\mathcal{S}$, action space $\mathcal{A}$ and reward function r are the same as defined at the beginning of Section 3.2. $\mathcal{P}_u$ is from the expert model. We then apply DDQN on $\hat{\mathcal{M}}$ and obtain a trained Q-network Q. And the expert function h_u can be set up as:

$$h_u(s) = \max_a Q(s, a), \tag{6}$$

where s is the state and a is the control action.

Learning Expert Function h_o from Offline RL: Another type of expert function h_o can be learned from the historical data via offline RL, as shown in Algorithm 3 (Line 6 in Algorithm 1). We leverage some of the techniques from conservative Q-learning (CQL) [21] because of its effectiveness in reducing a large number of hyper-parameters.

Algorithm 3 Learning Expert Function from Offline RL

1: n_{ep1}: number of training epochs
2: n_{max}: maximum training time of an epoch
3: n_{tar}: time interval to update target network
4: Randomly initialize Q-network Q_o
5: **for** $Epoch$ = 1 to n_{ep1} **do**
6: **for** t = 0 to n_{max} **do**
7: Randomly sample a batch $B = (\mathcal{S}, \mathcal{S}', \mathcal{A}, \mathcal{R})$ from D_h
8: Update Q-network Q_o with B and $\tilde{\gamma}$ following Equation 8
9: Update target network Q_o' with interval n_{tar}
10: **end for**
11: **end for**
12: Set expert function h_o using the learned Q-networks Q_o

First, we build an offline RL model based on DDQN, but with different Q-network updating rules as the DRL presented in the beginning of Section 3.2. In particular, compared with Equation (4), we add an extra regularization term:

$$\mathcal{L}_{reg} = \min_Q \mathbb{E}_{s_t \sim D} \left[\log \sum_{a_t} \exp(Q(s_t, a_t)) - \mathbb{E}_{a_t \sim \pi(a_t|s_t)} [Q(s_t, a_t)] \right], \tag{7}$$

where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$. Q is the Q-network, and D is the dataset produced by the behaviour policy π. In the equation, the first part $\log \sum_{a_t} \exp(Q(s_t, a_t))$ describes a penalty term for minimizing the Q-value of the action produced by current policy on the states in the historical dataset. It helps learn a smaller and more conservative Q-value estimator. The second term $-\mathbb{E}_{a_t \sim \pi(a_t|s_t)} [Q(s_t, a_t)]$ counts average Q-value in the state-action pairs in the historical dataset and maximizes it to push the current learned policy closer to the behavior policy in the historical dataset.

Then the policy updating is changed as follows:

$$\mathcal{L}_{off} = \frac{1}{2} \mathbb{E}_{(s_t, a_t, s_t') \sim D} \left[(r_t + \gamma \max_{a_{t+1}} Q'(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2 \right] + \xi \mathcal{L}_{reg}, \tag{8}$$

where $s_t, s_{t+1} \in \mathcal{S}$, ξ is a mixing coefficient, and Q' is the target Q-network. With enough training iterations, the offline RL agent can

provide a good expert function h_o following the same procedure as in Equation (6).

Note that we observe that not all offline RL algorithms can be a suitable choice for our framework. For example, approaches like TD3+BC [9] may not always provide a good value estimation for the given states. We suspect that this may be due to two factors. One is related to the reward design, as the value function estimation in some offline RL algorithms is sensitive to the scale of the accumulated reward. The other is that because algorithms like TD3+BC only add regularization on the actor updating and do not set constraints on the Q function, which could enlarge the error in estimating the (Q-)value function when combined with possible numerical issues.

Generating Integrated Expert Function h from h_u, h_o and Expert Rules: The expert function h_u learned from the expert model and the expert function h_o learned via offline RL tend to perform differently because of the complexity of the system dynamic and the sufficiency of the data. Moreover, the accuracy of their Q-value estimation can vary at different states depending on the data distribution within the historical dataset. Thus, it is a natural thought to form an ensemble of the two. And the ensemble of multiple expert functions calculated in different ways can further reduce the overestimation of Q-values through a conservative way, which we will introduce in this section later.

To begin with, after having h_u and h_o, we can combine them with *expert rules* to generate an integrated expert function h. The expert rules are often set by domain experts or building operators based on past experience and domain expertise. They do not provide an optimized control action for a given state, but instead offer suggestions that could be viewed as guidance or soft constraints – e.g., not turning on the cooling system when the indoor temperature is below the lower bound of the comfortable temperature range by certain threshold. Formally, we define that the expert rules f_{rule} can generate an action candidate set $\mathbf{U}_s$ for each state:

$$\mathbf{U}_s = \{a | a \in f_{rule}(s), a \in \mathcal{A})\}. \tag{9}$$

We can then generated an integrated expert function h based on $\mathbf{U}_s$, h_u and h_o (Line 7 in Algorithm 1). Specifically, we apply a pessimistic ensemble strategy for selecting the value function estimation among different expert functions, and only choose corresponding actions from the expert rules' action candidate set $\mathbf{U}_s$. Thus, the integrated expert function h can be formulated as:

$$h(s) = \min_i (\max_{a \in \mathbf{U}_s} Q_i(s, a)), \tag{10}$$

where Q_i is the Q-value estimation from expert functions i. Note that this is a *general formulation* that can unify multiple expert functions – e.g., we may have more than one abstract physical models that provide multiple h_u expert functions.

Prior-guided Learning: Once we have the integrated expert function h, we can use it to guide the underlying DRL with prior-guided learning. There are several algorithms that could guide online RL with a single prior policy, such as HuRL [5] and JSRL [38]. Our framework is flexible in choosing those and we select HuRL [5] in our implementation. In the original HuRL, the Q-value estimation in the RL agent is guided by a simple heuristic function that is learned from the Monte-Carlo regression. In our work, we instead leverage the integrated expert function h from above. By dynamically changing a mixing coefficient λ that controls the trade-off between the bias from the expert function h and the complexity of a reshaped MDP, we are able to accelerate the DRL training with a shortened MDP horizon. Specifically, given the state space $\mathcal{S}$, action space $\mathcal{A}$, reward function r that are mentioned at the beginning of Section 3.2, as well as the transition dynamics of the building HVAC system $\mathcal{P}$ and a discount factor γ, we consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$. We use the learned integrated expert function h as a prior guess for the optimal value function of $\mathcal{M}$. Thus our online DRL can be described as a reshaped MDP $\tilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \tilde{r}, \tilde{\gamma})$, where λ is a mixing coefficient,

$$\tilde{r} = r + (1 - \lambda)\gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[h(s')] \tag{11}$$

and

$$\tilde{\gamma} = \lambda\gamma, \tag{12}$$

which is shown at Line 16 in Algorithm 1.

Policy Initialization from Expert Function Distillation: In the above section, we use the integrated expert function h to reshape the reward function and shorten the MDP horizon. In addition, we can also speed up the DRL training through better initialization, by leveraging the expert functions for determining the initial policy (Lines 8 and 9 in Algorithm 1).

Specifically, we initialize the deep Q-network through knowledge distillation [15] on the expert functions. The first step is to extract the knowledge from multiple expert functions (h_u and h_o in our case) to a dataset D_{init}. We set the input dataset as D^x_{init} and the corresponding label set as D^y_{init}. In setting D^x_{init}, we utilize all the unlabeled historical data, which only contain the system state. And the corresponding labels are calculated in a way that is similar to the strategy introduced earlier for integrating expert functions. That is, suppose we have n_h expert functions, then

$$D^y_{init} = \{y | y = (q_1, q_2, \cdots, q_m)\}, \tag{13}$$

$$q_j = \min_i (Q_i(s, f_j)), \tag{14}$$

where $s \in D^x_{init}$, $j \in [1 \cdots m]$, $i \in [1 \cdots n_h]$. As the expert functions we utilize are not as accurate as of the optimal (Q-) value function, we further add two mixing coefficients λ^α_{init}, λ^β_{init} for balancing the relative size of the Q value from different actions. So the new definition of D^y_{init} is

$$\begin{aligned} D^y_{init} = \{y | y = (&\frac{q_1 + (\lambda^\alpha_{init} - 1)\mu_q}{\lambda^\alpha_{init}\lambda^\beta_{init}}, \frac{q_2 + (\lambda^\alpha_{init} - 1)\mu_q}{\lambda^\alpha_{init}\lambda^\beta_{init}}, \\ &\cdots, \frac{q_m + (\lambda^\alpha_{init} - 1)\mu_q}{\lambda^\alpha_{init}\lambda^\beta_{init}})\}, \quad \mu_q = \frac{\sum_{j=1}^m q_j}{m}, \end{aligned} \tag{15}$$

where the definition of $q_j (j \in [1 \cdots m])$ remains the same. Then the next step is to train the deep Q-network of our DRL agent by using the obtained dataset D_{init}. As we consider a regression task, we apply the mean square error as the loss function

$$\mathcal{L}_{init} = \| y - y_{pred} \|^2, y \in D^y_{init}, \tag{16}$$

where y_{pred} is the deep Q-network prediction. We obtain the network weight initialization by training for n_{init} epochs. Moreover, with such policy initialization, we can use a smaller learning rate to tune the deep Q-network in the later DRL stages.

Parameter	Value	Parameter	Value
Expert-model	$[len(s \in \mathcal{S})$, 256,256,256, 256,256,256,2]	Deep Q-network	$[len(s \in \mathcal{S})$, 256, 256, 256, 256, 51]
m	51	Δt	15 mins
γ	0.99	α	1.0
T_{lower} (occupied)	22 ℃	T_{upper} (occupied)	26 ℃
T_{lower} (unoccupied)	12 ℃	T_{upper} (unoccupied)	30 ℃
β	100.0	m_u	7
ξ	1.0	n	5760

Table 1: Hyper-parameters used in our experiments.

4 EXPERIMENTAL RESULTS

4.1 Experiment Settings

We conduct our experiments on a Ubuntu 20.04 OS server equipped with NVIDIA RTX A5000 GPU cards. Docker [29] is utilized for the environment configuration, with Python 3.7.9 and learning framework Pytorch 1.9.0. All neural networks are optimized through the Adam optimizer [19].

We use the building simulation tool in [7] to simulate the behavior of single-zone commercial buildings, with an OpenAI-Gym [3] interface. We model two buildings as defined in the Building Energy Simulation Test validation suite [18]: one is with a lightweight construction (known as Case600FF) and the other is with a heavyweight construction (known as case900FF). Both buildings have the same model settings except that the wall and floor construction have either light or heavy materials. The floor dimensions are $6m$-by-$8m$ and the floor-to-ceiling height is $2.7m$. There are four exterior walls facing the cardinal directions and a flat roof. The walls facing east-west have the short dimension. The south wall contains two windows, each $3m$ wide and $2m$ tall. The use of the building is assumed to be a two-person office with a light load density. The lightweight building is assumed to be located at Riverside, California, USA, and the heavyweight building is assumed to be located at Chicago, Illinois, USA. The weather data for different locations are obtained from the Typical Meteorological Year 3 database [43]. In addition, the various parameters and hyper-parameters mentioned in the previous sections are listed in Table 1.

4.2 Evaluation of Our Framework and Comparison with Standard DDQN

We apply our proposed framework to building HVAC control and demonstrate its effectiveness in accelerating the DRL training, in particular the standard DDQN algorithm. We repeat each experiment 4 times and show the average results.

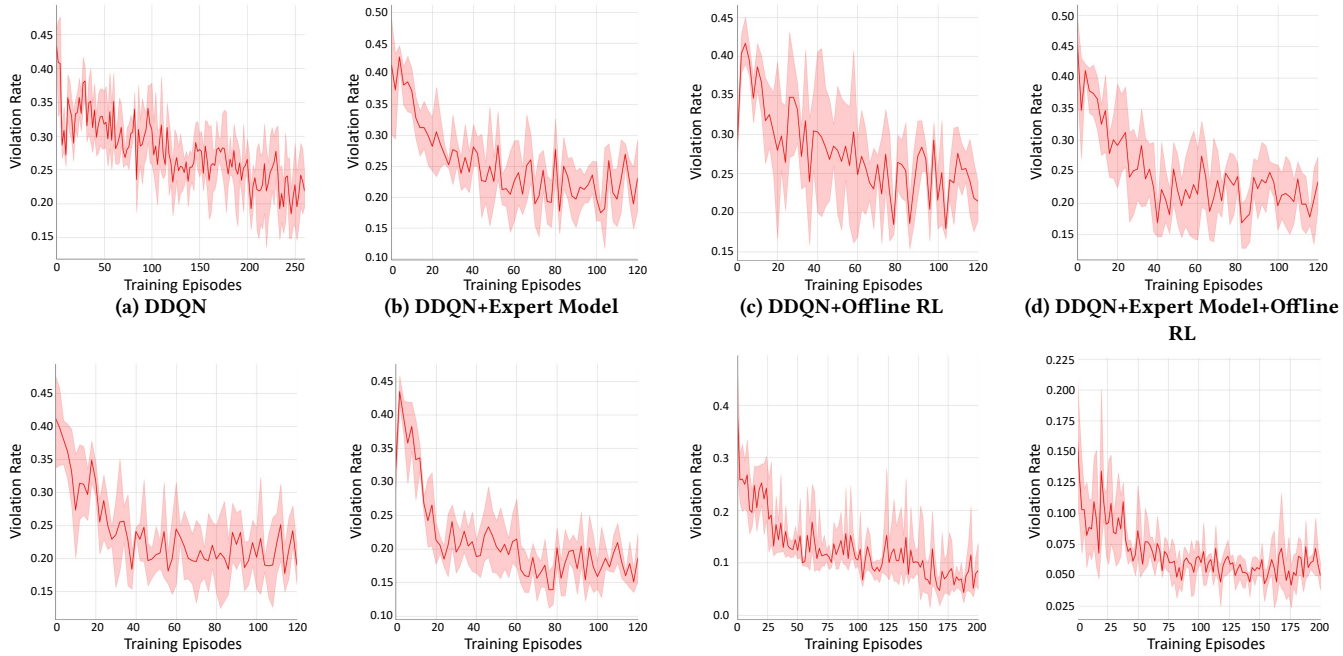
Comparison with Standard DDQN on Training Efficiency: Figure 2 demonstrates the temperature violation rate of the trained controller under different approaches for the lightweight building with weather data from Riverside. Temperature violation rate is one of the main objectives for DRL. It is defined as the percentage of the time the indoor temperature is outside of the comfortable temperature zone, similarly as used in [7, 41, 45, 46].

Method	Number of Episodes
DDQN	212
DDQN+Expert Model	68
DDQN+Offline RL	78
DDQN+Expert Model+Offline RL	40
DDQN+Expert Model+Offline RL +Expert Rules	36
DDQN+Expert Model+Offline RL +Expert Rules+Init	24

Table 2: Number of episodes required to reach the violation rate of 0.2 for the standard DDQN baseline and our approach with various techniques included (the last line being our approach with all techniques in Algorithm 1).

Figure 2a shows the training process of the standard DDQN, and the model needs **about 212 episodes to reach a violation rate at around 20%** for this building from [7] (20% may seem high, but it is due to the limitation of this particular building and its cooling-only HVAC system; more explanation on this later with Figure 4). Figure 2b shows the training process when we add a neural network-based expert model that generates the expert function h_u. About 68 episodes are needed to reach the same violation rate. Figure 2c shows the training process when we add offline RL that generates the expert function h_o, and about 78 episodes are needed to reach the violation rate of 20%. Figure 2d shows the results when we apply both expert functions h_u and h_o, but without the expert rules. We can see that about 40 episodes are needed. Figure 2e shows the results when we integrate the two expert functions h_u and h_o, as well as an expert rule f using the method introduced in Section 3.2. f is defined as follows: when the indoor temperature is below 22℃, the control action is suggested to be set within the set of $\{f_0, f_1, f_2, f_3\}$; if the indoor temperature is above 27℃, the control action is suggested to be set within the set of $\{f_{m-3}, f_{m-2}, f_{m-1}, f_m\}$. We can see that the number of episodes needed is about 36. Finally, Figure 2f shows the training process when we apply all of our proposed techniques, including integrating the expert functions from expert model and offline RL as well as the expert rules, using the integrated expert function to guide DRL training, and conducting policy initialization with the expert functions. We can see that **now only 24 episodes are needed to reach the same violation rate as the standard DDQN, an 8.8X reduction in training time**. Table 2 summarizes the above number of episodes required to reach the violation rate of 0.2 for the standard DDQN baseline and our approach with various techniques included.

For further evaluation, we also conduct experiments on the heavyweight building with weather data from Chicago. In this set of experiments, the major change of the parameters is that the scaling factor β is set to 1.0 in Equation (1). This is because that the average energy consumption of this HVAC system is much higher than that of the previous building, and we need to re-balance the energy cost and the temperature violation in the reward design. Figure 2g and Figure 2h shows the comparison between our approach and the standard DDQN. And the experiments show that the number of episodes needed to reach a violation rate of 5% is reduced from 160 to 80. The improvement, while still significant, is



(a) DDQN (b) DDQN+Expert Model (c) DDQN+Offline RL (d) DDQN+Expert Model+Offline RL

(e) DDQN+Expert Model+Offline RL+Expert Rules (f) DDQN+Expert Model+Offline RL+Expert Rules+Init (g) DDQN (h) DDQN+Expert Model+Offline RL+Expert Rules+Init

Figure 2: Figure 2a to Figure 2f show the comparison between our approach (in different settings with various techniques included) and the standard DDQN method on the lightweight building. The weather data is from Riverside, CA, USA. The x-axis shows the training episodes. The y-axis shows the temperature violation rate. Figure 2a shows the training process under the standard DDQN method. About 212 episodes are needed to reach a violation rate of 0.2. Figure 2b, Figure 2c, Figure 2d, and Figure 2e show the results when we gradually add an expert model that generates expert function h_u, offline RL that generates expert function h_o, an expert rule, and policy initialization based on expert functions, respectively. And we can observe the improvement on the required episodes step by step. Figure 2f shows the training process when we apply all of our techniques. In this case, only 24 episodes are needed to reach the violation rate of 0.2, an $8.8X$ improvement over standard DDQN. Then Figure 2g and Figure 2h show the comparison between our approach with all techniques included (right) and the standard DDQN baseline (left) on the heavyweight building with larger thermal capacity under the weather data from Chicago, IL, USA.

much less than the lightweight building. We suspect that this may be due to the quality of the historical data and plan to investigate it further in future work.

Energy Cost and Other Details: Besides temperature violation rate and the number of episodes for reaching the goal of violation rate below 0.2 (i.e., training efficiency), we also assess the energy cost of the learned controllers during our experiments. We observed that different methods, including the standard DDQN baseline and our approach with various techniques included, achieve very similar energy cost for the learned controllers – in fact within 1% for both the lightweight building and the heavyweight building we tested.

Figure 3 shows the normalized energy cost of our approach with all techniques included for the lightweight building with weather data from Riverside. We can observe that the energy cost quickly decreases to a lower value within 5 to 10 epochs and slightly fluctuates in the later training epochs.

Figure 4 illustrates the building temperature over 2 days, under the controller learned with our approach with all techniques

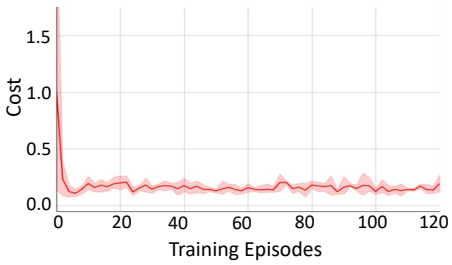


Figure 3: Normalized energy cost during training for our approach with all techniques included for the lightweight building with weather data from Riverside.

included, for the lightweight building with weather data from Riverside. We can see that the temperature violation rate is around 20%. It is relatively high because some violations are very hard to avoid for this particular building. Specifically, the HVAC system is set to only work during the occupied hours (from 7am to 7pm) and

the comfortable temperature range is much more strict during that time (22℃ to 26℃) compared to during the unoccupied time (12℃ to 30℃) [7]. This makes it almost impossible to meet the comfortable temperature range early in the morning since the HVAC system only provides cooling. We can see that after the early morning hours, the temperature is controlled well within the comfortable range by our controller.

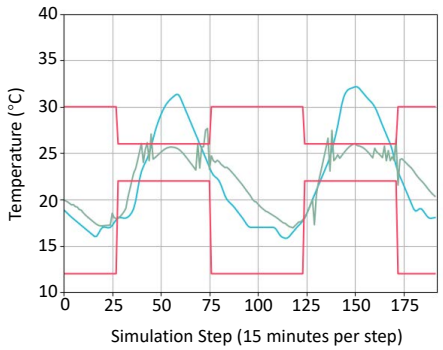


Figure 4: An illustration of the building temperature over 2 days under the controller learned from our approach with all techniques included. The red lines bound the comfortable temperature range. The blue line is the outdoor temperature in Riverside, CA. The green line is the indoor temperature under the learned controller.

4.3 Ablation Studies

Impact of the Historical Data Quantity: We are interested in knowing how the quantity of the historical data may affect the performance of our approach. We conduct a series of experiments that have the quantity of the historical data chosen from $\{5760, 2880, 1440, 720\}$ (i.e., from 2 months of data to 7.5 days of data). The results are shown in Table 3. We can observe that the training becomes faster as the quantity of the historical data becomes larger, as what we would expect.

#Samples	720	1440	2880	5760
#Episodes	116	78	62	24

Table 3: The number of epochs needed by our approach (with all techniques included) for reaching the violation rate of 20% for the lightweight building, under different quantity of the historical data.

Impact of the Historical Data Quality: We also study the performance of our approach under different level of quality for the historical data. Previously we use the historical data collected from an existing controller on the target building. To study different historical data quality, we choose to take random actions with a probability of p. Table 4 shows the results. Our approach performs better with a smaller p, i.e., when our approach learns from historical data based on more reasonable control actions.

p	1.0	0.8	0.4	0.2	0.0
#Episodes	110	104	88	60	24

Table 4: The number of epochs needed by our approach (with all techniques included) for reaching the violation rate of 20% for the lightweight building, under different quality of the historical data.

The Usage of Abstract Phyiscal Model: In addition, we also try to utilize an abstract physical model, i.e., the ARX model from [45], as the expert model to generate h_u, instead of learning a neural network. The training process is shown in Figure 5. About 64 episodes are needed to reach the same violation rate, more than the case where the expert model is a neural network learned from historical data. We think that this is due to the simplicity of the ARX model, and plan to investigate the performance of other abstract physical models in future. Nevertheless, it still provides considerable improvement over the standard DDQN.

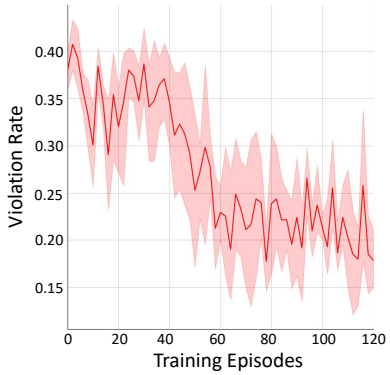


Figure 5: Training result for the lightweight building when the expert model in our approach (with all techniques included) is constructed from an abstract physical model.

5 CONCLUSIONS

In this paper, we present a systematic, unified framework to accelerate online RL for building HVAC control with heterogeneous expert guidances, including abstract physical models, historical data, and expert rules. These guidances are unified through the learning of expert functions, which are then used to accelerate DRL with prior-guided learning and policy initialization. A series of experiments demonstrate that our approach can significantly reduce the training time over previous DRL methods. We believe that our approach not only addresses a critical challenge in applying DRL to building domain, but also has the potential in other domains where existing expertise could be leveraged in improving learning efficiency and performance. We plan to investigate this further in future work.

ACKNOWLEDGMENTS

We gratefully acknowledge the support from Department of Energy (DOE) award DE-EE0009150 and National Science Foundation (NSF) awards 1834701 and 2038853.

REFERENCES

[1] Saber Abrazeh, Saeid-Reza Mohseni, Meisam Jahanshahi Zeitouni, Ahmad Parvaresh, Arman Fathollahi, Meysam Gheisarnejad, and Mohammad-Hassan Khooban. 2022. Virtual Hardware-in-the-Loop FMU Co-Simulation Based Digital Twins for Heating, Ventilation, and Air-Conditioning (HVAC) Systems. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2022).

[2] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. 2020. An optimistic perspective on offline reinforcement learning. In *ICML*. PMLR.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[4] Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. 2020. BAIL: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 18353–18363.

[5] Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. 2021. Heuristic-guided reinforcement learning. *NeurIPS* (2021).

[6] Drury B Crawley, Linda K Lawrie, Curtis O Pedersen, and Frederick C Winkelmann. 2000. Energy plus: energy simulation program. *ASHRAE journal* 42, 4 (2000), 49–56.

[7] Yangyang Fu, Shichao Xu, Qi Zhu, and Zheng O'Neill. 2021. Containerized framework for building control performance comparisons: model predictive control vs deep reinforcement learning control. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 276–280.

[8] Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. 2019. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708* (2019).

[9] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *NeurIPS* (2021).

[10] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*. PMLR, 2052–2062.

[11] Guanyu Gao, Jie Li, and Yonggang Wen. 2019. Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning. *arXiv preprint arXiv:1901.04693* (2019).

[12] Guanyu Gao, Jie Li, and Yonggang Wen. 2020. DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning. *IEEE Internet of Things Journal* 7, 9 (2020), 8472–8484.

[13] Yijie Guo, Shengyu Feng, Nicolas Le Roux, Ed Chi, Honglak Lee, and Minmin Chen. 2020. Batch reinforcement learning through continuation method. In *International Conference on Learning Representations*.

[14] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).

[16] Bo Huang, Yimin Zhu, Yongbin Gao, Guohui Zeng, Juan Zhang, Jin Liu, and Li Liu. 2021. The analysis of isolation measures for epidemic control of COVID-19. *Applied Intelligence* 51, 5 (2021), 3074–3085.

[17] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456* (2019).

[18] R Judkoff and J Neymark. 1995. International Energy Agency building energy simulation test (BESTEST) and diagnostic method. (2 1995). https://doi.org/10.2172/90674

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Neil E Klepeis, William C Nelson, Wayne R Ott, John P Robinson, Andy M Tsang, Paul Switzer, Joseph V Behar, Stephen C Hern, and William H Engelmann. 2001. The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology* 11, 3 (2001), 231–252.

[21] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

[22] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[23] Paulo Lissa, Michael Schukat, Marcus Keane, and Enda Barrett. 2021. Transfer learning applied to DRL-Based heat pump control to leverage microgrid energy efficiency. *Smart Energy* 3 (2021), 100044.

[24] C Lombard and EH Mathews. 1992. Efficient, steady state solution of a time variable RC network, for building thermal analysis. *Building and Environment* 27, 3 (1992), 279–287.

[25] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. 2012. Model Predictive Control for the Operation of Building Cooling Systems. *IEEE Transactions on Control Systems Technology* 20, 3 (2012), 796–803.

[26] Mehdi Maasoumy, Alessandro Pinto, and Alberto Sangiovanni-Vincentelli. 2011. Model-based hierarchical optimal control design for HVAC systems. In *Dynamic Systems and Control Conference*, Vol. 54754. 271–278.

[27] Mehdi Maasoumy, M Razmara, M Shahbakhti, and A Sangiovanni Vincentelli. 2014. Handling model uncertainty in model predictive control for energy efficient buildings. *Energy and Buildings* 77 (2014), 377–392.

[28] Sven Erik Mattsson, Hilding Elmqvist, and Martin Otter. 1998. Physical system modeling with Modelica. *Control Engineering Practice* 6, 4 (1998), 501–510.

[29] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux journal* 2014, 239 (2014), 2.

[30] Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. 2020. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359* (2020).

[31] Aviek Naug, Ibrahim Ahmed, and Gautam Biswas. 2019. Online energy management in commercial buildings using deep reinforcement learning. In *2019 IEEE SMARTCOMP*. IEEE, 249–257.

[32] U.S. Department of Energy. 2011. *Buildings energy data book*.

[33] United States Department of Labor. 2021. OSHA Technical Manual (OTM) Section III: Chapter 2.

[34] Bjarne W Olesen and Gail S Brager. 2004. A better way to predict comfort: The new ASHRAE standard 55-2004. (2004).

[35] Saran Salakij, Na Yu, Samuel Paolucci, and Panos Antsaklis. 2016. Model-Based Predictive Control for building energy management. I: Energy modeling and optimal control. *Energy and Buildings* 133 (2016), 345–358.

[36] Jorren Schepers, Reinout Eyckerman, Furkan Elmaz, Wim Casteels, Steven Latré, and Peter Hellinckx. 2021. Autonomous Building Control Using Offline Reinforcement Learning. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. Springer, 246–255.

[37] Mohamed Toub, Chethan R Reddy, Meysam Razmara, Mahdi Shahbakhti, Rush D Robinett III, and Ghassane Aniba. 2019. Model-based predictive control for optimal MicroCSP operation integrated with building HVAC systems. *Energy Conversion and Management* 199 (2019), 111924.

[38] Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, et al. 2022. Jump-Start Reinforcement Learning. *arXiv preprint arXiv:2204.02372* (2022).

[39] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *AAAI 2016*.

[40] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. 2020. Critic regularized regression. *Advances in Neural Information Processing Systems* 33 (2020), 7768–7778.

[41] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *54th Annual Design Automation Conference*.

[42] Tianshu Wei, Qi Zhu, and Nanpeng Yu. 2015. Proactive demand participation of smart buildings in smart grid. *IEEE Trans. Comput.* 65, 5 (2015), 1392–1406.

[43] Stephen Wilcox and William Marion. 2008. Users manual for TMY3 data sets. (2008).

[44] Qingqing Xu and Stevan Dubljevic. 2017. Model predictive control of solar thermal system with borehole seasonal storage. *Computers & Chemical Engineering* 101 (2017), 59–72.

[45] Shichao Xu, Yangyang Fu, Yixuan Wang, Zheng O'Neill, and Qi Zhu. 2021. Learning-based framework for sensor fault-tolerant building HVAC control with model-assisted learning. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 1–10.

[46] Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. 2020. One for many: Transfer learning for building hvac control. In *Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 230–239.

[47] Liang Yu, Yi Sun, Zhanbo Xu, Chao Shen, Dong Yue, Tao Jiang, and Xiaohong Guan. 2020. Multi-agent deep reinforcement learning for HVAC control in commercial buildings. *IEEE Transactions on Smart Grid* 12, 1 (2020), 407–419.

[48] Liang Yu, Weiwei Xie, Di Xie, Yulong Zou, Dengyin Zhang, Zhixin Sun, Linghua Zhang, Yue Zhang, and Tao Jiang. 2019. Deep reinforcement learning for smart home energy management. *IEEE Internet of Things Journal* 7, 4 (2019), 2751–2762.

[49] Kyungtae Yun, Rogelio Luck, Pedro J Mago, and Heejin Cho. 2012. Building hourly thermal load prediction using an indexed ARX model. In *Energy and Buildings*.

[50] Xiangyu Zhang, Xin Jin, Charles Tripp, David J Biagioni, Peter Graf, and Huaiguang Jiang. 2020. Transferable reinforcement learning for smart homes. In *Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities*. 43–47.

[51] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, and Khee Poh Lam. 2019. Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning. *Energy and Buildings* 199 (2019), 472–490.

[52] Zhiang Zhang, Adrian Chong, Yuqi Pan, Chenlu Zhang, Siliang Lu, and Khee Poh Lam. 2018. A deep reinforcement learning approach to using whole building energy model for hvac optimal control. In *2018 Building Performance Analysis Conference and SimBuild*, Vol. 3. 22–23.

Cross-Zone and Extreme-Aware Mobility Learning of Crowd Interactions with Built Environments

Suining He[1] Bing Wang[1] Kang G. Shin[2] Mahan Tabatabaie[1]
University of Connecticut[1] University of Michigan[2]
{suining.he, bing, mahan.tabatabaie}@uconn.edu[1], kgshin@umich.edu[2]

ABSTRACT

We propose an ***A***daptive ***Cro***wd mobility analytics system based on ***C***ross-zone ***I***nteractive learning (**ACroCI**) to capture, interpret, and forecast how the mobility of human crowds *interacts* with built or man-made environments (e.g., building functions, event occurrences, and changes in the crowd sensing infrastructures). We have conducted a large-scale real-world case study of ACroCI by leveraging the collective and anonymized association data harvested from the campus Wi-Fi access points (APs), to understand how the interactions affect the forecast of crowd flows. We have analyzed the large-scale Wi-Fi association data and derived adaptive learning and data-driven designs on important crowd mobility features like multi-level co-flows and local-global cross-zone interactions. ACroCI accounts for these interactive features and adaptively learns the multi-scale crowd distributions with a novel capsule neural network augmented by interactive attention routing, and accurately predicts the arrivals at, and departures from, each AP. Strengthened by multi-task extreme-aware learning and efficient data imputation, ACroCI further adapts to extreme flows when the crowds interact with events, and initializes its model for altered APs. Our extensive experimental evaluations with $>4.8 \times 10^7$ association data from two large universities have corroborated the accuracy, adaptivity, robustness, and effectiveness of ACroCI in forecasting the crowd interactions with the man-made environments (in terms of crowd flows), achieving a >20% accuracy improvement over the other state-of-the-arts.

1 INTRODUCTION

Crowd mobility analytics (CMA) — i.e., interpreting and understanding the movement of crowds — have become increasingly important for many spacious urban and/or public places, such as college campuses, large airports, and malls, where many people are likely to gather together within and across the human-built or artificial environments like building rooms, corridors, and sidewalks. While the social and behavioral analysis of crowds [11, 24], including the interactions among crowds, has been extensively studied in various ubiquitous and urban computing contexts, how crowds *interact* with the built or man-made environments, such as the building functions, event occurrences inside/across buildings, and infrastructure alterations or changes [2], and how such interaction may benefit mobility modeling, remain a largely unexplored but important subject of CMA.

Human crowds usually move responsively given various built environment settings. For instance, the college students may form certain travel routines on a campus, given their interpretation of relative proximity of buildings (as well as their purposes) such as residential halls (living), recreational centers (recreation), dining halls (dining), and libraries (studies). One may, therefore, expect interactive co-presences of crowds at multiple locations during certain periods of a day. Crowds may also interactively pick alternative entrances, exits, or routes in response to closure of certain corridors or sidewalks. Understanding and incorporating such dynamic interactions within the CMA system design, particularly learning and predicting interactive crowd movements, will benefit various important ubiquitous, mobile, and urban applications, such as event monitoring and facility management.

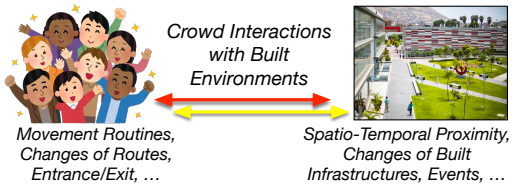


Fig. 1: Illustration of design motivations.

To this end, we propose a novel CMA system with insights of such interactions, using collective and anonymized Wi-Fi association and disassociation data harvested from a campus network infrastructure as a practical case study. Through the system deployment study, we would like to understand how incorporating interactions, as illustrated in Fig. 1(a), between human crowds and the built environments impacts the accuracy and effectiveness of CMA, especially in predicting the mobility of crowd flows. Towards such a ubiquitous CMA system, we must address the following two major technical challenges:

(1) Lack of modeling the local-global and cross-zone interactions with the man-made environments: Crowd mobility is highly complex and often involves *multi-scale* complexities due to hourly, daily, and weekly routines, periodic personal activities and preferences. Furthermore, *co-presences* and *co-flows* of crowds occur not only *locally* at neighboring campus zones but also *globally* across *distant* zones. Conventional learning without considering multi-scale and local-global dependency and interaction modeling cannot capture the underlying sophisticated campus activities and provide satisfactory prediction.

(2) Absence of extreme awareness and adaptivity to interactions with dynamic sensor measurement environments: Besides the inherent spatial and temporal complexities, the distributions of crowds may be affected greatly by transient or abnormal

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564065

campus events (e.g., a conference gathering or an emergency), leading to a surge or drop of crowd interaction behaviors (say, arrivals or departures). Conventional *extreme-agnostic* Wi-Fi data learning, however, fails to take into account the potential *extreme awareness* within the crowd mobility data. Furthermore, the Wi-Fi infrastructure may change due to network maintenance and reconfiguration, which can impact the Wi-Fi data collection and subsequent model learning. In particular, how to learn and predict crowd mobility with APs that are newly-introduced or relocated without historical data for model training, i.e., handling "*cold-start*" for model initialization, remains largely unexplored.

To address the above challenges, we propose an **A**daptive **Cro**wd mobility analytics system based on **C**ross-zone **I**nteractive learning (**ACroCI**). To better understand how the crowd interactions with the built/artificial environments affect the mobility modeling, we have further designed a crowd flow prediction experiment, i.e., forecasting the flows of crowds at each individual Wi-Fi AP, as a practical and easy-to-evaluate case study. In this case study, at each time interval, for each AP, the *in-flow* is considered to represent the total number of clients entering its close proximity, while the *out-flow* denotes that of those leaving. Specifically, our crowd interaction studies with ACroCI take in the AP-level associations/disassociations and their zone-level representations, statistical time-series features, and external factors. Then, our crowd interaction learning framework, with a novel interactive **A**ttention **R**outing **Cap**sule network (ARCap) as the core, forecasts the AP-level crowd flows.

This paper makes the following contributions:

C1. Spatio-temporal interactive campus crowd mobility analytics (Sec. 2): We have conducted a comprehensive analysis of campus mobile associations and crowd flows using real-world Wi-Fi association datasets, one collected from our university campus and an open-sourced dataset from a university in Northern Europe [18]. We have derived several spatial and temporal representation designs, motivations, and insights regarding the interactions between the crowds and the built environments, including multi-scale temporal and co-flow complexities, local-global cross-zone dependencies, as well as extreme crowd flows, to motivate our adaptive interaction learning model formulation.

C2. Learning interactions between crowds and built environments (Sec. 3): Within ACroCI, the complex spatial correlations across different APs and their neighborhoods are characterized via a novel spatio-temporal mobility heatmap representation, and the crowd flow patterns are further learned by a novel attention routing capsule neural network. The multi-scale co-flows and local-global cross-zone interactions between the crowds and the built environments can be jointly and interactively captured by our novel vectorized capsule structure with attention routing. Then, ACroCI predicts the AP-level crowd flows based on the fusion of multiple learners, yielding high accuracy and robustness.

C3. Awareness and adaptivity augmentation (Sec. 4): We have further designed a novel auxiliary module with novel multi-task extreme-aware learning to capture statistical crowd flow time-series features and forecast the extreme crowd distributions introduced by transient and potentially abnormal university events. We have also designed an efficient data imputation mechanism for our CMA system to handle the cold-start issue of newly-introduced or relocated APs after network reconfiguration.

C4. Extensive experimental evaluations (Sec. 5): With association data from the above-mentioned university campuses, we have conducted extensive experimental evaluations with $>4.8\times10^7$ Wi-Fi association records from over 2.36×10^5 clients in total. These results have corroborated the importance of incorporating interactions between crowds and the built environments, as well as accuracy, robustness, extreme-awareness, and adaptivity of ACroCI in predicting the campus mobility and crowdedness.

2 MOTIVATIONS AND FORMULATIONS

2.1 Crowd Mobility Data Pre-processing

Crowd Mobility Data: ACroCI is based on the following two large-scale campus WLANs.

(a) Campus A: We used our information technology (IT) service of University of Connecticut to collect the Wi-Fi association and disassociation records at our university campus (denoted as Campus A) via the Cisco Prime Infrastructure. The selected APs with the given GPS coordinates (longitudes and latitudes) cover a campus area of 4.4036×10^6 m^2. 1,059 APs and 126,923 clients are studied, obtaining a total of 9.677×10^6 records over a 15-month data collection period (2015/06–2016/09). Each record corresponds to one AP association event, with the MAC addresses of both the associated AP and the user's connected client device, user name (ID), the start time as well as the association duration, and the IP address. We infer the departure or disassociation time by adding the association duration to the start time.

(b) Campus B: The 880 APs selected from another university campus in Northern Europe [18] (denoted as Campus B) cover an area of 1.308×10^6 m^2 and 109,197 clients, producing a total of 3.8593×10^7 Wi-Fi association records during the 16-month long data collection period (2014/01–2015/04).

Crowd Mobility Data Preprocessing: For Campus A, the data collection of individual/crowd locations is enabled by a single-sourced Wi-Fi service provision system managed by our university IT service, and the user anonymity, including privacy of the user ID and device's MAC address, is maintained. Related privacy and ethical considerations have been discussed and vetted by the university's Institutional Review Board or IRB; we were informed that no IRB application was required as only aggregate anonymized data is used in our study. We have anonymized the user names (IDs), client device MAC addresses, and IP addresses (if any) of the association records. We use these anonymized user IDs to map devices to individual users, hence mapping the Wi-Fi association data to the crowd flows, and the IDs are discarded immediately after the mapping. Identities in the dataset from Campus B are also anonymized before its distribution to preserve individual privacy. We infer the AP-level crowd flows (in/out) by:

AP-level Associations/Disassociations: For ease of modeling, we discretize the time domain into intervals (say, each with 60min in our settings). The period k and the index k are used interchangeably, both referring to the k-th time interval. Let M be the number of access points (APs) on a campus. Then, we denote the number of associations and disassociations at AP i ($i \in \{1,\ldots,M\}$) in the k-th time interval as $A_i^{(k)}$ and $D_i^{(k)}$, which form the resultant M-dimensional (M-d) vectors of AP-level associations and disassociations, i.e., $\mathbf{A}^{(k)} = \left[A_1^{(k)},\ldots,A_M^{(k)}\right]$ and $\mathbf{D}^{(k)} = \left[D_1^{(k)},\ldots,D_M^{(k)}\right]$.

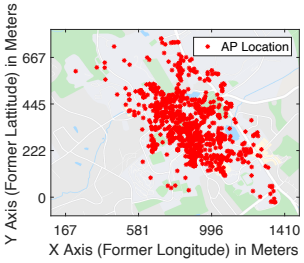


Fig. 2: AP locations studied at Campus A.

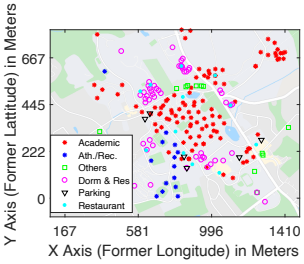


Fig. 3: Points of interests (POIs) at Campus A.

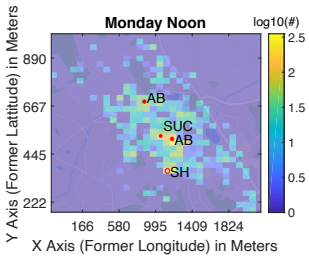


Fig. 4: Heatmap of associations at Campus A (a Monday noon).

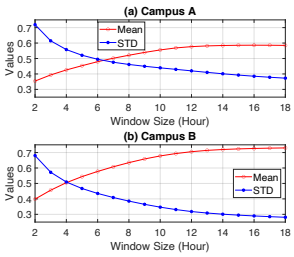


Fig. 5: Multi-level temporal interactions.

2.2 System Design Motivations

We derive the following data-driven motivations regarding interactions of the crowds with the built environments on campus (denoted as **M1**–**M5**).

M1 – Spatial Interactions with Points-of-Interest (POIs).

Taking our university campus (Campus A) as an example, we show in Fig. 2 the AP locations, demonstrating the pervasive coverage of the campus Wi-Fi infrastructures. We also show in Fig. 3 the 219 points of interests (POIs) on Campus A, including academic buildings, athletics or recreation facilities, dormitory and residential area, restaurants and cafes, and parking lots. The geolocations of the POIs are derived from the free editable geographic database OpenStreetMap. We illustrate the spatial heatmap of aggregated AP associations (in $\log_{10}(\cdot)$) during 12:00–13:00 of Monday in Fig. 4. On Monday dense AP associations and "implied" crowdedness can be observed from multiple locations such as academic buildings (denoted as AB) and the student union canteen (denoted as SUC). However, we also observe that on Sunday (the same time period) the magnitude is much lower and student halls (denoted as SH in Fig. 4) as well as academic buildings remain to be hotspots.

One can expect that the campus facility and building distributions have strong impacts on the crowd flows. However, given such a complex and non-uniform distribution of APs, it is very difficult to model their mutual *spatial* interactions for effective crowd interaction learning. To handle the above issue, we leverage the inspiration of image learning, and conduct zone discretization to obtain the aggregate zone-level association data as the important inputs and feature representations for our model training.

Zone Discretization, Zone-level Associations, & Disassociations: We discretize the campus map into G *zones*. Specifically, we divide the map longitudinally and latitudinally into $W \times H$ equal-sized rectangular *grids*. Note that the shape and size of the zones can be customized according to each specific data analysis and prediction task need, and adapted to the map/building accessibility.

We aggregate the number of Wi-Fi associations or crowd arrivals at all APs installed within zone j in the interval k as $Y_j^{(k)}$. Then, we have the zone-level association as $\mathbf{Y}^{(k)} = \left[Y_1^{(k)}, \ldots, Y_G^{(k)}\right]$. Similarly, we find the disassociations or crowd departures as $O_j^{(k)}$, as $\mathbf{O}^{(k)} = \left[O_1^{(k)}, \ldots, O_G^{(k)}\right]$. Then, for each time interval k, we process the given zone-level Wi-Fi associations/disassociations into sequential mobility heatmap frames, denoted as $\mathbf{F}^{(k)}$, as *spatio-temporal mobility representations* for the model input. Note that while the spatio-temporal mobility representations are at the zone level, our prediction output is at each *individual* AP for fine-grained crowd monitoring.

M2 – Multi-scale Temporal Interactions.

We have investigated the temporal intervals to evaluate the multi-scale temporal correlations. Recall that the time domain has been discretized into multiple intervals, each 1 hour long in our setting. Then, by varying the sliding window of T consecutive time intervals, we evaluate the temporal correlations between the associations of APs i and j based on the Pearson correlations, i.e., $\text{corr}\,(i,j) = \frac{\sum_{t=1}^{T}\left(A_i^{(t)} - \bar{A}_i\right)\cdot\left(A_j^{(t)} - \bar{A}_j\right)}{\sqrt{\sum_{t=1}^{T}\left(A_i^{(t)} - \bar{A}_i\right)^2}\sqrt{\sum_{t=1}^{T}\left(A_j^{(t)} - \bar{A}_j\right)^2}}$, where $\bar{A}_i$ represents the mean flow in the window of T intervals at AP i. Similar results can be observed from departures. By varying the sliding window size, we show the means and variations of the temporal correlations of APs at each campus in Fig. 5. Overall, Campus A experiences the lower mean correlations and higher variations than Campus B. From both campus datasets, we can observe that for short-term intervals (say, <3 hours), the correlations tend to be small and varied, which is likely due to the short-term mobility dynamics. On the other hand, for the long-term intervals (say, >9 hours) the correlations tend to be large and consistent, mainly because there exist mobility patterns on various campus function zones. After the sliding window reaches a certain value (say, >12 hours in our cases), the temporal correlations start to converge.

M3 – Co-Flow and Local-Global Interactions.

Complex crowd interactions with different built environments (e.g., buildings) often result from their functions, which can be characterized by their points-of-interest (POIs) in Fig. 3. We further look at the APs in the same POI and obtain the correlations among APs at a pair of different POIs. Specifically, we consider several POIs that are closely related to a student's daily life: *(a)* academic buildings, *(b)* recreational/athletics facilities, *(c)* dormitory and residential zones, and *(d)* restaurants and cafes.

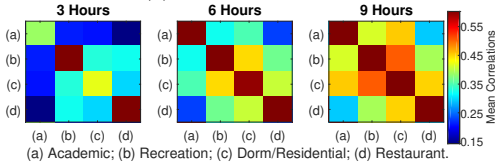


Fig. 6: Temporal interactions at selected POIs of Campus A.

To demonstrate the interactions, Fig. 6 plots the average correlations between each pair of POIs at multiple temporal scales (3, 6, and 9 hours). From the higher correlations (warmer colors) between the two categories of POIs, we can infer more similar trends of client arrivals, i.e., *co-flows*, at the two types of POI zones at the same time. We observe that for short sliding windows, more co-flows are likely to happen in campus zones with recreation/athletics facilities, dormitories, and restaurants, which are closely related to campus life apart from class activities. While for large sliding windows, more co-flows, with much higher correlations, can be observed at dormitories, recreational facilities as well as academic

buildings, demonstrating the long-term routine patterns on campus. We also note from Figs. 3 and 6 that as the recreation and dormitory buildings are largely at the peripheral areas of the campus and with noticeable mutual distances, such dependencies should be carefully modeled via not only *locally* (local nearby zones) but also *globally* adaptive (across distant zones) scopes.

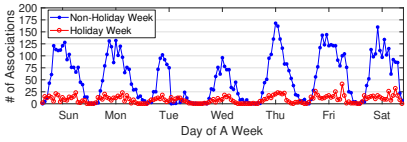


Fig. 7: Dynamics of associations during non-holiday and holiday weeks at an academic building (Campus A).

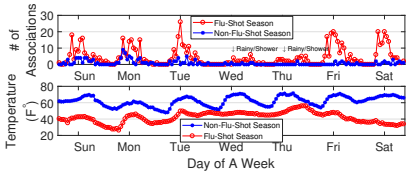


Fig. 8: Dynamics of associations during non-flu-shot and flu shot seasons at the student health center (Campus A).

M4 – Interactions with External Factors.

For Campus A, we further select two locations to further analyze how the crowds may respond to the external factors. Specifically, we select two APs, one inside an academic building and the other inside the student health center (clinic), and illustrate the associations and the crowd flows in Figs. 7 and 8. Fig. 7 shows a clear difference in Wi-Fi associations between holiday and non-holiday weeks on the campus. Another interesting finding comes from Fig. 8, where we compare the crowd flows between two weeks in a flu-shot season (usually October/November in our case) and a non-flu-shot season (in September). The respective temperature trends (average 41°F vs. 62°F, or 5°C vs. 16°C) are also shown in Fig. 8. More clinic visits are likely due to the lowering temperature and flu seasons, implying the needs of forecasting such interactions for public health concerns. We also observe some drops on Wednesday and Thursday due to rain and shower since the student health center is disjoint from other academic buildings and dorms which might deter the crowd mobility.

Motivated by above, for each campus, we collect weather conditions and day of week as the external factors for the crowd flow prediction. Specifically, for Campus A we collect the weather data from the National Oceanic and Atmospheric Administration (NOAA), while for Campus B we find the weather conditions based on the open data portal Weather Archive [1]. As shown in Table 1, we consider weather conditions, meteorological data, as well as event/time. For the categorical factors such as weather conditions and public holidays, we use one-hot encoding [8], i.e., the value is 1 if a condition exists (say, rainy) and 0 otherwise. For the numerical factors such as meteorological data, we conduct the max-min normalization [8]. We concatenate the categorical and numerical factors, and form the external feature vectors $\mathbf{E}$ (the dimension is 11 for Campus A and 15 for Campus B).

M5 – Interactions with Events & Environmental Changes.

We have conducted the following studies upon the crowds' interactions with the event dynamics and environmental alterations (in terms of AP alteration):

Table 1: External factors considered for the two datasets.

Factors	Campus A	Campus B
Weather Conditions	Foggy/Rainy/Misty/ Haze/Snowy (5-D)	Misty/Drizzle/Light Rain/Shower/Snow/ Freezing/Foggy/No Significant Clouds (8-D)
Meteorological Data	Temperature/Humidity/ Wind Speed (3-D)	Temperature/Pressure/Humidity/ Wind Speed (4-D)
Event/Time	Day of Week/Hour of Day/Public Holiday or Not (3-D)	

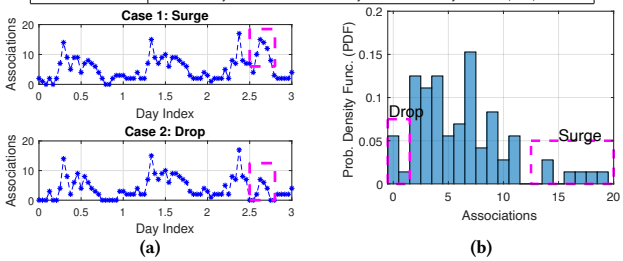


Fig. 9: Time series (a) and the PDF (b) of historical associations for an AP.

(i) Interactions with Events: Transient and abnormal campus events (e.g., a conference or emergency) may lead to significantly higher (surge) or lower (drop) volumes of crowd flows at certain APs than historical average, i.e., the average flows of the same time interval of a day in a week. We show in Fig. 9 regarding *(a)* time series and *(b)* probability distributions of flows in the same historical time interval (14:00–15:00) at an AP in an academic building on Campus A. We can observe that the crowd flow distribution at this AP can have significantly high or low values (highlighted). To model such an *extreme* surge or drop and enhance ACroCI's adaptivity, we take into account multiple *statistical time-series features* (detailed in Sec. 4.1) of the crowd flows at different campus zones as additional inputs for ACroCI. Meanwhile, we use *auxiliary labels* for the extremely high/low AP-level flows for the *joint* model training.

(ii) Interactions reflected by AP Alterations: Due to network maintenance or reconfiguration, some Wi-Fi APs may be introduced, relocated, or removed. A removed AP can be simply masked without further prediction. However, an AP that is either newly-installed or relocated (same MAC addresses but treated as "new" at the model side), may have different spatio-temporal neighborhood features, leading to "cold-start" initialization problem for the crowd interaction learning model due to lack of initial data. For instance, we have observed on Campus A that 11.08% of the APs have been newly introduced within an academic year (fall and spring semesters).

2.3 Problem Definition & Model Overview

Problem Definition. Motivated by the above data analysis, in our CMA prototype study, we consider the crowd flow prediction as a case study to evaluate the usefulness of incorporating interactions within ACroCI.

Specifically, we aim at designing a crowd interaction learning model $\mathcal{P}_{\boldsymbol{\theta}}(\cdot)$ (with model hyperparameters $\boldsymbol{\theta}$) to forecast AP-level crowd flows (arrivals or departures) in a target time interval k, denoted as $\hat{\mathbf{X}}^{(k)}$, which is either associations $\hat{\mathbf{A}}^{(k)}$ or disassociations $\hat{\mathbf{D}}^{(k)}$, given a sequence of w historical representations, $\mathbf{F}^{(\text{hist})} = \left\{\mathbf{F}^{(k-w)}, \mathbf{F}^{(k-w+1)}, \ldots, \mathbf{F}^{(k-1)}\right\}$, as well as the AP-level crowd flows in the previous interval, $\mathbf{X}^{(k-1)}$, external factors, $\mathbf{E}^{(k-1)}$, and other statistical time-series features, $\mathbf{F}^{(\text{stat})}$. It is formally given by

$$\hat{\mathbf{X}}^{(k)} = \mathcal{P}_{\boldsymbol{\theta}}\left(\mathbf{F}^{(\text{hist})}; \mathbf{X}^{(k-1)}; \mathbf{E}^{(k-1)}; \mathbf{F}^{(\text{stat})}\right). \quad (1)$$

Model and System Overview. Based on motivations **M1** – **M5**, we have designed and implemented ACroCI, whose model and

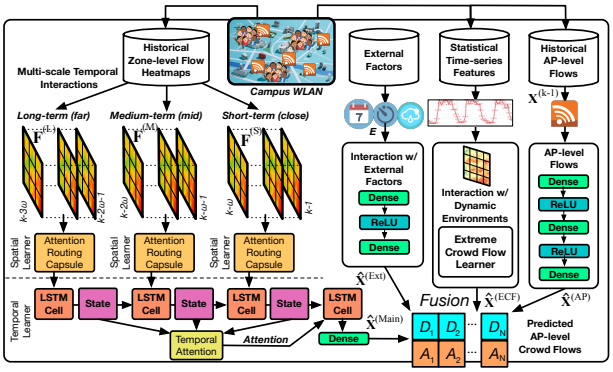


Fig. 10: Model and system framework overview of ACroCI.

system framework is illustrated in Fig. 10. To train the model, we first collect the historical Wi-Fi AP association records, AP locations, campus map, as well as other external factors (including the weather conditions and the university calendar) with the campus IT services. ACroCI pre-processes the above inputs, and the *data-driven studies* derive the model inputs and system parameters for the ACroCI's core model. Specifically, these features are processed and learned by the following learners.

1. Spatial Cross-Zone Interactive Learner (Sec. 3.1): This module captures the zone-to-zone correlations via the heatmap representation of spatial crowd flow distributions. Taking into account the spatial variations, multi-scale co-flows, and local-global cross-zone dependencies (**M1** & **M3**), we pre-process the crowd flows of an interval into a heatmap frame, where each grid element represents aggregated arrivals or departures, and leverage the attention routing capsule for interactive learning.

2. Multi-Scale Attention Temporal Learner (Sec. 3.2): To model the temporal interactions and dependencies of sequential time intervals (**M2**), we further feed the outputs of the spatial learners after processing multiple consecutive heatmap frames, to the temporal learner consisting of long short-term memory (LSTM) encoder and attention decoder.

3. Auxiliary Extreme Crowd Flow Learner (Sec. 4.1): To capture the correlations with extreme flows (**M5**), we extract multiple statistical features of the zone-level crowd flow time series to form another series of heatmap frames for ConvLSTM [8], and simultaneously forecast the crowd flows $\hat{\mathbf{X}}^{(\mathrm{ECF})}$ and the auxiliary labels $\hat{\mathbf{Z}}$. We provide a multi-task extreme-aware learning design beyond the spatial and temporal learners to *jointly* minimize the prediction errors as well as differences in predicted and estimated auxiliary labels.

4. External and AP-level Learners (Sec. 4.2): These two learners are deeply connected neural networks (with Dense layers) that predict $\hat{\mathbf{X}}^{(\mathrm{Ext})}$ and $\hat{\mathbf{X}}^{(\mathrm{AP})}$, given the input vectors of external factors (**M4**) and the crowd flows of the latest time interval prior to the target one, respectively. The outputs $\hat{\mathbf{X}}^{(\mathrm{Main})}$, $\hat{\mathbf{X}}^{(\mathrm{ECF})}$, $\hat{\mathbf{X}}^{(\mathrm{Ext})}$, and $\hat{\mathbf{X}}^{(\mathrm{AP})}$ from all four modules are merged for final prediction.

3 CORE DESIGNS OF ACROCI

3.1 Cross-Zone & Local-Global Interaction Learning

Input Spatio-Temporal Representations. In our prototype studies, we formulate the input spatial distributions of the associations and disassociations in the time interval k into the heatmap frames, i.e.,

$$\mathbf{Y}^{(k)} = \begin{bmatrix} Y_{11}^{(k)} & \cdots & Y_{1W}^{(k)} \\ \vdots & \ddots & \vdots \\ Y_{H1}^{(k)} & \cdots & Y_{HW}^{(k)} \end{bmatrix}, \quad \mathbf{O}^{(k)} = \begin{bmatrix} O_{11}^{(k)} & \cdots & O_{1W}^{(k)} \\ \vdots & \ddots & \vdots \\ O_{H1}^{(k)} & \cdots & O_{HW}^{(k)} \end{bmatrix}, \quad (2)$$

where $Y_{ij}^{(k)}$ ($O_{ij}^{(k)}$) is the number of associations (disassociations) within the i-th row (latitudinal), the j-th column (longitudinal) of the grid map in the k-th interval. The inherent correlations across different campus locations, including their functions and POIs, can be incorporated within the heatmap frames and processed by our subsequent capsule network learning module.

To accommodate and characterize different temporal scales of spatial distributions (Fig. 5 in **M2** and Fig. 6 in **M3**), we formulate spatio-temporal mobility representations $\mathbf{F}$'s in Definition 4 (Sec. 2.3) into the **s**hort, **m**edium, and **l**ong-term input tensors, denoted as $\mathbf{F}^{(\mathrm{S})}$, $\mathbf{F}^{(\mathrm{M})}$, and $\mathbf{F}^{(\mathrm{L})}$, respectively. If we are to predict the associations (or arrivals), each input tensor, $\mathbf{F}^{(x)} \in \{\mathbf{F}^{(\mathrm{S})}, \mathbf{F}^{(\mathrm{M})}, \mathbf{F}^{(\mathrm{L})}\}$, that is fed to a spatial learner (with the notation $x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$), comprises ω consecutive heatmap frames from a sliding window or a predetermined number of time intervals, i.e.,

$$\mathbf{F}^{(x)} = \left[\mathbf{Y}^{(k-m\cdot\omega)}, \mathbf{Y}^{(k-m\cdot\omega+1)}, \ldots, \mathbf{Y}^{(k-(m-1)\cdot\omega-1)}\right], \quad (3)$$

where the symbol $x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$ (either short-, mid- or long-term scale) with, respectively, $m \in \{1, 2, 3\}$. If we are to predict disassociations (or departures), we have $\mathbf{O}$ instead of $\mathbf{Y}$ in Eq. (3). Each heatmap frame in $\mathbf{F}^{(x)}$ is fed as a channel [8] for the input layer (total ω channels per temporal scale). We have empirically studied ω and observed a larger ω reduces errors but with a diminishing return, and hence adopt $\omega = 6$ in our studies.

Designs of Attention Routing Capsule. Given the input heatmap frames of crowd flows, we aim at capturing the complex spatial distributions, correlated co-flows, and local-global cross-zone dependencies (i.e., **M1** & **M3** in Sec. 2.2). However, via our later experimental observations, solely using conventional *scalar*-based networks, like convolutional neural network (CNN), cannot adequately characterize these and encode the transformation of the features. Therefore, we can observe large prediction errors for the complex Wi-Fi data learning and prediction scenarios.

To address this, we introduce the capsule neural network (Cap) [22] within ACroCI, where a structured group of neurons, i.e., *capsule*, forms a *vector* representation of the features. Specifically, each capsule can describe how the input heatmap distribution, as a target of interest, is instantiated as a *vector*, including its spatial position relative to the map, and captures more co-flow and local-global features than the scalar-based CNN.

We note that the conventional capsule network adopts dynamic routing and squash activation function [10, 22]. Such dynamic routing iteratively adjusts the values of the vectors between the capsule layers during training to find the vector agreement and learn the input features. However, the routing cannot further *differentiate* the importance of connections, dependencies, and interactions across spacious campus crowd distributions. To strengthen the model, ACroCI further integrates the *attention* mechanism to parameterize the routing among the capsules [5]. The attention mechanism helps

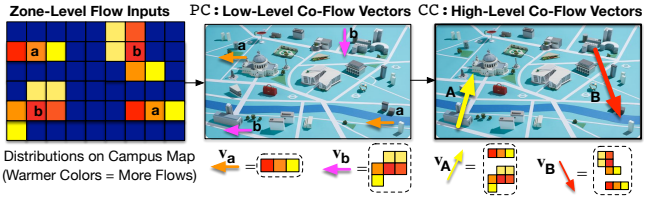


Fig. 11: Illustration of interactive cross-zone learning.

ACroCI learn a compatibility function between low-level and high-level features [17], thus empowering the learning performance and prediction accuracy of the capsule network.

Detailed Layer Designs. In the following, we present the *attention routing capsule* (denoted as ARCap) architecture, which consists of the primary capsule (PC) layer and the convolutional capsule (CC) layer. Before PC and after CC, ACroCI leverages the 2D convolutional layer or Conv2D for input preprocessing and output restructuring. Final output $\mathbf{X}^{(x)}$ is returned after an activation layer.

In an implementation, a capsule layer (either PC or CC) can be reshaped and restructured from the convolutional (say, $\mathsf{Conv}_{K\times K}(\cdot)$ with $K \times K$ kernels) module consisting of $N^{(l)} \times S^{(l)}$ filters [9, 22], where l is the index of the layer, and $N^{(l)}$ and $S^{(l)}$ represent the numbers of capsule channels and dimensions at the l-th layer, respectively. Compared to the convolutional layer, the capsule layer becomes $N^{(l)}$ channels, each of which is a group of neurons returning an output vector. We denote for each capsule layer l the height and width of the input as $H^{(l)}$ and $W^{(l)}$, respectively.

We present the detailed layer designs of ARCap as follows.

(a) *Input*: At the input of the spatial learner, ACroCI first preprocesses the tensor via batch normalization (denoted as $\mathsf{BN}(\cdot)$) and $\mathsf{Conv2D}(\cdot)$, deriving the initial coarse-grained features, i.e.,

$$\mathbf{X}^{(x,0)} = \sigma\left(\mathsf{Conv2D}\left(\mathsf{BN}\left(\mathbf{F}^{(x)}\right)\right)\right), \tag{4}$$

where we adopt ReLU for the activation $\sigma(\cdot)$. For each dataset, the batch normalization is intended to reduce the internal covariate shift within the data. The output $\mathbf{X}^{(x,0)}$ is then fed to the PC Layer at the ARCap structure.

(b) PC *Layer*: Crowd interactions between the nearby zones (say, low-level features of co-flows **a** and **b** in Fig. 11) are captured by the capsule layer characterizing such spatial structural information. Specifically, the PC layer first processes and returns the vector regarding each of the N capsule channels (indexed by n), i.e.,

$$\mathbf{v}_n^{(0)} = \sigma\left(\mathsf{Conv}_{K\times K}\left(\mathbf{X}^{(x,0)}\right)\right), \tag{5}$$

where $x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}$, and we adopt $K = 3$ in the convolution kernel, and ReLU for the activation $\sigma(\cdot)$. The correlations between multiple neighboring zones can be further extracted via the stacked layers. Then, the capsule activation conducts the affine convolutional transformation upon each capsule channel n, i.e.,

$$\mathbf{v}_n = \sigma\left(\mathsf{Conv}_{K\times K}\left(\mathbf{v}_n^{(0)}\right)\right). \tag{6}$$

where we adopt $K = 1$, and tanh for $\sigma(\cdot)$, which converts the output into the range of $[-1, 1]$ to restrict the value scale.

(c) CC *Layer*: CC aims at deriving the crowds' global interactions with the zones (say, high-level co-flow features in groups **A** and **B** from multiple distant zones in Fig. 11) via the capsule attention mechanism. As shown in Fig. 11, the high-level features in groups **A** and **B** in CC (as vectors $\mathbf{v_A}$ and $\mathbf{v_B}$), contain structural information from **a** and **b** (as vectors $\mathbf{v_a}$ and $\mathbf{v_b}$) after PC.

First, *convolutional transformation* converts each capsule channel, and forms new capsule channels with parameters shared locally with multiple channels of the preceding layer's channels, i.e.,

$$\tilde{\mathbf{v}}_n = \mathsf{Conv}_{K\times K}\left(\mathbf{v}_n\right), \tag{7}$$

where we adopt $K = 3$, and the resultant structure enables the attention mechanism interleaving the transformed capsules.

Second, between the transformed capsules and the *CC* layer, we conduct the *attention routing*. The attention weights λ_j's are the softmax [8] outputs of the logarithm probabilities [22] along the capsule channel axis at the previous layer $(l-1)$, i.e.,

$$\lambda_j \triangleq \mathsf{softmax}\left(e_j\right) = \frac{\exp\left(e_j\right)}{\sum_{n=1}^{N^{(l-1)}} \exp\left(e_n\right)}, \tag{8}$$

where the logarithm probabilities e_n [22] can be obtained through the 3D convolution upon the input $\tilde{\mathbf{v}}$, i.e.,

$$\mathbf{e} = [e_1, e_2, \ldots, e_N] = \mathsf{Conv3D}_{1\times 1\times D^{(l)}}\left(\tilde{\mathbf{v}}\right). \tag{9}$$

Unlike conventional dynamic routing [22], attention routing helps ACroCI learn the logarithm probabilities of the agreement coefficients between the ℓ-th and $(\ell-1)$-th layers. Attention routing adjusts the weights λ_j for each spatial location in the convolutional transformed capsules, such that the important locations of interest, as well as the relevant crowd interactions can be further derived.

Third, given Eqs. (7) and (8), each channel output is the weighted average of the logarithm probabilities and the vector output from each preceding capsule channel for the attention routing, i.e.,

$$\mathbf{v} \triangleq \sum_{j=1}^{N^{(l-1)}} \lambda_j \cdot \tilde{\mathbf{v}}_j. \tag{10}$$

The *compatibility* between the preceding (like $\mathbf{v_a}$ and $\mathbf{v_b}$ in Fig. 11) and succeeding features (say, $\mathbf{v_A}$ and $\mathbf{v_B}$ in Fig. 11), i.e., the local-global interactions, is captured by the attention weights, and strengthened during model training. This way, ACroCI captures more co-flow features and cross-zone interactions than other network designs, yielding better accuracy in our experimental studies.

(d) *Output*: The capsule activation at the CC layer is done via another affine transformation, which outputs

$$\mathbf{X}_{\text{out}}^{(x)} = \sigma\left(\mathsf{Conv2D}(\mathbf{v})\right), \quad x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\}, \tag{11}$$

where tanh is used for the activation of $\sigma(\cdot)$. Then, we apply and feed $\mathbf{X}^{(x)} = \mathsf{Dense}\left(\mathbf{X}_{\text{out}}^{(x)}\right)$ $(x \in \{\mathrm{S}, \mathrm{M}, \mathrm{L}\})$ to the temporal learner.

3.2 Learning Temporal Interactions

The *Long Short-Term Memory* (LSTM) *encoder* first processes the input time-series and encodes the sequence into a vector representing the context. Let $\circ$ be the Hadamard product, i.e., the element-wise multiplication operation, t be the index of time step within the time-series, (f_t, i_t, o_t) be the output activation vectors from the forget, input/update and output gates [8], and $(\mathbf{c}_t, \mathbf{H}_t)$ be the cell and hidden states at t. ACroCI captures the temporal features via the LSTM structures as follows:

$$f_t = \sigma_g\left(\mathbf{W}_f\mathbf{X}^{(x)} + \mathbf{U}_f\mathbf{H}_{t-1} + \mathbf{b}_f\right), \quad i_t = \sigma_g\left(\mathbf{W}_i\mathbf{X}^{(x)} + \mathbf{U}_i\mathbf{H}_{t-1} + \mathbf{b}_i\right),$$

$$o_t = \sigma_g\left(\mathbf{W}_o\mathbf{X}^{(x)} + \mathbf{U}_o\mathbf{H}_{t-1} + \mathbf{b}_o\right),$$

$$\mathbf{c}_t = f_t \circ \mathbf{c}_{t-1} + i_t \circ \sigma_c\left(\mathbf{W}_c\mathbf{X}^{(x)} + \mathbf{U}_c\mathbf{H}_{t-1} + \mathbf{b}_c\right), \quad \mathbf{H}_t = o_t \circ \sigma_{\mathrm{H}}\left(\mathbf{c}_t\right),$$

where $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c, \mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_o$, and $\mathbf{U}_c$ are the weight matrices for the input and hidden states, and $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o$, and $\mathbf{b}_c$ are the biases, all of which are hyperparameters to be trained.

To further identify the multi-scale temporal dependency (i.e., **M2** in Sec. 2.2), we integrate within ACroCI the *temporal attention decoder* [3], given the encoded results from the LSTM encoder. By decoding the compressed information from the encoder, the attention decoder finds the locations where the most relevant features are concentrated. The temporal attention mechanism adaptively selects the more correlated hidden states of the encoder in order to match the sample features and produce the final output. Specifically, the output cell state $\mathbf{c}_t$ at the attention decoder is given by the weighted sum of the input mapping (hidden states) from the LSTM encoder of the previous inputs: $\mathbf{c}_t = \sum_{t'=1}^{T} \alpha_{(t',t)} \mathbf{H}_{t'}$. Similar to Eq. (8), the attention weight $\alpha_{(t',t)}$ in the temporal attention is given by a softmax function of

$$\alpha_{(t',t)} \triangleq \texttt{softmax}\left(e_{(t',t)}\right) = \frac{\exp\left(e_{(t',t)}\right)}{\sum_{k=1}^{T} \exp\left(e_{(t',k)}\right)}, \tag{12}$$

where $e_{(t',k)}$ represents a score of how the k-th heatmap frame $\mathbf{F}_k$ in the input sequence matches the hidden state $\mathbf{H}_{t'}$. Specifically, each score is characterized by a feed-forward neural network $\mathbb{F}(\cdot,\cdot)$ with concentration operation [3, 8], i.e.,

$$e_{(t',k)} \triangleq \mathbb{F}\left(\mathbf{H}_{t'}, \mathbf{F}_k\right) = \mathbf{v}_a^{\intercal} \sigma\left(\mathbf{W}_a[\mathbf{H}_{t'-1}; \mathbf{c}_{t'-1}] + \mathbf{U}_a \mathbf{F}_k + \mathbf{b}_a\right), \tag{13}$$

where $\mathbf{v}_a, \mathbf{b}$, $\mathbf{W}_a$, and $\mathbf{U}_a$ are all trainable parameters within the attention decoder, and we adopt tanh as activation function $\sigma(\cdot)$. Let $\mathbf{H}^{(\text{Out})}$ be the output of the last hidden states. The output of ACroCI's temporal learner, denoted as $\hat{\mathbf{X}}^{(\text{Main})} \in \mathbb{R}^M$, is an M-d vector of flows at all M APs, and is generated by the dense neural network following the hidden state output at the LSTM encoder, i.e.,

$$\hat{\mathbf{X}}^{(\text{Main})} = \texttt{Dense}\left(\mathbf{H}^{(\text{Out})}\right), \tag{14}$$

which is the final output of the spatio-temporal integration.

4 ADAPTIVITY MODULE INTEGRATION

4.1 Auxiliary Extreme-aware Learner

To enhance ACroCI's awareness towards the crowd interactions with the transient or emergent campus events (**M5** in Sec. 2.2), we have further designed an auxiliary extreme-aware crowd flow learner based on the statistical time-series features derived for ACroCI.

Processing Time-Series Features. For each campus zone we find the following 9 statistical time-series features of the recent $\omega^{(\text{ECF})}$ historical time intervals: *mean*, *variance*, *autocorrelation*, *entropy*, *trend* coefficients (total 3 features: *trend value* and its *p-value* via linear trend model estimation, and the *variance of the residuals*), *spike*, and *crossing points* (the number of times the mean is crossed by the time series) [7, 13, 15]. Compared with zone-level crowd flows in Eq. (2), these auxiliary features (each forms an $H \times W$ heatmap) provide additional information related to the flow dynamics.

In the meantime, using extensive empirical studies we label the corresponding Wi-Fi associations and disassociations for each AP i at an interval k, $Z_i^{(k)}$, with three conditions when compared with all historical readings at the same time interval (for instance, 8:00–9:00 of all historical Mondays): *(i)* $Z_i^{(k)} = 0$ when the value lies between 5th and 95th percentiles of all historical values (considered *normal*); *(ii)* $Z_i^{(k)} = 1$ when the value rises above 95th percentile (considered *surge*); and *(iii)* $Z_i^{(k)} = -1$ when the value falls below 5th percentile (considered *drop*). Then, for all APs at time interval k, we have $\mathbf{Z}^{(k)} = \left[Z_1^{(k)}, \ldots, Z_M^{(k)}\right]$ as the *auxiliary labels* in our multi-task extreme-aware learning design.

Learning Crowd Interaction with Events. For each campus zone, we find the aforementioned 9 features within a window of $\omega^{(\text{ECF})}$ consecutive time intervals, and the window rolls over the most recent Ω time intervals. Given the Ω most recent $H \times W \times 9$ tensors which form $\mathbf{F}^{(\text{stat})}$, we leverage ConvLSTM [8] to capture spatial and temporal correlations of the features (each heatmap of the 9 features as one channel), followed by a Dense layer (with ReLU for $\sigma(\cdot)$), and forecast the future AP-level crowd flows, denoted as $\hat{\mathbf{X}}^{(\text{ECF})}$. In order to enhance ACroCI's learnability regarding the extreme crowd flows, we design a *multi-task extreme-aware learning* mechanism (Sec. 4.2) which *jointly* captures and outputs both the AP-level crowd flows as well as the auxiliary labels, i.e.,

$$\left[\hat{\mathbf{X}}^{(\text{ECF})}; \hat{\mathbf{Z}}\right] = \sigma\left(\texttt{Dense}\left(\texttt{ConvLSTM}\left(\mathbf{F}^{(\text{stat})}\right)\right)\right). \tag{15}$$

This way, ACroCI correlates the surging, normal, and dropping crowd flows with the statistical features, fits the flows and auxiliary labels simultaneously, and thus enhances the adaptivity towards the transient or abnormal events.

4.2 Output and Model Training

External Learner. ACroCI also takes into account the external features (i.e., **M4** in Sec. 2.2) to assist in the crowd flow inference. Specifically, we normalize each dimension of the external factors into the range of $[0, 1]$, and concatenate them into a vector $\mathbf{E}$, which is fed to a neural network returning $\hat{\mathbf{X}}^{(\text{Ext})} \in \mathbb{R}^M$ as follows:

$$\hat{\mathbf{X}}^{(\text{Ext})} = \texttt{Dense}\left(\sigma\left(\texttt{Dense}\left(\mathbf{E}\right)\right)\right). \tag{16}$$

AP-level Learner. To integrate the most recent AP-level crowd flow features and further adapt to the latest transient crowd flow dynamics, we feed the crowd flows of the last time interval to another neural network, and predict $\hat{\mathbf{X}}^{(\text{AP})} \in \mathbb{R}^M$, i.e.,

$$\hat{\mathbf{X}}^{(\text{AP})} = \texttt{Dense}\left(\sigma\left(\texttt{Dense}\left(\sigma\left(\texttt{Dense}\left(\mathbf{X}^{(k-1)}\right)\right)\right)\right)\right), \tag{17}$$

where $\mathbf{X}^{(k-1)} \in \left\{\mathbf{A}^{(k-1)}, \mathbf{D}^{(k-1)}\right\}$. For Eqs. (16) and (17), we adopt ReLU for activation $\sigma(\cdot)$.

Integration, Output, and Training Objective Function. The final prediction of AP-level crowd flows, $\hat{\mathbf{X}} \in \mathbb{R}^M$, is given by the fusion of the outputs based on spatial-temporal learners ($\hat{\mathbf{X}}^{(\text{Main})}$), extreme crowd flows ($\hat{\mathbf{X}}^{(\text{ECF})}$), external factors ($\hat{\mathbf{X}}^{(\text{Ext})}$), and AP-level crowd flows ($\hat{\mathbf{X}}^{(\text{AP})}$),

$$\begin{aligned}\hat{\mathbf{X}} = \sigma\big(&\mathbf{W}^{(\text{Main})} \circ \hat{\mathbf{X}}^{(\text{Main})} + \mathbf{W}^{(\text{ECF})} \circ \hat{\mathbf{X}}^{(\text{ECF})} \\ &+ \mathbf{W}^{(\text{Ext})} \circ \hat{\mathbf{X}}^{(\text{Ext})} + \mathbf{W}^{(\text{AP})} \circ \hat{\mathbf{X}}^{(\text{AP})}\big),\end{aligned} \tag{18}$$

where $\mathbf{W}^{(\text{Main})}$, $\mathbf{W}^{(\text{ECF})}$, $\mathbf{W}^{(\text{Ext})}$, and $\mathbf{W}^{(\text{AP})}$ are all the learnable parameters which adjust the degrees affected by the different factors, and we also adopt ReLU for the activation function $\sigma(\cdot)$.

As discussed in Sec. 4.1, ACroCI is trained as *multi-task extreme-aware learning*, i.e., to *jointly* minimize the mean squared error between the predicted flows $\hat{\mathbf{X}}$ ($\hat{\mathbf{A}}$ or $\hat{\mathbf{D}}$) and the ground-truth $\mathbf{X}$, as well as the mean squared error between the predicted and ground-truth crowd flow labels, $\hat{\mathbf{Z}}$ and $\mathbf{Z}$, i.e.,

$$\texttt{Loss}(\boldsymbol{\theta}) = \left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_2^2 + \lambda \left\|\mathbf{Z} - \hat{\mathbf{Z}}\right\|_2^2, \tag{19}$$

where $\boldsymbol{\theta}$ represents all the trainable parameters within all the learners of ACroCI and $\lambda > 0$ is a weight parameter. We adopt the Adam optimizer [8] in our training.

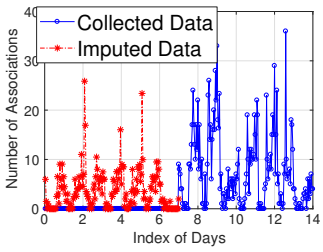


Fig. 12: Collected and imputed data for a newly-installed AP.

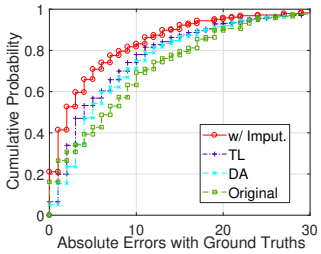


Fig. 13: Cumulative probability of prediction absolute errors.

4.3 Adapting to AP Alteration

As discussed in **M5** of Sec. 2.2, when an AP is newly added or relocated, there is no prior data for model training, leading to the "cold-start" issue. Thus, we design efficient training data imputation for each newly-installed (or relocated) AP using one preceding week's data from its nearby APs in the same zone. Specifically, for each target AP i, we find its geographic distances (in km), dist(i, j)'s ($j \in \{1, \ldots, M'\}$), from all its M' ($M' < M$) nearest APs already deployed in the same zone, and impute AP i's flows using weighted average for ACroCI's model training and initialization, i.e.,

$$\tilde{\mathbf{X}}_i^{(k)} = \sum_{j=1}^{M'} \frac{\beta_j}{\sum_{j=1}^{M'} \beta_j} \mathbf{X}_j^{(k)}, \quad \text{where} \quad \beta_j \triangleq \frac{1}{1 + \text{dist}\,(i, j)}. \tag{20}$$

We show in Fig. 12 one week's associations (blue) of a newly added AP (due to Campus A's facility reconstruction) after its deployment. We also conduct the data imputation (red) upon the earlier week (no association data) for the model training. We then show in Fig. 13 the improvement via our efficient data imputation, with mean absolute errors (MAEs) reduced from 9.02 (original) to 5.10, showing the applicability of the efficient data imputation. We also compare our approach with transfer learning (TL) [20] and domain adaptation (DA) [19] methods, and our efficient imputation more effectively adapts to such a "cold-start" scenario (~27% error reduction), as TL and DA still largely rely on availability of sufficient training data. We also note that ACroCI can also get timely model updates given newly accumulated association data for those APs after a few hours or days for further accuracy improvement.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Settings

Baseline Methods. Our proposed model is compared with the following conventional and state-of-the-art methods.

- HA and S-HA (**S**easonal HA): HA predicts the crowd flows as **h**istorical **a**verage of those of the same time periods or intervals (e.g., finding the average flow during 8:00–9:00 of all past Mondays to predict the same time slot for a Monday). *(a)* S-HA takes the same historical periods but of the same seasons.
- ARIMA, GP, RNN, LSTM, and DNN: Each of these approaches forecasts with *(3)* **a**uto-**r**egressive **i**ntegrated **m**oving **a**verage (ARIMA), *(4)* **G**aussian **p**rocess, *(5)* **r**ecurrent **n**eural **n**etwork (RNN) [8], *(6)* **l**ong **s**hort-**t**erm **m**emory (LSTM) [8], and *(7)* fully-connected **d**eep **n**eural **n**etwork (DNN).
- CNN and CNN-Att: only capture the spatial distributions of the associations with *(8)* **c**onvolutional **n**eural **n**etwork (CNN) [8] and *(9)* CNN with spatial **att**ention mechanism [4].
- GCN and MTGNN [27]: which leverage *(10)* **g**raph **c**onvolutional **n**eural network [14] and *(11)* **m**ultiple **t**ime-series **g**raph **n**eural **n**etwork, respectively, to predict the crowd flows.
- STCNN, STRNet, and STCaNet: process crowd distribution via *(12)* **s**patio-**t**emporal **c**onvolutional **n**eural **n**etwork [28], *(13)* **s**patio-**t**emporal **r**esidual neural **net**work [28], and *(14)* **s**patio-**t**emporal **ca**psule neural **net**work [9].
- CHAT [12]: is adapted to model the cross-event interactions in predicting the dynamics of the crowd flows.
- CSTN [16]: predicts with the convolution embedded LSTM-based method with a **c**ontextualized **s**patio-**t**emporal **n**etwork.
- T-LSTM: predicts the crowd flows based on **t**emporal pattern attention **l**ong **s**hort-**t**erm **m**emory [25].

Parameter Settings & Evaluation Metrics. Unless otherwise stated, we use the following parameters by default. We set the number of epochs to 1,000, batch size to 256, and learning rate to 0.001. We set 32×32 for heatmap frames on both campuses, 1h for time discretization, and $\omega = 6$ for each scale of temporal correlations. For the spatial learner of ACroCI, we set $\left(W^{(1)}, H^{(1)}, N^{(1)}, S^{(1)}\right) = (14, 14, 8, 8)$, $\left(W^{(2)}, H^{(2)}, N^{(2)}, S^{(2)}\right) = (7, 7, 32, 32)$ for ARCap, the number of filters to 32 for the two Conv2D(·) in Eqs. (4) and (11). For the temporal learner, we set the number of units to 8 for the LSTM modules, and the number of attentions to 16. For the auxiliary extreme crowd flow learner, we have empirically studied and set $\omega^{(\text{ECF})} = \Omega = 5$ and number of filters as 32 in ConvLSTM, and $\lambda = 0.1$ for the multi-task learning objective. For the external learner, we set the output dimension for $\mathbf{E}^0$ to 8. For the AP-level learner, we set the output dimension for $\mathbf{X}^0$ and $\mathbf{X}^1$ to 128 and 32, respectively. For the AP alteration adaptation, we use $M' = 5$ for Eq. (20).

For each campus dataset, we take the first 600 time intervals as the validation dataset and test upon the following 100 intervals to evaluate the model and parameter sensitivity. Apart from that validation dataset, for each campus, we conduct model training and testing on a monthly basis to emulate the real-world deployment. Specifically, we train the model based on data of 25 days (600 intervals in total) before each target test month. For each AP that is newly installed or relocated, we impute one-week data prior to its deployment for model training. We train and test the models through Python/Tensorflow using a GPU server with Intel Core i9 9900K, 32Gb RAM and two NVIDIA RTX 2080Ti 11Gb GPUs. The ACroCI model training time is 1.45s per epoch for Campus A, and 1.51s per epoch for Campus B; its model testing is fast (~0.92ms per time interval) for both datasets.

We evaluate all the schemes using the mean absolute error (MAE) to interpret the overall error *trend*, the root mean square error (RMSE) to demonstrate the *variance* of error distributions, and the poor case rate (PCR) to show the *relative scale* of prediction errors. The PCR is given by the percentage of all predictions which have the excessive over- or under-estimations compared to the ground-truth values, i.e., $\frac{1}{\mathbf{X}_i}|\mathbf{X}_i - \hat{\mathbf{X}}_i| \geq \eta$, where we consider $\eta = 0.6$.

5.2 Evaluation Results

Overall Performance, Rush Hours, and Extreme Events. Table 2(a) lists performance comparison of ACroCI with other algorithms for the two datasets. Overall, ACroCI achieves significantly higher accuracy than others. The conventional time-series-based techniques, (1)–(4), cannot capture the spatial correlations across the campus zones, hence achieving less accurate results. The sequence learning approaches like (5)–(7) cannot handle the spatial and temporal complexities within the crowd flows. Spatial learning approaches like (8)–(11) focus on the zone-to-zone correlations, and hence cannot handle multi-scale temporal complexity.

Table 2: Overall and rush-hour performance for Campuses A and B.

Schemes	(a) Overall Performance						(b) Rush-Hour Performance					
	Campus A			Campus B			Campus A			Campus B		
	MAE	RMSE	PCR	MAE	RMSE	PCR	MAE	RMSE	PCR	MAE	RMSE	PCR
(1)	5.65	9.98	0.313	7.86	12.92	0.346	9.33	13.01	0.447	12.29	18.18	0.508
(2)	4.35	8.19	0.251	7.84	12.30	0.336	8.72	11.92	0.413	11.43	17.17	0.477
(3)	6.15	9.04	0.304	7.57	11.79	0.323	9.24	11.02	0.405	10.08	13.06	0.386
(4)	6.75	10.82	0.351	8.76	12.99	0.363	9.72	13.52	0.465	11.28	14.21	0.425
(5)	3.45	6.76	0.204	6.34	8.98	0.255	6.39	9.59	0.320	10.12	13.68	0.397
(6)	2.97	6.66	0.193	6.26	8.72	0.250	6.85	8.57	0.308	10.18	12.31	0.375
(7)	5.05	7.61	0.253	6.86	9.52	0.273	9.59	10.94	0.411	9.41	11.75	0.353
(8)	2.74	5.75	0.170	6.38	8.37	0.246	3.16	7.29	0.209	8.54	10.03	0.310
(9)	2.89	6.06	0.179	5.27	7.52	0.213	3.42	7.06	0.210	7.23	9.75	0.283
(10)	2.99	4.74	0.154	7.49	10.14	0.294	3.94	8.33	0.245	9.13	12.00	0.352
(11)	3.13	5.82	0.180	6.02	9.11	0.252	3.69	7.74	0.229	8.82	11.40	0.337
(12)	2.53	5.17	0.164	5.68	8.00	0.228	2.89	5.74	0.173	7.61	9.41	0.284
(13)	2.45	4.77	0.144	5.27	7.22	0.208	2.79	5.93	0.174	7.07	9.52	0.277
(14)	2.32	4.57	0.138	5.17	7.16	0.206	2.86	6.87	0.195	7.12	9.46	0.276
(15)	2.35	4.75	0.142	5.12	7.10	0.204	2.97	6.62	0.192	7.26	9.58	0.281
(16)	3.12	6.16	0.186	6.17	9.71	0.265	3.63	7.94	0.231	8.92	11.70	0.344
(17)	2.95	5.73	0.174	6.12	9.12	0.254	3.29	6.73	0.200	7.06	10.52	0.293
ACroCI	**1.53**	**3.18**	**0.094**	**3.69**	**5.63**	**0.125**	**1.92**	**4.39**	**0.126**	**4.47**	**7.27**	**0.196**

ACroCI, however, formulates comprehensive spatio-temporal representations, and hence achieves respectively 51%~77%, 35%~70%, and 25%~51% accuracy improvements over the time-series, sequence learning and spatial learning approaches. Compared to prior efforts on spatio-temporal learning, e.g., (12)–(17), ACroCI captures the spatial heatmap frames via attention routing capsule, which *jointly* leverages vectorization of neural outputs to characterize spatial features, and identifies multi-scale temporal complexities. Furthermore, with the local and global cross-zone dependencies modeled, ACroCI demonstrates superior accuracy than scalar-based networks, leading to 21%~51% accuracy improvements.

We further pick the rush-hour periods (7:00–9:00, 12:00–14:00, and 16:00–18:00 of the weekdays) to evaluate the prediction robustness of ACroCI and other schemes. Table 2(b) shows higher MAEs, RMSEs, and PCRs during rush hours compared to the overall performance due to more dynamic and larger volumes of crowd flows. We can see that ACroCI still outperforms the other schemes by at least 22.7% for both datasets thanks to the multi-scale designs which adapt to the complex crowd flows.

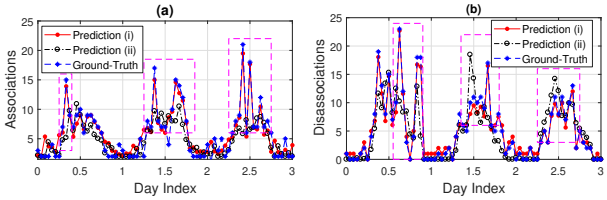


Fig. 14: Ground-truth & predictions for two APs.

We have further studied ACroCI's awareness to extreme crowd flows by showing the results of two APs ((a) & (b)) in an academic building. We compare in Fig. 14 their ground-truth time-series as well as the predicted ones by: *(i)* ACroCI with auxiliary extreme crowd flow learner, and *(ii)* ACroCI without it. We can observe from Fig. 14 that the case *(i)* achieves much better prediction results than case *(ii)* particularly for the extreme crowd flows (highlighted in pink boxes; ~31.5% MAE reduction). Though extreme crowd flows happen less frequently than normal ones, such adaptivity and extreme-aware improvements will benefit the campus management and response given the foreseen potential extreme events in campus crowd flows.

Model Ablation and Sensitivity Analysis. We have conducted ablation studies on the different components of ACroCI. Fig. 15 presents the performance of several ACroCI variations: *(a)* without entire **t**emporal **l**earner designs (w/o TL), *(b)* without **t**emporal **a**ttention (w/o TA), *(c)* without **ext**ernal factors (w/o ext), *(d)* without **a**ttention **r**outing designs (w/o AR), *(e)* without spatio-temporal representations (w/o **F**), *(f)* without **e**xtreme **c**rowd **f**low learner (w/o ecf), *(g)* without **m**odel **i**nitialization (w/o mi), and *(h)* with all components (w/ all). For w/o AR, we adopt the conventional dynamic routing [22] within the capsule neural network structure of ACroCI. We can see that the MAE benefits more from the representations, spatial attention routing, extreme crowd flow learner, and model initialization (~27% reduction), while the RMSE benefits more (~45% reduction) from the temporal learner (including the temporal attention). Our proposed representations and spatial attention in ARCap can help mitigate the overall error trends, and the temporal learner designs assist in adapting to crowd dynamics. Performance improvements from w/o AR also imply ACroCI's preeminence over conventional capsule-based approaches.

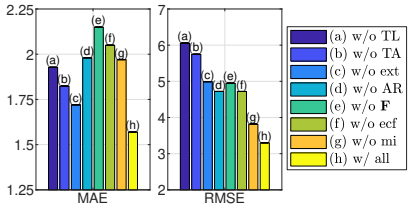


Fig. 15: Ablation studies on ACroCI's various components.

Case Studies & Interaction Visualization. We further focus on Campus A as we are more familiar with the local environments.

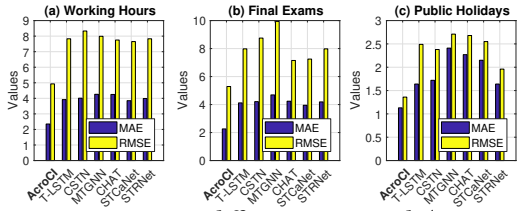


Fig. 16: Various cases at different time periods (Campus A).

Fig. 16 plots the performance of ACroCI and several state-of-the-art schemes, i.e., T-LSTM, CSTN, MTGNN, CHAT, STCaNet, and STRNet for three time periods: *(a)* working hours (8:00–17:00) of the weekdays (holidays excluded), *(b)* final exam weeks (one week in middle December), and *(c)* public holidays (including Thanksgiving Recess and Christmas). Compared to working hours, the mobility patterns during the final exams are more dynamic and hence larger errors can be observed. During the public holidays, all schemes have better performance due to lower crowd flow volumes. In all the above cases, ACroCI outperforms the other schemes by at least 26%, 31%, and 36%, respectively.

6 RELATED WORK

Wireless Mobility Analysis. Wireless mobility analysis has attracted much interest due to its business and social values. Qin et al. [21] mined the user behaviors based on mobile Wi-Fi usage inference. To the best of our knowledge, few of these efforts *predict* AP-level crowd flows using *adaptive* learning upon Wi-Fi association data in a campus network. We fill this gap by presenting an adaptive data learning system for accurate AP-level crowd flow prediction. In addition, we conduct comprehensive studies on how to deal with extreme flows and altered measurement environments, providing new insights for extreme-aware and adaptive wireless mobility analytics.

Various techniques have been proposed for group mobility sensing. Shen et al. [24] studied group detection based on the signal features of Wi-Fi probes; a similar study was conducted by Hong et al. [11]. Vision-based approaches [6, 26] can provide fine-grained crowd sensing, but their pervasiveness and applicability are subject to privacy concerns as well as environmental effects (e.g., none-line-of-sight and poor lighting). Unlike these techniques based on wireless/visual signals, we consider the campus Wi-Fi network setups as a case study and *forecast* crowd flows at each AP, using anonymized and less privacy-intrusive Wi-Fi association and disassociation data.

Crowd Mobility Modeling. Deep learning has recently been adopted to support various urban crowd mobility applications. For mobility learning, Zhang et al. [28] proposed leveraging a residual neural network to predict urban bike and taxi traffic. Huang et al. [12] studied the cross-interaction hierarchical attention networks for urban anomaly prediction. Scellato et al. [23] conducted a pioneering study that leverages time-series analysis models to predict the next location of a user based on her/his historical location visits. `ACroCI`, serving as an interactive wireless mobility data learning and crowd prediction system, differs from others as follows. `ACroCI` focuses on real-world Wi-Fi data learning, and is grounded on comprehensive and extensive data-driven system studies for *smart connected campus* application scenarios.

7 DEPLOYMENT DISCUSSION

Although one user may carry multiple mobile devices like laptops and smartphones, and certain pieces of office equipment like printers may connect to the campus Wi-Fi network, users can be easily identified by their IDs, allowing us to count the number of people (instead of devices) associated with an AP. Through such data pre-processing, we can use $\mathbf{A}^{(k)}$ and $\mathbf{D}^{(k)}$, at the user/client level, as reasonable indicators or ground-truths of user arrivals and departures. In future, we will make further improvements on the ground-truth labeling of arrivals and departures, e.g., calibrating session duration, handling unassociated devices and ping-pong effects, albeit beyond our current scope.

8 CONCLUSION

We have proposed `ACroCI`, a novel crowd mobility analytics system which models the crowds' interactions with the built environments with adaptive Wi-Fi data learning. Integrating with spatial crowd heatmaps, temporal flow dynamics, extreme features, environmental changes, and other external factors, `ACroCI` captures the crowds' interactions with built environments via a novel design of attention routing capsule network. Our extensive experimental studies have corroborated the accuracy, adaptivity, and robustness of `ACroCI`.

We would like to thank the University of Connecticut Information Technology Services (UITS) for their assistance in collecting Wi-Fi association data.

REFERENCES

[1] 2022. Weather Archive. https://rp5.ru/.
[2] Hamed S Alavi, Denis Lalanne, Julien Nembrini, Elizabeth Churchill, David Kirk, and Wendy Moncur. 2016. Future of human-building interaction. In *CHI*. 3408–3414.
[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[4] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. 2019. Attention augmented convolutional networks. In *ICCV*. 3286–3295.
[5] Jaewoong Choi, Hyun Seo, Suii Im, and Myungjoo Kang. 2019. Attention routing between capsules. In *IEEE ICCV Workshops*.
[6] W. Ge, R. T. Collins, and R. B. Ruback. 2012. Vision-Based Analysis of Small Groups in Pedestrian Crowds. *IEEE TPAMI* 34, 5 (2012), 1003–1016.
[7] M Ghil, P Yiou, Stéphane Hallegatte, BD Malamud, P Naveau, A Soloviev, P Friederichs, V Keilis-Borok, D Kondrashov, V Kossobokov, et al. 2011. Extreme events: dynamics, statistics and prediction. *Nonlinear Processes in Geophysics* 18, 3 (2011), 295–350.
[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
[9] Suining He and Kang G. Shin. 2019. Spatio-Temporal Adaptive Pricing for Balancing Mobility-on-Demand Networks. *ACM TIST* 10, 4 (2019).
[10] Suining He and Kang G Shin. 2019. Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination. In *The World Wide Web Conference*. 2806–2813.
[11] Hande Hong, Girisha Durrel De Silva, and Mun Choon Chan. 2018. CrowdProbe: Non-invasive crowd monitoring with Wi-Fi probe. *ACM IMWUT* (2018).
[12] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. 2020. Cross-Interaction Hierarchical Attention Networks for Urban Anomaly Prediction. In *IJCAI*.
[13] Rob J Hyndman, Earo Wang, and Nikolay Laptev. 2015. Large-scale unusual time series detection. In *IEEE ICDM Workshop*. 1616–1619.
[14] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[15] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. 2017. Time-series extreme event forecasting with neural networks at Uber. In *ICML Time Series Workshop*, Vol. 34. 1–5.
[16] Lingbo Liu, Zhilin Qiu, Guanbin Li, Qing Wang, Wanli Ouyang, and Liang Lin. 2019. Contextualized spatial–temporal network for taxi origin-destination demand prediction. *IEEE T-ITS* 20, 10 (2019), 3875–3887.
[17] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[18] Ljubica Pajevic, Gunnar Karlsson, and Viktoria Fodor. 2019. CRAWDAD dataset kth/campus (v. 2019-07-01). https://crawdad.org/kth/campus/20190701/wifi-mapping. https://doi.org/10.15783/c7-5r6x-4b46
[19] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2010. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* 22, 2 (2010), 199–210.
[20] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE TKDE* 22, 10 (2009), 1345–1359.
[21] Zhou Qin, Yikun Xian, Fan Zhang, and Desheng Zhang. 2020. MIMU: Mobile WiFi Usage Inference by Mining Diverse User Behaviors. *ACM IMWUT* 4, 4, Article 149 (Dec. 2020), 22 pages.
[22] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NIPS*. 3856–3866.
[23] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. 2011. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive*. Springer, 152–169.
[24] Jiaxing Shen, Jiannong Cao, and Xuefeng Liu. 2019. BaG: Behavior-Aware Group Detection in Crowded Urban Spaces Using WiFi Probes. In *WWW*. 1669–1678.
[25] Shun-Yao Shih, Fan-Keng Sun, and Hung-Yi Lee. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning* 108, 8 (2019), 1421–1441.
[26] F. Solera, S. Calderara, and R. Cucchiara. 2016. Socially Constrained Structural Learning for Groups Detection in Crowd. *IEEE TPAMI* 38, 5 (2016), 995–1008.
[27] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *ACM SIGKDD*. 753–763.
[28] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*. 1655–1661.

Building Thermal Dynamics Modeling with Deep Learning exploiting Large Residential Smart Thermostat Dataset

Han Li
hanli@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, USA

Giuseppe Pinto
Politecnico di Torino
Torino, Italy
giuseppe-pinto@polito.it

Alfonso Capozzoli
Politecnico di Torino
Torino, Italy
alfonso.capozzoli@polito.it

Tianzhen Hong*
thong@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, USA

ABSTRACT

In this paper, we present a deep learning approach to model building thermal dynamics with large-scale smart thermostat data collected from residential buildings. We developed a Long Short-Term Memory (LSTM) model as a baseline and compared it to a CNN-LSTM model to predict indoor air temperature in a multi-step time horizon in 164 buildings. The study showed that the proposed CNN-LSTM achieved an average of 0.26 °C Mean Absolute Error (MAE) for one-hour-ahead (12 future steps) predictions, which is over 6% of improvement comparing with the baseline. Furthermore, the results indicated that the CNN-LSTM models achieved more robust performance across different building characteristics, system configurations and locations, with a standard deviation reduction of 22%, proving the effectiveness and generalizability of the proposed approach.

CCS CONCEPTS

• **Applied computing** → **Physical sciences and engineering**; • **Software and its engineering** → *Software notations and tools.*

KEYWORDS

Building thermal dynamics, data-driven model, deep learning

ACM Reference Format:
Han Li, Giuseppe Pinto, Alfonso Capozzoli, and Tianzhen Hong. 2022. Building Thermal Dynamics Modeling with Deep Learning exploiting Large Residential Smart Thermostat Dataset. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3564056

1 INTRODUCTION

Building thermal dynamics models which predict future indoor air temperature given control actions, are essential for optimizing HVAC system controls and building energy management. Such thermal dynamics models are usually classified into white-box, grey-box, and black-box models. White-box models are based on first principles that govern the energy and mass transfer in buildings. Those models are developed in a forward approach where information about building geometry, energy systems, occupant behaviors, and weather conditions are known. However, despite their effectiveness, this often requires extensive expertise in developing and calibrating white-box models [3]. In contrast, grey-box and black-box models use the inverse approach, exploiting measured data to identify a model that describes a building's thermal processes. Grey-box models use lumped parameters to represent buildings' thermal properties. For example, as an analog to the electric circuit, thermal resistance (R) and thermal capacity (C) are two types of parameters of reduced-order models that describe the heat transfer in buildings [10]. Although these parameters can be identified using regressions with measured data, they often require simplified assumptions about external and internal loads. Furthermore, they could not directly utilize data that is recorded by smart thermostat, such as occupant motion detection and HVAC system runtime, which leads to a waste of information. On the other hand, black-box models are purely data-driven as they often do not need prior-knowledge about the building and can fully exploit the data provided by smart thermostat. In recent years, Deep Learning (DL) techniques have been extensively used in the built environment due to their ability to approximate complex dynamics. Wang et al. compared 9 ML algorithms for building thermal load prediction, and found that LSTM could achieve lower than 0.4 °C MAE for one-hour ahead predictions [11]. Pinto et al. developed LSTM thermal dynamics models of multiple buildings to support reinforcement learning based energy management [8]. Mtibaa et al. compared LSTM-based model architectures for indoor air temperature predictions [5]. Elmaz et al. proposed a CNN-LSTM model architecture for indoor air temperature predictions [2]. However, most existing studies developed models only for specific buildings and did not examine the model generalizability and consistency for buildings with various characteristics across different climate regions. Moreover, very few studies used real measurements for model training and testing, limiting the effectiveness of the proposed approaches. In this paper, we introduce a methodology to pre-process, enrich and exploit smart thermostat data that spans over different buildings in three U.S. states, three space types and two heating system

*Corresponding author: thong@lbl.gov

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3564056

configurations. Furthermore, we introduce a CNN-LSTM model trained and tested on over 150 residential buildings, with the aim to predict indoor air temperature over a multi-step horizon.

2 BACKGROUND

2.1 Deep Learning for Building Thermal Dynamics

Before the era of deep learning, most data-driven models for building thermal dynamics were linear and time-invariant [10]. As deep learning algorithms and computing resources became more and more available and mature in recent years, they have been increasingly applied in modeling building thermal dynamics due to their ability to approximate non-linearity and time-variance. The building indoor air temperature prediction problem involves multivariate inputs, sequential modeling, and multi-horizon outputs. A brief background of those three components is presented below:

- Feature extraction: it starts with the initial set of variables with the goal to extract a new set of processed variables that facilitate further pattern recognition. For multivariate time series data such as the thermostat measurements, 1D CNN is a commonly used technique.
- Sequential modeling: it receives sequential data such as time-series data, and outputs a single value or another sequence. Recurrent Neural Networks (RNNs), are a popular category of DL algorithms for sequential modeling. LSTM is a type of RNNs that specializes in both short-term and long-term memories [4]. It uses gating mechanisms that control non-linearity and information memory, and addresses the vanishing gradient issue in standard RNNs [9].
- Multi-horizon prediction: it forecasts the target variable for the next several steps at once. Depending on the model architecture, multi-horizon prediction can be classified into (1) iterative methods, where the single-step outputs and historical data are iteratively used for the next step prediction until the desired horizon is reached; (2) direct methods, where a complete sequence is output from the model and can be considered as sequence-to-sequence (seq2seq) methods. In this study, we used the direct methods to predict indoor air temperature.

3 METHODOLOGY

3.1 Data Processing

We used the smart thermostat data collected by ecobee's Donate Your Data (DYD) program where more than 190,000 households in the U.S. and Canada had voluntarily shared their data anonymously for research purposes as of 2022. Each thermostat has user-reported metadata about the building, including location (at city level), space type, gross floor area, number of floors, and time when the thermostat first connected. Figure 1 shows the three steps adopted to pre-process the data.

In step 1, we randomly sampled a subset of buildings using the building metadata. Specifically, the subset include buildings with three space types (i.e., apartment, townhouse, and detached single family houses), and two HVAC system configurations (i.e., with and without electric auxiliary heating) from three U.S. states with distinct climates (i.e., California, Texas, and New York). In step 2, we processed the time-series data for each building. To avoid the influence of behavioral change on building thermal dynamics due to COVID19 pandemic, we decided to only use a whole year of data in 2019. The raw time-series data includes information on the indoor environment and the HVAC systems, such as: indoor air temperature and humidity, cooling and heating setpoint temperature, supply fan runtime, cooling and heating system runtime, and occupant motion detections with five-minute temporal resolution. Furthermore, due to the high correlation of energy use with occupancy, we added temporal features such as time of the day, day of the week, and month of the year encoded as cosine and sine values, while differentiating between holidays with a binary encoding. Lastly, since the ecobee thermostat dataset does not include outdoor weather data, we added outdoor air temperature data to each thermostat using each thermostat's latitude and longitude to find the closest weather station listed by the National Oceanic and Atmospheric Administration (NOAA). In conclusion, a total of 23 features were used to describe the building thermal dynamics, which have been further standardized by scaling to unit variance using the scikit-learn package [7]. Table 1 shows the name, unit, and type of the variables used to train the deep learning models. The detailed descriptions and source code for data processing are available at the GitHub repository: https://github.com/tsbyq/EcoBee_BTD.

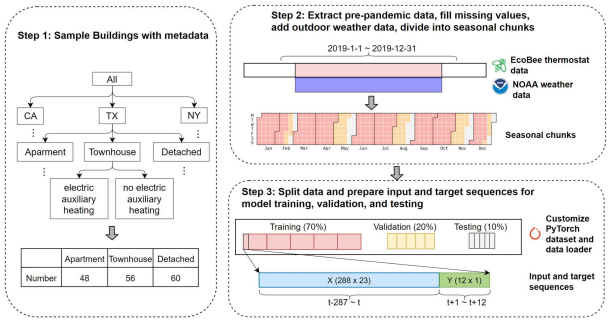


Figure 1: Data processing steps

3.2 Deep Learning Model Development

In this study, we implemented a simple LSTM model as the baseline, where data is directly input to the LSTM cell, followed by a linear layer for multi-horizon indoor temperature predictions. Then, we propose a modified version of the previous architecture, adding a 1D CNN module before the LSTM cell for feature enhancing. Figure 2 shows the proposed CNN-LSTM architecture. Both models are implemented using PyTorch [6]. We then performed a sensitivity analysis on models hyperparameters, optimizer, and learning rate scheduler, which will be presented in section 4.

3.3 Performance Metrics

In this study, we used MAE for evaluation purposes, since it has easy-to-interpret physical meaning (°C deviations) and increases steadily as the error grows. We also used performance improvement ratio (PIR) to quantify the relative performance improvement. The formula of MAE and PIR are shown below.

Variable Name	Meaning	Type	Unit
TemperatureExpectedCool	thermostat cooling setpoint	numerical	°C
TemperatureExpectedHeat	thermostat heating setpoint	numerical	°C
Humidity	relative humidity	numerical	%
auxHeat1	auxiliary heating system 1 runtime	numerical	seconds/5minutes
auxHeat2	auxiliary heating system 2 runtime	numerical	seconds/5minutes
auxHeat3	auxiliary heating system 3 runtime	numerical	seconds/5minutes
compCool1	cooling compressor 1 runtime	numerical	seconds/5minutes
compCool2	cooling compressor 1 runtime	numerical	seconds/5minutes
compHeat1	heating compressor 1 runtime	numerical	seconds/5minutes
compHeat2	heating compressor 1 runtime	numerical	seconds/5minutes
fan	supply air fan runtime	numerical	seconds/5minutes
Thermostat_Temperature	aggregated thermostat temperature	numerical	seconds/5minutes
Thermostat_Motion	occupant presence	binary	N.A.
T_out	outdoor air temperature from NOAA	numerical	°C
sin_hour	sine of an hour in a 24-hour day	numerical	N.A.
cos_hour	cosine of an hour in a 24-hour day	numerical	N.A.
sin_day_of_week	sine of an day in a 7-day week	numerical	N.A.
cos_day_of_week	cosine of an day in a 7-day week	numerical	N.A.
sin_month	sine of an day in a month	numerical	N.A.
cos_month	cosine of an day in a month	numerical	N.A.
sin_week_of_year	sine of a week in a 52-week year	numerical	N.A.
cos_week_of_year	cosine of a week in a 52-week year	numerical	N.A.
is_holiday	whether a day is holiday	binary	N.A.

Table 1: Time-series data variables

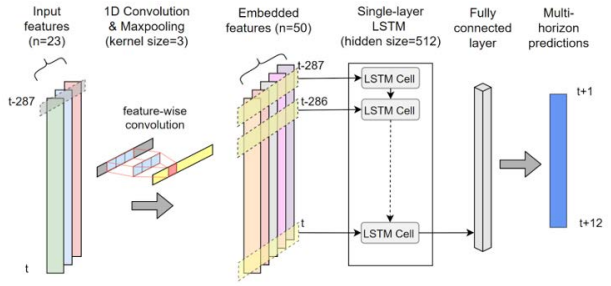


Figure 2: CNN + LSTM model architecture

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{1}$$

$$PIR = \frac{MAE_{baseline} - MAE_{new}}{MAE_{baseline}} \times 100\% \tag{2}$$

4 EXPERIMENT AND RESULTS

4.1 Model Training

Deep learning model performance are highly influenced by hyperparameters settings; with this in mind, we used the Optuna hyperparameter optimization framework [1] to search for the best hyperparameters. The goal is to find the combination of hyperparameters that lead to the best MAE on a randomly selected subset from the whole dataset (10 homes). Among the hyperparameters selected for the optimization include neural network architecture (number of LSTM layers and convolutional kernel size), learning rate with a cosine annealing scheduler to gradually decrease the value, and Adam optimizer. The hyperparameter search space and training configurations for machine learning are shown in Table 2. The optimization process then starts with random sampling from the search space, and tries to improve using an evolutionary optimization approach minimizing a loss function (mean squared error (MSE)). The analysis highlighted how model architecture parameters: CNN kernel size, LSTM hidden size, together with learning rate, have the greatest influence on model performance, while the other hyperparameters only have marginal impacts. Therefore, we chose a single-layer LSTM with no dropout for the sequential model.

Hyperparameter	Distribution	Range	Selected
learning rate	log uniform	[2e-4, 2e-2]	2e-3
Adam optimizer weight decay	log uniform	[1e-6, 1e-4]	1e-5
Conv1D kernel size	discrete with step=32	[32, 256]	50
LSTM number of layers	discrete with step=1	[1, 4]	1
LSTM hidden size	discrete with step=128	[128, 1024]	512
LSTM dropout probability	discrete with step=0.1	[0, 0.8]	0
batch size	discrete with step=128	[128, 1024]	512
number of epochs	N.A.	N.A.	60

Table 2: Hyperparameter search space and selected values

We trained our models with an NVIDIA Titan RTX graphic card with 24GB graphics RAM, using mixed precision training with half precision floating point numbers enabled by PyTorch's automatic

mixed precision (AMP) package, which provided 30% speedup compared with full-precision training, leading to a simulation time of minutes per model. Furthermore, due to the relatively simple CNN feature extraction, we did not observe significant time differences in training the vanilla LSTM and CNN-LSTM model.

4.2 Results and Discussions

We assessed the performance of machine learning models using the 10% test data discussed in section 3.2. The dataset contains 48 apartments, 56 townhouses, and 60 single family houses in California, Texas, and New York. We evaluated the performance of machine learning models from two aspects: (1) comparison of vanilla LSTM and CNN-LSTM model performance in general and by different prediction horizon, (2) prediction accuracy of CNN-LSTM models by different seasons, building locations, types, and HVAC system configurations.

Figure 3 shows the comparison of MAE distribution of different prediction horizons between the vanilla LSTM and CNN-LSTM models. It can be seen that except for the first three prediction steps (t+1 t+3), CNN-LSTM models achieved lower average MAE than vanilla LSTM, with 6.6% overall PIR for all prediction steps. Furthermore, the standard deviation of MAE of CNN-LSTM models are 22% lower than vanilla LSTM models except for the first prediction step (t+1), meaning they achieved more consistent performance than vanilla LSTM for most prediction steps. In summary, CNN-LSTM models performed better than vanilla LSTM models especially for longer-term predictions.

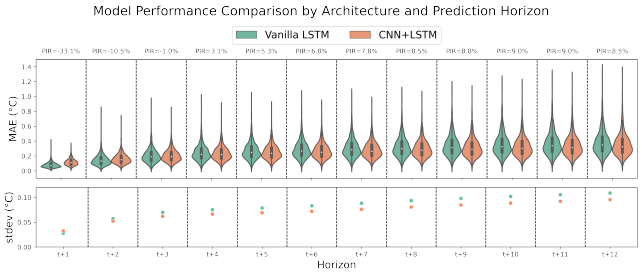


Figure 3: Vanilla LSTM vs CNN-LSTM performance comparison by prediction horizon

We then looked into the robustness of the CNN-LSTM models by breaking down the performance on all buildings by different seasons, locations, building types, and whether there is electric auxiliary heating in the building. The results shows that 98% of the CNN-LSTM models have an average MAE of less than 0.5 °C for the entire prediction horizon, which are adequate for applications such as thermal load prediction and optimal control. The proposed model generalized well as we do not see significant differences among seasons and building characteristics among most models.

5 CONCLUSIONS

In this study, we proposed a deep learning approach for multi-horizon indoor air temperature prediction using large scale smart thermostat data from residential buildings. The dataset we used in this study includes 164 buildings from three U.S. states with three building types and two HVAC system configurations. We developed a data processing pipeline which could support large-scale time-series forecasting and analytics using the ecobee dataset in the future. Overall, our proposed CNN-LSTM models achieved an average MAE of 0.26 °C for 1-hour-ahead (12-step-ahead) predictions, which is 6.6% better than vanilla LSTM models. We also investigated the model performance breakdown by different seasons, building types, locations, and HVAC system configurations. The results suggested that our proposed models can generalize well across residential buildings with different characteristics, maintaining the fast inference speed, that can support model predictive control (MPC) or deep reinforcement learning (DRL) control applications, where such speed is required.

Several future research opportunities exist beyond on this study. Firstly, we will look into DL models with longer-horizon predictions and extra covariates that influence the indoor thermal conditions such as occupancy and weather forecast. More sophisticated model architectures including attention-based seq2se2 models will be evaluated. Furthermore, we can investigate the effectiveness of transfer learning in presence of large amount of real building metadata and thermostat data, whose findings could help model-based controller deployments for buildings with newly connected thermostats.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[2] Furkan Elmaz, Reinout Eyckerman, Wim Casteels, Steven Latré, and Peter Hellinckx. 2021. CNN-LSTM architecture for predictive indoor temperature modeling. *Building and Environment* 206 (2021), 108327.

[3] Enrico Fabrizio and Valentina Monetti. 2015. Methodologies and advancements in the calibration of building energy models. *Energies* 8, 4 (2015), 2548–2574.

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[5] Fatma Mtibaa, Kim-Khoa Nguyen, Muhammad Azam, Anastasios Papachristou, Jean-Simon Venne, and Mohamed Cheriet. 2020. LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings. *Neural Computing and Applications* 32, 23 (2020), 17569–17585.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[8] Giuseppe Pinto, Davide Deltetto, and Alfonso Capozzoli. 2021. Data-driven district energy management with surrogate models and deep reinforcement learning. *Applied Energy* 304 (2021), 117642.

[9] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.

[10] Zequn Wang and Yuxiang Chen. 2019. Data-driven modeling of building thermal dynamics: Methodology and state of the art. *Energy and Buildings* 203 (2019), 109405.

[11] Zhe Wang, Tianzhen Hong, and Mary Ann Piette. 2020. Building thermal load prediction through shallow machine learning and deep learning. *Applied Energy* 263 (2020), 114683.

The Impact of Resolution of Occupancy Data on Personal Comfort Model-Based HVAC Control Performance

Eikichi Ono
Kajima Technical Research Institute
Japan
onoe@kajima.com

Yue Lei
National University of Singapore
Singapore
bdgleiy@nus.edu.sg

Kuniaki Mihara
Kajima Technical Research Institute Singapore
Singapore
k.mihara@kajima.com.sg

Adrian Chong
National University of Singapore
Singapore
adrian.chong@nus.edu.sg

ABSTRACT

Data-driven thermal comfort models play an important role in occupant-centric HVAC controls to satisfy occupants' preferences in thermal comfort. Although personal comfort models are likely to predict occupants' preferences well, the performance of personal comfort model-based controls might be affected by the resolution of occupancy data when incorporating them into a lower resolution of control (i.e., group- or zone-level control). We conducted a field experiment at a university office air-conditioned by a group-level control to evaluate the thermal comfort performance of personal comfort model-based optimal controls with different resolutions of occupancy data (personal- and group-level). The results showed that using personal-level occupancy data increased the percentage of "comfortable" votes by 8% compared to group-level occupancy data. Even group-level controls can behave similarly to personal-level control in partially-occupied situations. This highlights the importance of selecting an appropriate resolution of occupancy data and thermal comfort models for occupant-centric HVAC controls.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in interaction design**; • **Computing methodologies** → *Control methods.*

KEYWORDS

occupant-centric HVAC control, thermal comfort model, occupancy data

ACM Reference Format:
Eikichi Ono, Yue Lei, Kuniaki Mihara, and Adrian Chong. 2022. The Impact of Resolution of Occupancy Data on Personal Comfort Model-Based HVAC Control Performance. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3564061

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3564061

1 INTRODUCTION

Recent years have seen a growing interest in data-driven thermal comfort models as researchers have recognized the importance of considering actual occupants' preferences in thermal comfort. It is crucial to incorporate data-driven thermal comfort models into an occupant-centric HVAC control framework that aims to meet occupants' thermal demand [6]. Recent studies have increasingly focused on developing and utilizing personal comfort models since they are likely to capture individual differences in thermal comfort well [8]. However, it is often necessary to mediate a conflict between different thermal demands from occupants in practice since most commercial buildings adopt zone-level VAV systems [12]. Consequently, the resolution of personal comfort models is sometimes reduced from "personal-level" to "zone-level" (e.g., zone-level comfort profile) to incorporate into zone-level controls [2].

When combining personal comfort models with HVAC controls with a lower resolution (i.e., zone-level or group-level), the mismatch of different resolutions might cause a potential loss in the performance improvement of occupant-centric HVAC controls since the HVAC controls cannot fully exploit the capability of personal comfort models [11]. Nonetheless, if personal-level occupancy detection is combined with personal comfort models, even lower resolutions of HVAC controls might behave similarly to personal control in partially-occupied situations. Previous studies incorporating personal comfort models into zone-level optimal HVAC controls have assumed full occupancy [2, 5] or personal-level detection [7] to compute an optimal setpoint. However, an understanding of the impact of different resolutions of occupancy data on the control performance is still lacking.

This paper presents the results of a field experiment conducted to understand the impact of occupancy resolution of occupancy data when combined with personal comfort models for occupant-centric HVAC control. We compare the thermal comfort performance between different occupancy resolutions (personal and group levels) of occupancy data with personal comfort models incorporated into an optimal control framework.

2 METHOD

Figure 1 illustrates the framework of a field experiment to investigate the impact of occupancy resolutions of occupancy data with personal comfort models on HVAC control performance. The experiment is conducted at a researcher's office in the SDE4 building

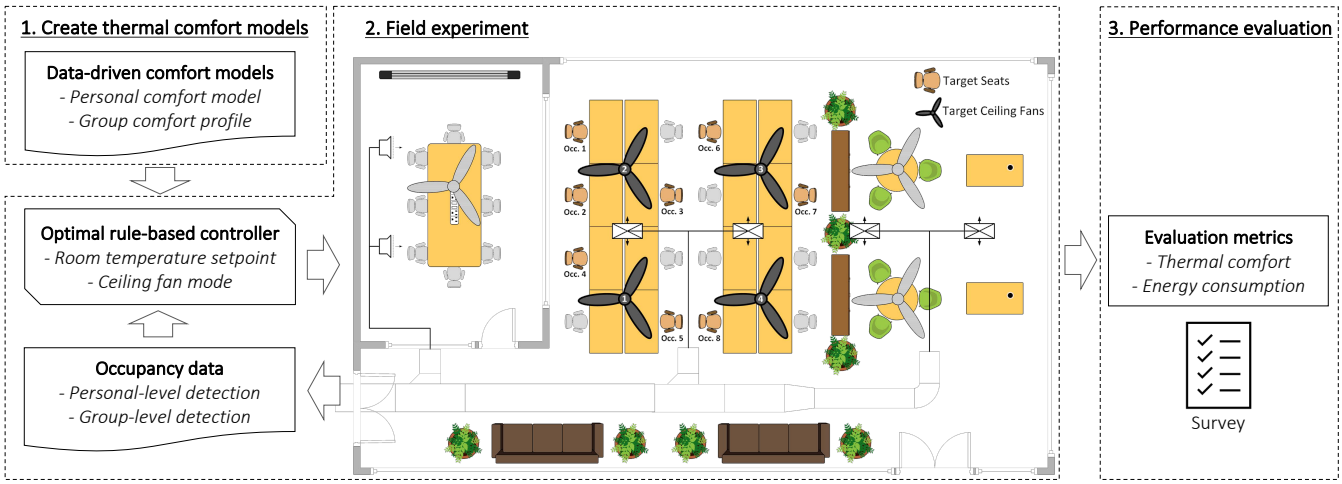


Figure 1: Framework of the field experiment; Prior to the experiment, personal comfort datasets for eight occupants are collected by a survey to create data-driven thermal comfort models. During the experiment, the room temperature setpoint and ceiling fan mode are optimized to maximize thermal comfort and minimize energy consumption based on optimal control rules, thermal comfort models, and occupancy data.

of the National University of Singapore (NUS). The space is air-conditioned by a dedicated outdoor air system with ceiling fans (DOAS-CF) [9], which can achieve energy savings by increasing indoor temperature while maintaining thermal comfort by elevated air movement. We recruited eight occupants who normally spend most of their time in the office, including six males and two females from different countries between twenty and forty years old. There are four ceiling fans in the target area. Thus, each ceiling fan is shared by 1–3 occupants (i.e., group-level control). The experiment has been approved by the Institutional Review Board of NUS in terms of a study involving human subjects (NUS-IRB-2021-157).

2.1 Data-driven thermal comfort models

A comfort experiment is conducted under different combinations of indoor temperature, 26–28 °C, and ceiling fan mode, 0–5, which correspond to 0.08–1.13 m/s of the mean air velocity at three heights (0.1, 0.75, and 1.1 m above floor). Real-time comfort surveys based on the Bedford scale (too cool, cool, comfortably cool, comfortable, comfortably warm, warm, and too warm) are carried out in each experiment condition to create data-driven comfort models.

Based on the survey results, personal comfort models and group comfort profiles are created using the following four steps referring to Jung and Jazizadeh [5]:

(1) The votes are categorized into three groups: cooler ("too cool," "cool," and "comfortably cool"), comfortable ("comfortable"), and warmer ("comfortably warm," "warm," and "too warm").
(2) Normal distributions are determined based on each group's relationship between the votes and the Standard Effective Temperature (SET*).
(3) Personal comfort models predict a probability of being comfortable by a Bayesian network, assuming that overall comfort can be derived from three comfort states corresponding to the three normal distributions.
(4) Group comfort profiles are then created by averaging and normalizing the outputs of personal comfort models for the occupants belonging to each ceiling fan group.

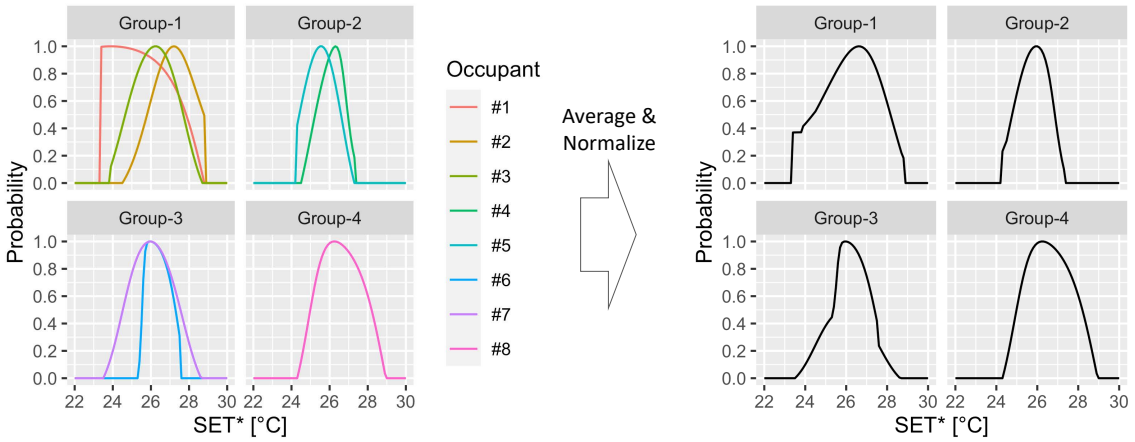
From the comfort survey, we collected 18–36 votes for each occupant (242 in total). Figure 2 shows the data-driven comfort models created based on the survey results. Personal comfort models represent the variance in occupants' preferences in thermal comfort. In contrast, group comfort profiles lose individual differences.

2.2 Optimal rule-based control

We create optimal control rules that can be easily implemented as optimal rule-based control (optimal RBC) for the field experiment [10]. Firstly, optimization is conducted with `EnergyPlus` in `R` using the `eplusr` package [4]. The objective functions are (1) to minimize energy consumption and (2) to maximize the number of occupants whose probabilities of being comfortable are equal to or greater than 80%. The optimization parameters, indoor temperature and ceiling fan mode, are optimized under various conditions, including outdoor climate and internal gains. From the optimization results, optimal control rules are then extracted using a linear tree model.

2.3 Field experiment

The field experiment was conducted for 15 days between 14th September and 10th November in 2021. The three experimental cases shown in Table 1 were randomly allocated to each day to minimize the differences in occupancy and outdoor conditions. The numbers of experimental days were 6, 5, and 4 days for baseline, group-optimal RBC, and personal-optimal RBC, respectively. The operational conditions for the baseline were fixed at 27 °C and ceiling fan mode = 3, referring to the optimal setpoint suggested by Mihara et al. [9]. For optimal RBCs, indoor temperature and ceiling fan mode were changed at 30-minute intervals according to the optimal control rules and the manually counted occupancy.



(a) Personal comfort model (b) Group comfort profile

Figure 2: Data-driven thermal comfort models. Personal comfort models are derived from the original comfort dataset. Group comfort profiles are created by averaging and normalizing personal comfort models' outputs.

Table 1: Experimental cases.

Case name	Description
Baseline	Constant at 27 °C and fan mode 3
Group-optimal RBC	Optimal control based on group-level occupancy detection and group comfort profiles
Personal-optimal RBC	Optimal control based on personal-level occupancy detection and personal comfort models

Personal-optimal RBC used personal-level occupancy data and personal comfort models, while group-optimal RBC combined group-level occupancy data with group comfort profiles. Real-time comfort surveys were carried out at hourly intervals between 10:00 AM and 6:00 PM. Note that we calculated the energy savings by combining the two optimal RBCs' periods since it is not easy to estimate the energy consumption for the target space from short-term data collection as the DOAS serves other spaces in the building.

3 RESULTS AND DISCUSSION

Figure 3 shows operational conditions on representative days which had similar mean outdoor temperatures at around 30 °C. The indoor temperatures for the two optimal RBCs were almost the same at 27.5–28.0 °C, while it was around 27.0 °C for the baseline. The difference between the two optimal RBCs can only be observed in the ceiling fan modes. Using personal-level occupancy data with personal comfort models, the ceiling fan mode more dynamically changed according to the preferences of occupants who were present at that time. The energy savings from the baseline to optimal RBC was 16%. This was attributed to the increased indoor temperature. Interestingly, although the optimal control was conducted based on the optimal control rule derived from simulation-based optimization, an optimal control strategy can be simply defined as (1) increasing indoor temperature to 28.0 °C to save HVAC energy and (2) adjusting ceiling fan mode according to the occupants' preferences to increase thermal comfort. The simple strategy can stem from stable

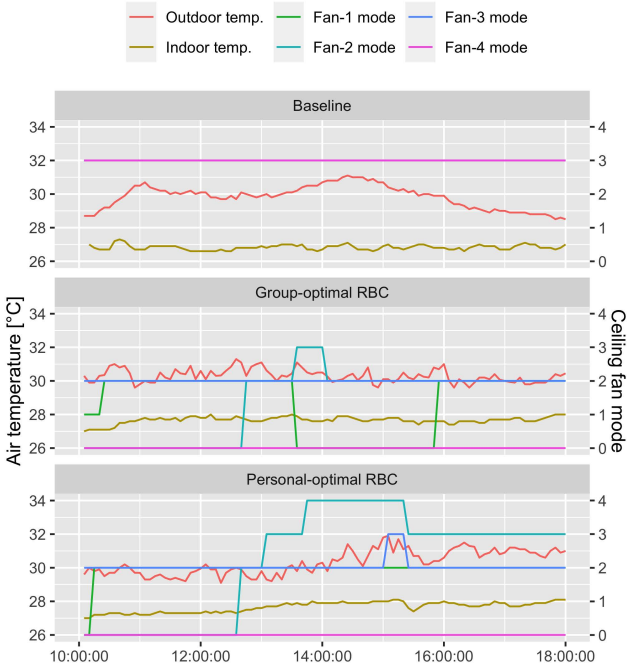


Figure 3: Indoor temperature and ceiling fan modes in a representative day. This illustrates the difference in operational conditions between different control strategies.

weather conditions in Singapore, stable operational conditions in the building, simplicity of the HVAC system, and much smaller energy consumption of ceiling fans than the air handling unit and chillers. This indicates that a complicated optimization framework, for instance, using simulation-based optimization, model-predictive control, or reinforcement learning-based control, may not always be necessary for realizing optimal operation.

Figure 4 depicts occupants' feedback on thermal comfort. For the baseline, some occupants felt slightly cool, although they were still

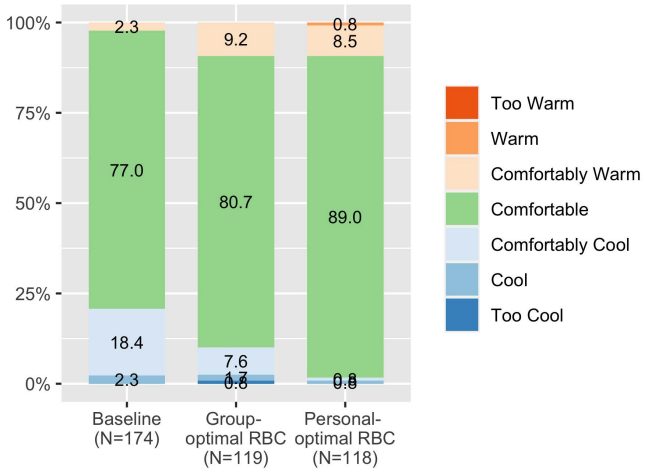


Figure 4: Comparison of thermal comfort performance evaluated by subjective feedback.

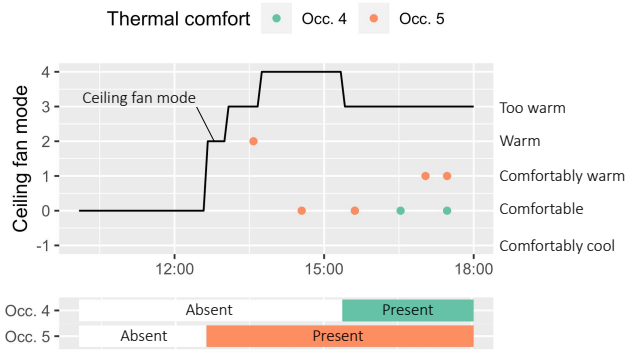


Figure 5: Ceiling fan mode, occupancy status, and thermal comfort votes for group-2 with personal-optimal RBC. The ceiling fan mode changed according to the occupancy status.

mostly "comfortably cool". Group- and personal-optimal RBCs increased the percentage of "comfortable" by 4 and 12% from the baseline, respectively. The 8% difference between group- and personal-optimal RBCs could be attributed to how well the optimal RBCs can capture the preferences of occupants present under a ceiling fan. As illustrated in Figure 5, personal-optimal RBC had adjusted the ceiling fan mode to meet the preference of Occ. 5 until Occ. 4 appeared. Consequently, Occ. 5 felt "comfortable" (except for the vote just after arrival) before Occ. 4 arrived, but then changed to feel "comfortably warm."

It is inherently difficult for group-level controls like ceiling fans to meet the preferences of multiple occupants. However, the results suggest that group-level controls combined with personal-level occupancy data and personal comfort models can improve operational performance similar to personal-level controls in partially-occupied situations. This highlights that selecting a resolution of occupancy data suitable for thermal comfort models and HVAC systems is crucial. Similar findings were reported for the relationships between occupancy data and occupant-centric controls [3], thermal comfort models and occupant-centric controls [11], and occupancy data and building energy model calibration [1].

4 CONCLUSION

This work presented the results of a field experiment to evaluate optimal rule-based controls combined with different resolutions of occupancy data and thermal comfort models. We then discussed the impact of the resolutions on occupant-centric HVAC control performance. A significant finding is that combining personal-level occupancy data with personal comfort models can improve HVAC control performance. Even group-level controls can behave similarly to personal-level control in partially-occupied situations. On the contrary, combining group-level occupancy data with personal comfort models might result in a potential loss in performance improvement. The results showed that personal-optimal RBC increased the percentage of "comfortable" votes by 8% compared to group-optimal RBC. This implies that the occupancy resolutions not only for the thermal comfort models but also for the inputs (e.g., occupancy data) and control devices (e.g., HVAC and lighting) must be considered when designing occupant-centric controls. Since the sample size was limited for our experiment, further studies will be needed to thoroughly understand the relationships between occupancy data, thermal comfort models, and occupant-centric controls' performance.

ACKNOWLEDGMENTS

This research is supported by Kajima Corporation through its Kajima Technical Research Institute Singapore (KaTRIS) (Project No. A-0008298-00-00).

REFERENCES

[1] Adrian Chong, Godfried Augenbroe, and Da Yan. 2021. Occupancy data at different spatial resolutions: Building energy performance and model calibration. *Applied Energy*, 286, 116492.

[2] Ali Ghahramani, Farrokh Jazizadeh, and Burcin Becerik-Gerber. 2014. A knowledge based approach for selecting energy-aware and comfort-driven HVAC temperature set points. *Energy and Buildings*, 85, 536–548.

[3] Brodie W. Hobson, Brent Huchuk, H. Burak Gunay, and William O'Brien. 2021. A workflow for evaluating occupant-centric controls using building simulation. *Journal of Building Performance Simulation*, 14, 6, 730–748.

[4] Hongyuan Jia and Adrian Chong. 2021. eplusr: A framework for integrating building energy simulation and data-driven analytics. *Energy and Buildings*, 237, 110757.

[5] Wooyoung Jung and Farrokh Jazizadeh. 2020. Energy saving potentials of integrating personal thermal comfort models for control of building systems: Comprehensive quantification through combinatorial consideration of influential parameters. *Applied Energy*, 268, 114882.

[6] Joyce Kim, Stefano Schiavon, and Gail Brager. 2018. Personal comfort models – A new paradigm in thermal comfort for occupant-centric environmental control. *Building and Environment*, 132, 114–124.

[7] Da Li, Carol C. Menassa, and Vineet R. Kamat. 2017. Personalized human comfort in indoor building environments under diverse conditioning modes. *Building and Environment*, 126, July, 304–317.

[8] Larissa Arakawa Martins, Veronica Soebarto, and Terence Williamson. 2022. A systematic review of personal thermal comfort models. *Building and Environment*, 207, 108502.

[9] Kuniaki Mihara, Chandra Sekhar, Yuichi Takemasa, Bertrand Lasternas, and Kwok Wai Tham. 2019. Thermal comfort and energy performance of a dedicated outdoor air system with ceiling fans in hot and humid climate. *Energy and Buildings*, 203, 109448.

[10] Eikichi Ono, Kuniaki Mihara, Takamasa Hasama, Yuichi Takemasa, Bertrand Lasternas, and Adrian Chong. 2021. Investigating the relationship between interpretability and performance for optimal rule-based control. In *Proceedings of the 17th IBPSA building simulation conference*, 2562–2569.

[11] Eikichi Ono, Kuniaki Mihara, Khee Poh Lam, and Adrian Chong. 2022. The effects of a mismatch between thermal comfort modeling and HVAC controls from an occupancy perspective. *Building and Environment*, 220, 109255.

[12] Jiaqing Xie, Haoyang Li, Chuting Li, Jingsi Zhang, and Maohui Luo. 2020. Review on occupant-centric thermal comfort sensing, predicting, and controlling. *Energy and Buildings*, 226, 110392.

Towards smartwatch-driven just-in-time adaptive interventions (JITAI) for building occupants

Clayton Miller
clayton@nus.edu.sg
National University of Singapore

Yun Xuan Chua
chuayunxuan@nus.edu.sg
National University of Singapore

Mario Frei
mario.frei@nus.edu.sg
National University of Singapore

Matias Quintana
maqr@nus.edu.sg
National University of Singapore

ABSTRACT

Building occupants are complex in the aspects of the indoor environment that satisfies them. Thermal comfort-driven systems control, office floor plan design, and lighting or acoustics considerations are all driven by conventional understandings of how occupants use spaces. Building operators rely on the use of smart system control; however, these approaches ignore the occupants' ability to make decisions that influence their comfort, satisfaction, and productivity. The future-of-work paradigm shift will introduce more flexibility and behavior-driven opportunities for occupants. The emerging just-in-time adaptive intervention (JITAI) concept is gaining momentum in fields such as mobile health as a means of influencing behavior. This paper outlines a novel methodology using smartwatch-based JITAI combined with micro-ecological momentary assessments (EMA) for field-based data collection and occupant behavior interventions in field-based scenarios. A proof-of-concept deployment of a single user of the methodology is illustrated to set the stage for future data collection. This framework can provide the foundation for techniques to enhance the future of work through behavioral interventions in the built environment.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Thermal comfort, Noise, Distraction, Wearable devices, Occupant behavior

ACM Reference Format:
Clayton Miller, Yun Xuan Chua, Mario Frei, and Matias Quintana. 2022. Towards smartwatch-driven just-in-time adaptive interventions (JITAI) for building occupants. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3566135

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3566135

1 INTRODUCTION

Building occupant satisfaction with their indoor environment is not optimal. A study of open office plans shows many occupants are dissatisfied with noise distractions, among other issues [10]. Noise issues can cause distraction and productivity decreases that occupants have to cope with [1]. Occupants of open office spaces often resort to using headphones to mask background noises to reduce distraction [8]. Thermal comfort is also an important issue as studies show that almost half of occupants are not satisfied [7], and the most commonly used prediction models are only accurate one-third of a time [5]. Recent work in adaptive modeling of thermal comfort [14, 20], personalized comfort systems [3], and the characterization of individual attributes that influence thermal comfort perception [15, 19] have made progress in the field. However, many challenges remain in understanding how thermal comfort can be improved for buildings.

The development of modern digital devices such as smartphones and smartwatches [9] have increased the ability and cost-effectiveness of collecting data from building occupants [12]. The convergence of these data streams with sensors from the built environment creates the opportunity for the personalization of comfort prediction models [2]. The use of these models provides innovation in the control of building systems. Still, there are further uses of these personalized models related to influencing the behavior of building occupants.

1.1 Just-in-time adaptive interventions (JITAI)

Just-in-time adaptive interventions (JITAI) is a method pioneered in the psychology and public health fields as a way to give people the right information at the right time to make improved decisions related to their well-being. The most prominent application of this technique targets the encouragement of people to increase their physical activity throughout the day, which would support healthier lifestyles [13]. These studies collect information about a person's habits related to how they get physical activity throughout the day and attempt to influence their behavior by sending intervention messages at a targeted time to encourage them to move [16]. This research has resulted in real-world implementations of messages in fitness-focused wearable devices on the market.

This paper outlines the adaptation of a platform to collect crowdsourced data from people in urban settings. This platform, Project Cozie[1], has already been developed and tested in the indoor thermal

[1]https://github.com/cozie-app

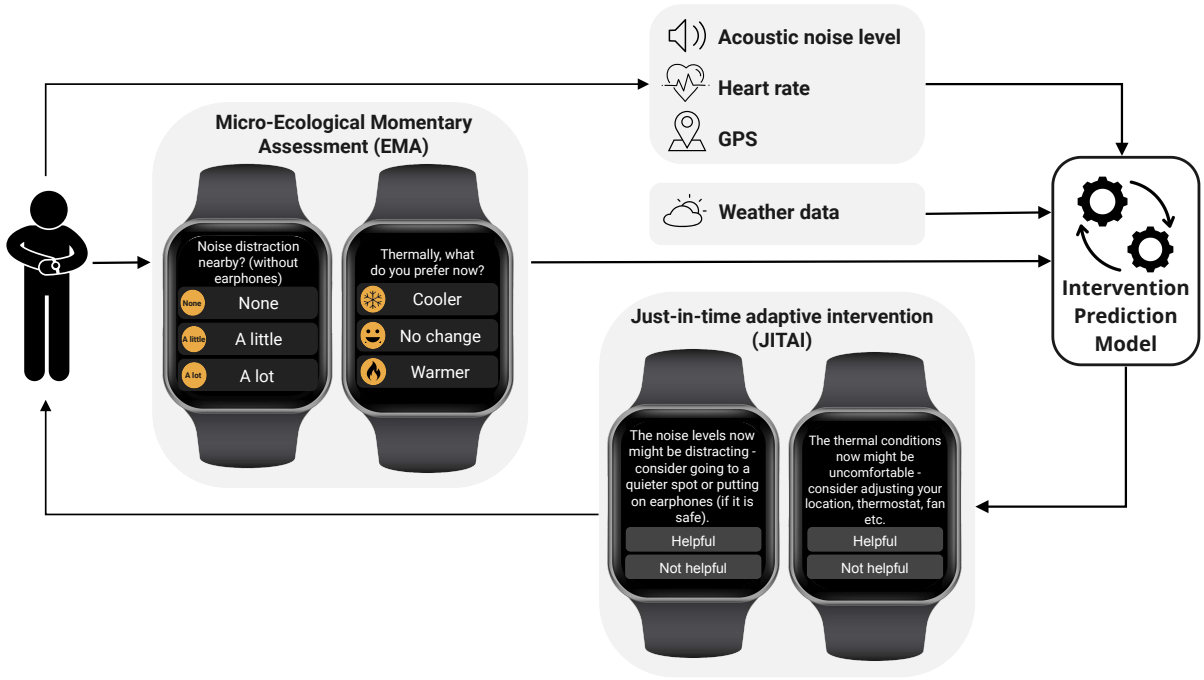


Figure 1: Schematic overview of how the proposed methodology collects information from a person using micro-ecological momentary assessment (EMA) question flows and combines this subjective information with physiological, location, and weather data to create just-in-time adaptive interventions (JITAI) to influence behavior that can supplement system control.

comfort, privacy, and infectious disease context. However, extending its use to collect information regarding users' perceptions of acoustic privacy and the use of interventions is novel. This paper outlines the platform's adaptation to use 5-9 micro-ecological momentary assessment (EMA) questions on the smartwatch about their experience when performing daily activities. The intention is that these micro-EMA questions provide the foundation for a JITAI framework that could be used to send context-aware messages to the smartwatch to provide tailored support to users at optimal times.

2 APPROACH

The goal of this paper is to outline a framework for the utilization of micro-EMA and JITAI in the context of a built environment application. Figure 1 outlines this framework. This methodology was deployed as a proof-of-concept in a city-scale context using data from a single participant. The results of this implementation are put in the geospatial context and utilized for producing interventions that impact the decision-making capabilities of the test participants.

Micro-EMA is a form of field-based spatially and temporally diverse occupant data collection. Micro-EMA via the smartwatch application in this project can accurately collect people's feedback regarding users' perceptions of acoustic privacy and thermal comfort. These feedback responses collected, together with JITAI, are used to develop personalized models that can be used to better control, optimize, and understand existing urban environments with better accuracy than aggregated population models. Figure 2 illustrates the question flow for this methodology.

The Apple Watch platform is utilized for the deployment of the methodology. This wearable platform allows for the collection of physiological data from a participant in the study. Physiological information can be collected, such as heart rate, blood oxygen, and step count, environmental data such as noise levels, and location data from GPS. The use of these data streams is established in the literature for use in research, an example being the use of the noise meter for hearing-related studies [6].

The Cozie smartwatch application has the functionality of generating up to four short buzzes as a reminder between 9:00 and 19:00 on weekdays. This feature can be activated to remind participants to answer the micro survey questions on the Apple Watch. The micro survey is expected to take less than a minute to complete. Participants are free to provide feedback whenever they want, provided that two consecutive responses are at least 1 hour apart.

In a large-scale deployment, once 50 micro surveys are collected from the participants, up to 30 adaptive intervention messages are sent to the Apple Watch between 9:00-19:00 through the rest of the other 50 feedback points. Figure 1 shows two of the possible questions of JITAI messages that could be sent to the participants. These messages focus on helping the occupants achieve a more appropriate environment through making spontaneous decisions.

2.1 Proof-of-concept deployment

The methodology was deployed in a city-scale deployment with one test user from the project team that deployed the methodology of the platform for a four-week trial. This test user filled out the EMA survey 30 times over the course of four weeks. The JITAI

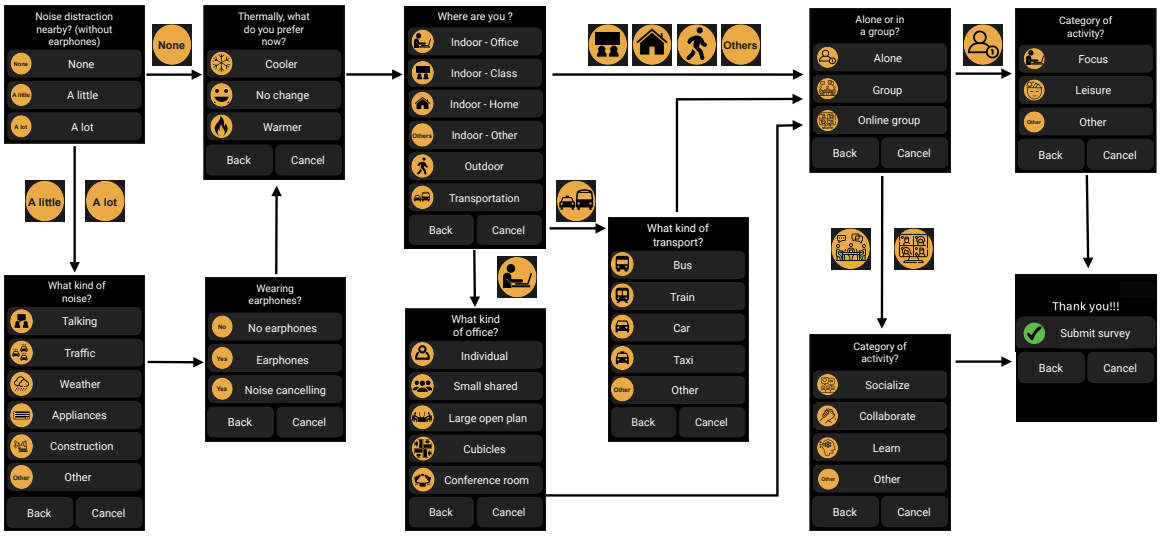


Figure 2: The micro-EMA question flow developed for this methodology enables the capture of baseline attributes of how a person experiences a diversity of spaces according to dimensions of noise distraction, thermal preferences, location, and context, such as who the person is with and what type of activity they are doing.

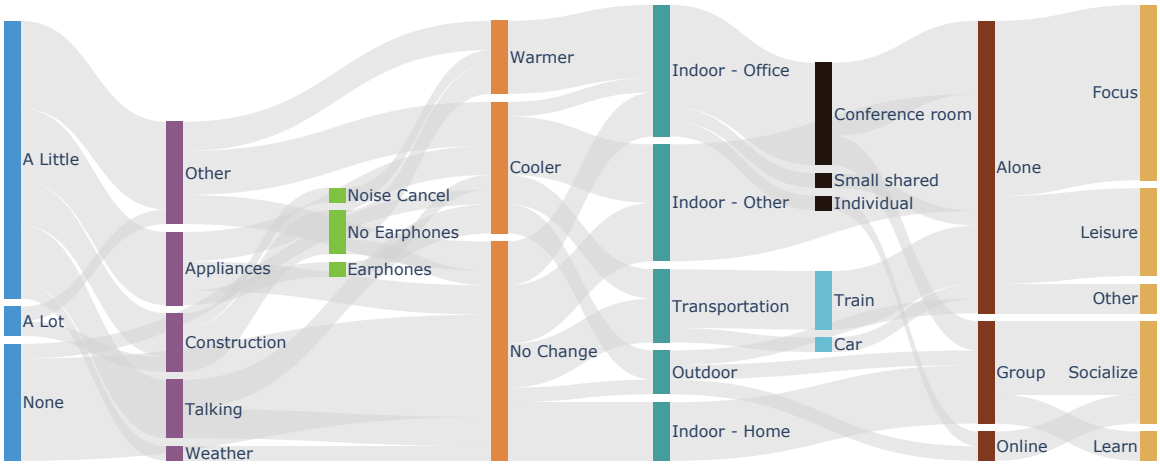


Figure 3: Overview of the EMA proof-of-concept results from 30 micro-surveys from a single test user.

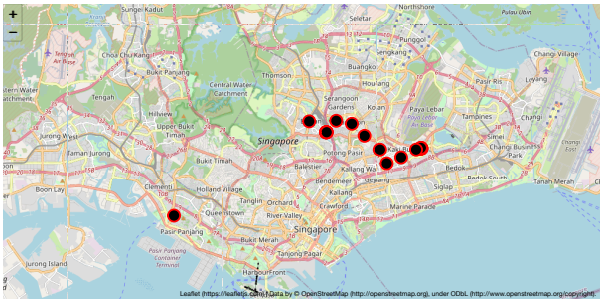


Figure 4: Example plot of the spatial urban-scale context of the data collected from this proof-of-concept deployment.

responses for this proof-of-concept deployment were also tested concurrently through their technical deployment, but the initial results of the effectiveness of those messages aren't included in the results of this work-in-progress paper.

3 RESULTS AND DISCUSSION

Figure 3 illustrates the data collection process for the proof-of-concept deployment. This Sankey diagram illustrates the breakdown of proportions of responses from the 30 micro-survey data points across the question flow found in Figure 2. From the ratio of response types for each of the questions, it can be seen that this user is usually encountering a noise/distraction issue the majority of the time. The user has a diverse breakdown of reasons for the noise distraction and sometimes uses earphones during these periods. This person is exposed to various contexts in which they feel like they would prefer warmer or cooler. And finally, the context of where the person is, who they are with, and what type of activity

they are doing can be seen from the collected data. Figure 4 shows the geotagged locations of data collected by the EMA process. The spatial diversity of this methodology and the ability to collect data in different contexts is an advantage of this type of data collection. For this proof-of-concept implementation, JITAI interventions were tested on the participant; however, no data was collected about their effectiveness. The EMA data is currently being tested in the ability to create the prediction of the right intervention message at the right place at the right time.

3.1 Future large-scale deployment

This paper is meant to be the starting point for a larger data collection process that utilizes the data from over 100 subjects, with each participant using the platform over the course of 2-4 weeks. For successful study completion, participants would be required to submit a minimum of 100 completed responses to the micro survey. This deployment would include the evaluation of the diversity of behaviors in the larger urban context and the effectiveness of JITAI in influencing behavior. This deployment will focus on the insights gained from the EMA deployment as well as the effectiveness and perceived helpfulness of JITAI to modify human behavior with respect to noise and thermal comfort.

The methodology outlined in this paper has many possible benefits; however, it raises several key questions. This method has a possible impact on the privacy concerns of occupants [11], challenges in incentivizing them to give their data, and could potentially have an influence on occupant expectations [18]. There could be non-measurable factors [4] and possible optimization issues with the way the EMA and JITAI messages are constructed [17]. These challenges will be explored in this deployment and should be considered by other future studies.

The various applications that this type of methodology unlocks in the context of the future-of-work focus on the ability of systems and humans in the built environment context to work better together. For example, when creating a recommendation system for spatial allocation in a flex-based working scenario, the development of detailed personalized models for each occupant improves the chance of satisfaction with an allocation. In addition, if the space has systems like mixed mode ventilation, then giving occupants information that encourages them to open the windows when it's nice outside can work in tandem with the building management system.

4 CONCLUSION

This paper outlines preliminary work focused on deploying micro-EMA combined with JITAI interventions to prompt building occupants to use their decision-making power to proactively improve their comfort and productivity. Behavior intervention is a key component of the future-of-work paradigm due to the increased flexibility and decision-making possibilities that occupants have in the built environment. The framework of this methodology is outlined, the results of a proof-of-concept deployment of a single test participant are shown, and the context for a future large-scale deployment is outlined.

ACKNOWLEDGMENTS

This research was funded by the following Singapore MOE Tier 1 Grants: *The Internet-of-Buildings (IoB) Platform* (A-0008305-01-00), *Ecological Momentary Assessment (EMA) for Built Environment Research* (A-0008301-01-00), and *Multi-scale Digital Twins for the Urban Environment: From Heartbeats to Cities* (A-8000139-01-00).

REFERENCES

[1] Rianne Appel-Meulenbroek, Sven Steps, Remy Wenmaekers, and Theo Arentze. 2020. Coping strategies and perceived productivity in open-plan offices with noise problems. *Journal of Managerial Psychology* 36, 4 (Jan. 2020), 400–414.
[2] Larissa Arakawa Martins, Veronica Soebarto, and Terence Williamson. 2022. A systematic review of personal thermal comfort models. *Build. Environ.* 207 (Jan. 2022), 108502.
[3] Ashrant Aryal and Burcin Becerik-Gerber. 2020. Thermal comfort modeling when personalized comfort systems are in use: Comparison of sensing and learning methods. *Build. Environ.* 185 (Nov. 2020), 107316.
[4] Veronica Lucia Castaldo, Ilaria Pigliautile, Federica Rosso, Franco Cotana, Francesco De Giorgio, and Anna Laura Pisello. 2018. How subjective and non-physical parameters affect occupants' environmental comfort perception. *Energy Build.* 178 (Nov. 2018), 107–129.
[5] Toby Cheung, Stefano Schiavon, Thomas Parkinson, Peixian Li, and Gail Brager. 2019. Analysis of the accuracy on PMV – PPD model using the ASHRAE Global Thermal Comfort Database II. *Build. Environ.* 153 (April 2019), 205–217.
[6] Tim Fischer, Stephan Schraivogel, Marco Caversaccio, and Wilhelm Wimmer. 2022. Are Smartwatches a Suitable Tool to Monitor Noise Exposure for Public Health Awareness and Otoprotection? *Front. Neurol.* 13 (March 2022), 856219.
[7] M Frontczak, S Schiavon, J Goins, E Arens, H Zhang, and P Wargocki. 2012. Quantitative relationships between occupant satisfaction and satisfaction aspects of indoor environmental quality and building design. *Indoor Air* 22, 2 (April 2012), 119–131.
[8] Helena Jahncke, Patrik Björkeholm, John E Marsh, Johan Odelius, and Patrik Sörqvist. 2016. Office noise: Can headphones and masking sound attenuate distraction by background speech? *Work* 55, 3 (Nov. 2016), 505–513.
[9] Prageeth Jayathissa, Matias Quintana, Mahmoud Abdelrahman, and Clayton Miller. 2020. Humans-as-a-Sensor for Buildings—Intensive Longitudinal Indoor Comfort Models. *Buildings* 10, 10 (Oct. 2020), 174.
[10] Jungsoo Kim and Richard de Dear. 2013. Workspace satisfaction: The privacy-communication trade-off in open-plan offices. *J. Environ. Psychol.* 36 (Dec. 2013), 18–26.
[11] Faith McCreary, Alexandra Zafiroglu, and Heather Patterson. 2016. The Contextual Complexity of Privacy in Smart Homes and Smart Buildings. In *HCI in Business, Government, and Organizations: Information Systems*. Springer International Publishing, 67–78.
[12] Clayton Miller, Mahmoud Abdelrahman, Adrian Chong, Filip Biljecki, Matias Quintana, Mario Frei, Michael Chew, and Daniel Wong. 2021. The Internet-of-Buildings (IoB) — Digital twin convergence of wearable and IoT data with GIS/BIM. *J. Phys. Conf. Ser.* 2042, 1 (Nov. 2021), 012041.
[13] Andre Matthias Müller, Ann Blandford, and Lucy Yardley. 2017. The conceptualization of a Just-In-Time Adaptive Intervention (JITAI) for the reduction of sedentary behavior in older adults. *Mhealth* 3 (Sept. 2017), 37.
[14] T Parkinson, R de Dear, and G Brager. 2020. Nudging the adaptive thermal comfort model. *Energy Build.* 206 (2020).
[15] Thomas Parkinson, Stefano Schiavon, Richard de Dear, and Gail Brager. 2021. Overcooling of offices reveals gender inequity in thermal comfort. *Sci. Rep.* 11, 1 (Dec. 2021), 23684.
[16] Matthew Saponaro, Ajith Vemuri, Greg Dominick, and Keith Decker. 2021. Contextualization and individualization for just-in-time adaptive interventions to reduce sedentary behavior. In *Proceedings of the Conference on Health, Inference, and Learning (CHIL '21)*. Association for Computing Machinery, New York, NY, USA, 246–256.
[17] Lucile Sarran, Christian A Hviid, and Carsten Rode. 2020. Correlation between perceived usability of building services and indoor environmental satisfaction in retrofitted low-energy homes. *Build. Environ.* 179 (July 2020), 106946.
[18] Marcel Schweiker, Romina Rissetto, and Andreas Wagner. 2020. Thermal expectation: Influencing factors and its effect on thermal perception. *Energy Build.* 210 (March 2020), 109729.
[19] Zhe Wang, Richard de Dear, Maohui Luo, Borong Lin, Yingdong He, Ali Ghahramani, and Yingxin Zhu. 2018. Individual difference in thermal comfort: A literature review. *Build. Environ.* 138 (June 2018), 181–193.
[20] P Zheng, C Wang, Y Liu, B Lin, H Wu, Y Huang, and X Zhou. 2022. Thermal adaptive behavior and thermal comfort for occupants in multi-person offices with air-conditioning systems. *Build. Environ.* 207 (2022).

B2RL: An open-source Dataset for Building Batch Reinforcement Learning

Hsin-Yu Liu*
hyl001@eng.ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Xiaohan Fu
x5fu@eng.ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Bharathan Balaji**
bhabalaj@amazon.com
Amazon
USA

Rajesh Gupta
gupta@eng.ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Dezhi Hong**
hondezhi@amazon.com
Amazon
USA

ABSTRACT

Batch reinforcement learning (BRL) is an emerging research area in the RL community. It learns exclusively from static datasets (i.e. replay buffers) without interaction with the environment. In the offline settings, existing replay experiences are used as prior knowledge for BRL models to find the optimal policy. Thus, generating replay buffers is crucial for BRL model benchmark. In our B2RL (Building Batch RL) dataset, we collected real-world data from our building management systems, as well as buffers generated by several behavioral policies in simulation environments. We believe it could help building experts on BRL research. To the best of our knowledge, we are the first to open-source building datasets for the purpose of BRL learning.

ACM Reference Format:
Hsin-Yu Liu*, Xiaohan Fu, Bharathan Balaji**, Rajesh Gupta, and Dezhi Hong**. 2022. B2RL: An open-source Dataset for Building Batch Reinforcement Learning. In *Third ACM SIGEnergy Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM) (RLEM '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3566164

1 INTRODUCTION

Reinforcement learning (RL) is widely studied in the building research area. Most studies focus on RL learning in an online paradigm [5, 11, 19, 25, 28, 30], assuming there is a simulation environment for RL models to interact with during training and evaluation stages before real-world deployment. Simulators such as EnergyPlus [3] and TRNSYS [14] are used to simulate the thermal states of a building. However, designing and calibrating such models for a large building is time-consuming and requires expertise.

In real-world scenarios, most large buildings are controlled via building management systems (BMS), where thermal data can be stored in database. With advances in sensing technologies and machine learning, data-driven models have been more popular in recent research. Batch reinforcement learning, a data-driven approach that learns only from fixed dataset generated with unknown behavioral policy, has not been explored widely in the building control community. BRL models are capable of learning the optimal policy without accurate environment models or simulation environments as oracles. In our study, we open-source both our dataset (https://github.com/HYDesmondLiu/B2RL) extracted from real building and the one generated with Sinergym [12], a building RL simulation environment which integrates EnegryPlus and BCVTB [26] with OpenAI Gym [2] interface. Furthermore, we experiment with several state-of-the-art BRL methods. The experimental results could be re-used as benchmarks for algorithm comparison.

2 RELATED WORK

2.1 Building batch reinforcement learning

Previously, several studies implement fitted Q-iteration (FQI) and batch Q-learning [20, 21, 23, 27]. However, for FQI and batch Q-learning, they are based on pure off-policy algorithms. Fujimoto et al. [23] show that off-policy methods exacerbate the extrapolation error in a pure offline setting. These errors are attributed to Q-network training on historical data but exploratory actions yield policies which are different from the behavioral ones.

Recently, several studies related to building deep BRL research have emerged. Zhang et al. [29] apply CQL [16] on the CityLearn [24] testbed as simulator. Liu et al. [17] incorporates a Kullback-Leibler term in Q-update to penalize policies that are far from the previous one to improve from state-of-the-art BRL algorithm and deploy in real environments without setting up simulators.

2.2 Batch reinforcement learning datasets

To our best knowledge, the only open-source BRL dataset is the D4RL dataset [8]. They have generated various robotic control datasets. In our study, we open-source two building datasets, one contains real building buffers extracted from our building database with sensor readings, setpoints control history, and the estimated energy consumption calculated by Zonepac [1]. Then, we process them as Markov Decision Process (MDP) tuples. The other one is a

*Corresponding author.
** Work unrelated to Amazon.

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3566164

set of buffers that contain different qualities of transitions generated by pre-trained behavioral agents with simulation environments.

3 APPROACH AND RESULTS

3.1 Real building buffers

3.1.1 Data acquisition. The real building buffer is extracted from the readings of student labs in one of the school buildings. The amount of datapoints in the buffers ranges from 170~260K, depending on the number of rooms involved and missing values. We obtain data of an entire year, from the beginning of July 2017 to the end of June 2018 for 15 rooms across 3 floors. The RL setup in our experiments is listed as below:

- *State*: Indoor air temperature, actual supply airflow, outside air temperature, and humidity.
- *Action*: Zone air temperature setpoint and actual supply airflow setpoint. Both are in continuous space and the action spaces are normalized in the range of $[-1, 1]$ as a standard RL settings.
- *Reward*: Our reward function is a linear combination of thermal comfort and energy consumption. The reward function at time step t is:

$$R_t = -\alpha|TC_t| - \beta P_t, \tag{1}$$

where α, β are the weights balancing different objectives and could be tuned to meet specific goals, TC_t is the thermal comfort index at time t, P_t is the HVAC power consumption at time t. We compute P_t attributed to a thermal zone using heat transfer equations [1].

3.1.2 BRL benchmarks.

- Batch-constrained deep Q-learning (BCQ) [10]: BCQ is a model-free RL method that mitigates extrapolation errors induced by incorrect value estimation of out-of-distribution actions selected out of existing dataset.
- Bootstrapping Error Accumulation Reduction (BEAR) [15]: BEAR identifies bootstrapping error as a key source of BRL instability. The algorithm mitigates out-of-distribution action selection by searching over the set of policies that is akin to the behavioral policy.
- Pessimistic Q-Learning (PQL) [18]: PQL uses pessimistic value estimates in the low-data regions in the Bellman optimality equation as well as the evaluation back-up. It can yield stronger guarantees when the concentrability assumption does not hold. PQL learns from policies that satisfy a bounded density ratio assumption similar to on-policy policy gradient methods.

3.1.3 Experiment details. Each algorithm is run in one room on each floor for an entire week so that outside air temperature (OAT) is the same. For instance, in one week we run algorithm A in rooms in the same stack on different floors, e.g. 2144, 3144, and 4144, and at the same time algorithm B runs on 2146, 3146, and 4146, and so forth. In each room, we train the algorithm for 1,000 time steps, which is about one week. We evaluate each algorithm in three different rooms (one room from each floor: 2F, 3F, and 4F). These rooms are of roughly the same size and occupancy capacity. Each time step is 10 minute due to the data writing rate in our BMS. More details of the experiments are described previously in our previous study [29].

Fig. 2 shows the learning curves of each algorithm, where each solid line is the average reward of all runs for the same method; semi-transparent bands represent the range of all runs for a particular algorithm. And gray dotted vertical lines indicate 00:00AM of each day. The horizontal black dotted line is the average reward in the buffer. Fig. 3 shows the analysis of the optimization objectives in the reward function, for energy consumption, the default control method rule-based contorl (RBC) method is normalized to 1. And for thermal comfort we are showing absolute averaged values.

As we need to calculate the thermal comfort level as required by our reward function, we adopt the widely used predicted mean vote (PMV) [6] measure as our thermal comfort index. In this metric, thermal comfort satisfaction ranges from −3 (cold) to 3 (hot), where PMV within the range of −0.5 to 0.5 is considered as thermal comfortable. We adopt the ASHRAE RP-884 thermal comfort data set [4] and train a simple gradient boosting tree (GBT) model [13] to predict the thermal comfort by taking the current thermal states given by our building system in real-time.

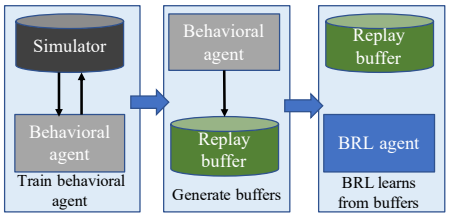


Figure 1: Flow of buffer generation and BRL training

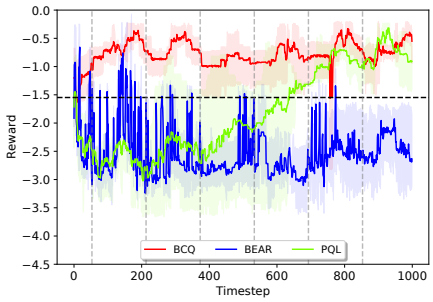


Figure 2: Episode reward comparison in real building

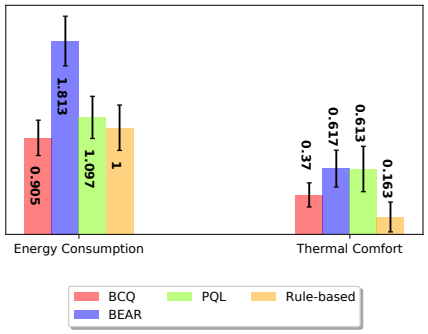


Figure 3: Optimization objectives analysis in real building

3.2 Simulated buffers

3.2.1 Data acquisition. We adopt Sinergym, an open-source simulation and control framework for training RL agents [12]. It is compatible with EnergyPlus models using Python APIs. Our approach follows the BRL paradigm. (1) We first train behavioral RL agents for $500K$ timesteps and select the one that gives the highest average score as the expert agent. Then we run on a 5-zone building, which is a single floor building divided into 5 zones, 1 interior and 4 exterior with 3 weather types: cool, hot, and mixed in continuous settings. We also experiment on two different kinds of response type, deterministic and stochastic. Then we generate expert buffer with $500K$ transitions as the expert buffer. (2) A medium buffer is generated when the behavioral agent is trained "halfway", which means the evaluation score reaches half of the expert agents' final average scores. (3) We randomly initialize the agent, which samples action from allowed action spaces with uniform distribution to generate buffers. (See Fig. 1)

- State: Site outdoor air dry bulb temperature, site outdoor air relative humidity, site wind speed, site wind direction, site diffuse solar radiation rate per area, site direct solar radiation rate per area, zone thermostat heating setpoint temperature, zone thermostat cooling setpoint temperature, zone air temperature, zone thermal comfort mean radiant temperature, zone air relative humidity, zone thermal comfort clothing value, zone thermal comfort Fanger model PPD, zone people occupant count, people air temperature, facility total HVAC electricity demand rate, current day, current month, and current hour.
- Action: Heating setpoint and cooling setpoint in continuous settings.
- Reward: We follow the default linear reward settings, it considers the energy consumption and the absolute difference to temperature comfort.

3.2.2 BRL benchmarks. With various qualities of buffers, we compare several most representative benchmarks in the BRL literature and summarize the average scores and standard deviation in the last 5 evaluations across 3 random seed runs (see Table 1). The scores of random policy is normalized to 0 and expert policy is normalized to 100.

- TD3+BC: An offline version of TD3, it simply adds a behavior cloning term to regularize actor policy towards behavioral policy [9] combined with mini-batch Q-values and buffer states normalization for stability improvement.
- CQL: Conservative Q-learning [16], derived from SAC, learns a lower-bound estimates of the value function, by regularizing the Q-values during training.
- BC: Behavior cloning, we train a VAE to reconstruct action given state. It simply imitate the behavioral agent without reward signals.

We train each algorithm for $500K$ timesteps. For every $25K$ timesteps of training we evaluate the models for one episode. As an example, we illustrate BRL learning curves with expert buffers in Fig. 4.

4 CONCLUSION AND FUTURE WORKS

We open-source our building control datasets for both real buildings and simulation environments for BRL learning. The goal is to encourage building domain experts to explore opportunities in building-BRL research. We provide these datasets for researchers to implement fast prototyping without generating buffers on their own. Recently, many building-RL libraries are published [7, 22, 24] for the purpose of building RL training without the need to set up thermal simulators beforehand. Our future work is to generate more diverse buffers with various building environments and different weather types for BRL benchmarks.

ACKNOWLEDGEMENT

This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

REFERENCES

[1] Bharathan Balaji, Hidetoshi Teraoka, Rajesh Gupta, and Yuvraj Agarwal. 2013. Zonepac: Zonal power estimation and control via hvac metering and occupant feedback. In *BuildSys*. 1–8.
[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
[3] Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. 2001. EnergyPlus: creating a new-generation building energy simulation program. *Energy and buildings* 33, 4 (2001), 319–331.
[4] Richard J De Dear. 1998. A global database of thermal comfort field experiments. *ASHRAE transactions* 104 (1998), 1141.
[5] Xianzhong Ding, Wan Du, and Alberto E Cerpa. 2020. MB2C: Model-Based Deep Reinforcement Learning for Multi-zone Building Control. In *BuildSys*. 50–59.
[6] Povl O Fanger et al. 1970. Thermal comfort. Analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering*. (1970).
[7] Arduin Findeis, Fiodar Kazhamiaka, Scott Jeen, and Srinivasan Keshav. 2022. Beobench: a toolkit for unified access to building simulations for reinforcement learning. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*. 374–382.
[8] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219* (2020).
[9] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.
[10] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *ICML*. PMLR, 2052–2062.
[11] Guanyu Gao, Jie Li, and Yonggang Wen. 2020. DeepComfort: Energy-Efficient Thermal Comfort Control in Buildings via Reinforcement Learning. *IEEE Internet of Things Journal* 7, 9 (2020), 8472–8484.
[12] Javier Jiménez-Raboso, Alejandro Campoy-Nieves, Antonio Manjavacas-Lucas, Juan Gómez-Romero, and Miguel Molina-Solana. 2021. Sinergym: a building simulation and control framework for training reinforcement learning agents. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 319–323.
[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *NIPS* 30 (2017), 3146–3154.
[14] SA Klein. 1976. University of Wisconsin-Madison Solar Energy Laboratory. *TRNSYS: A transient simulation program. Eng. Experiment Station* (1976).
[15] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949* (2019).
[16] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.
[17] Hsin-Yu Liu, Bharathan Balaji, Sicun Gao, Rajesh Gupta, and Dezhi Hong. 2022. Safe HVAC Control via Batch Reinforcement Learning. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 181–192.

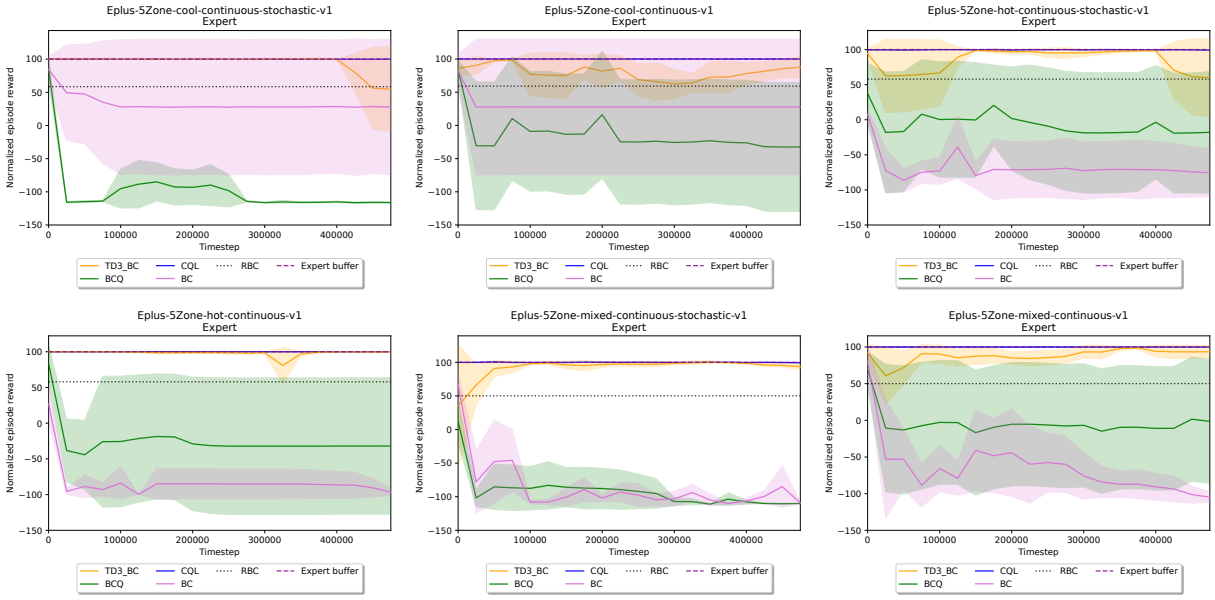


Figure 4: Learning curves of BRL models that learn from expert buffers. Solid line shows the averaged value across three random seeds per algorithm, and the half-transparent region indicates the range with one standard deviation.

Environment	Buffer	TD3+BC	CQL	BCQ	BC
hot-deterministic	Expert	**99.72±0.1**	**100.00±0.00**	-32.02±0.07	-89.2±3.95
hot-deterministic	Medium	-49.59±8.19	**67.65±17.06**	13.41±16.59	-12.55±7.27
hot-deterministic	Random	-45.73±15.13	-23.19±4.52	**69.21±18.52**	-26.74±15.91
mixed-deterministic	Expert	**94.67±2.04**	**100.00±0.00**	-6.22±5.24	-95.46±6.6
mixed-deterministic	Medium	36.23±4.31	37.36±19.31	**64.46±0.65**	-103.4±2.12
mixed-deterministic	Random	-13.72±22.25	-23.46±20.33	-65.30±20.40	-27.82±11.79
cool-deterministic	Expert	81.11±5.24	**100.00±0.00**	-29.75±3.18	27.76±0
cool-deterministic	Medium	-49.97±0.00	55.44±6.46	**70.19±17.06**	10.48±22.11
cool-deterministic	Random	-58.40±3.21	12.99±2.28	**27.77±31.39**	8.62±41.97
hot-stochastic	Expert	77.69±17.18	**99.49±0.20**	-15.35±5.92	-72.86±1.73
hot-stochastic	Medium	-14.85±0.00	**39.93±2.64**	-62.21±19.31	-10.45±12.85
hot-stochastic	Random	-1.82±2.68	**36.65±11.95**	-1.24±14.80	31.22±13.51
mixed-stochastic	Expert	**96.61±2.13**	**99.77±0.26**	-108.38±2.58	-102.02±9.32
mixed-stochastic	Medium	9.49±0.00	**80.13±8.19**	70.75±6.46	-107.41±3.41
mixed-stochastic	Random	28.02±8.69	**94.05±2.08**	-109.47±0.17	38.66±24.64
cool-stochastic	Expert	78.27±20.01	**99.97±0.12**	-115.86±0.41	28.15±0.35
cool-stochastic	Medium	16.09±0.00	**81.57±4.31**	-11.55±2.64	-50.37±2.45
cool-stochastic	Random	-44.33±16.01	-97.35±2.09	-53.92±10.07	**25.44±13.42**
Sum		339.50±127.23	**960.99±101.81**	-295.49±175.47	-527.93±193.48

Table 1: Average normalized score over the final 5 evaluations and 3 random seeds. ± corresponds to standard deviation over the last 5 evaluations across runs.

[18] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. 2020. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202* (2020).

[19] Naren Srivaths Raman, Adithya M Devraj, Prabir Barooah, and Sean P Meyn. 2020. Reinforcement learning for control of building HVAC systems. In *2020 American Control Conference (ACC)*. IEEE, 2326–2332.

[20] Frederik Ruelens, Bert J Claessens, Stijn Vandael, Bart De Schutter, Robert Babuška, and Ronnie Belmans. 2016. Residential demand response of thermostatically controlled loads using batch reinforcement learning. *IEEE Transactions on Smart Grid* 8, 5 (2016), 2149–2159.

[21] Frederik Ruelens, Bert J Claessens, Stijn Vandael, Sandro Iacovella, Pieter Vingerhoets, and Ronnie Belmans. 2014. Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning. In *2014 Power Systems Computation Conference*. IEEE, 1–7.

[22] Paul Scharnhorst, Baptiste Schubnel, Carlos Fernández Bandera, Jaume Salom, Paolo Taddeo, Max Boegli, Tomasz Gorecki, Yves Stauffer, Antonis Peppas, and Chrysa Politi. 2021. Energym: A building model library for controller benchmarking. *Applied Sciences* 11, 8 (2021), 3518.

[23] José Vázquez-Canteli, Jérôme Kämpf, and Zoltán Nagy. 2017. Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration. *Energy Procedia* 122 (2017), 415–420.

[24] José R Vázquez-Canteli, Sourav Dey, Gregor Henze, and Zoltán Nagy. 2020. CityLearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management. *arXiv preprint arXiv:2012.10504* (2020).

[25] Zhe Wang and Tianzhen Hong. 2020. Reinforcement learning for building controls: The opportunities and challenges. *Applied Energy* 269 (2020), 115036.

[26] Michael Wetter, Philip Haves, and Brian Coffey. 2008. *Building controls virtual test bed.* Technical Report. Lawrence Berkeley National Laboratory.

[27] Lei Yang, Zoltan Nagy, Philippe Goffin, and Arno Schlueter. 2015. Reinforcement learning for optimal control of low exergy buildings. *Applied Energy* 156 (2015), 577–586.

[28] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2019. Building HVAC scheduling using reinforcement learning via neural network based model approximation. In *BuildSys*. 287–296.

[29] Chi Zhang, Sanmukh Rao Kuppannagari, and Viktor K Prasanna. 2022. Safe Building HVAC Control via Batch Reinforcement Learning. *IEEE Transactions on Sustainable Computing* (2022).

[30] Zhiang Zhang and Khee Poh Lam. 2018. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *BuildSys*.

ComfortLearn: Enabling agent-based occupant-centric building controls

Matias Quintana
maqr@nus.edu.sg
National University of Singapore

Zoltan Nagy
nagy@utexas.edu
The University of Texas at Austin

Federico Tartarini
federico.tartarini@bears-berkeley.sg
Berkeley Education Alliance for Research in Singapore

Stefano Schiavon
schiavon@berkeley.edu
University of California, Berkeley

Clayton Miller
clayton@nus.edu.sg
National University of Singapore

ABSTRACT

The intersection of buildings control and thermal comfort modeling may seem obvious, but there are still prevalent challenges in combining them. "Occupant centric" control strategies are mainly trained using building data but rarely leverage occupants' feedback. While thermal comfort models are developed using occupants' data but are seldom integrated into building controls. To bridge this gap, we developed an open-source simulation tool named ComfortLearn. ComfortLearn is an OpenAI Gym-based environment that leverages historical building management system data from real buildings and existing longitudinal thermal comfort datasets for occupant-centric control strategies and benchmarking. We used an evaluation metric named 'exceedance' to evaluate occupants' thermal comfort and provide a more realistic picture than traditional evaluations like comfort bands. This setup allows the analysis of different building control strategies and their effect on real occupants, based on empirical data, without the need for computationally expensive co-simulations. A theoretical case study implementation shows that an as-is schedule-based controller complies with its comfort band more than 93% of the time, but the simulated occupants are comfortable for only 25% of the occupied time.

CCS CONCEPTS

• **Software and its engineering**; • **Human-centered computing**; • **Computing methodologies** → **Modeling and simulation**;

KEYWORDS

Thermal comfort, Building Control, Agent-based, Smart Buildings

ACM Reference Format:
Matias Quintana, Zoltan Nagy, Federico Tartarini, Stefano Schiavon, and Clayton Miller. 2022. ComfortLearn: Enabling agent-based occupant-centric building controls. In *Third ACM SIGEnergy Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM) (RLEM '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3566167

RLEM '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3566167

1 INTRODUCTION

Multidisciplinary collaboration between the building control and thermal comfort communities is of paramount importance. From the building controls perspective, the motivation that advances control strategies for operating Heating Ventilation, and Air-Conditioning (HVAC) systems is to reduce energy consumption of buildings [5, 6]. This has sparked research toward data-driven methods like Reinforcement Learning (RL). Although these "occupant-centric" methods rely on building data to train their controllers, limited occupant data are used besides whether the room is empty or not [6] or probabilistic occupancy trends [5].

Among the thermal comfort research community, the advent of data-driven models has pushed researchers to investigate the benefit of using personal, environmental, and physiological variables to predict thermal comfort. Models are trained and tested using the collected data, however, seldom these are used for building control purposes. Existing attempts to bridge the gap between these two fields are real-world deployments of Occupant-Centric Control (OCC) strategies or work using simulation environments based on the OpenAI Gym environment [3]. OpenAI Gym is an open-source Python library for developing and comparing control algorithms.

In this work, we introduce ComfortLearn, an OpenAI Gym environment that leverages empirical data from both buildings and occupants. ComfortLearn standardizes the way occupants are modeled and leverages the available datasets captured by both the building controls and the thermal comfort community. ComfortLearn aims, for the first time, to link real occupant data with building controls in a standardized manner. We present the tool's flexibility and usage through a case study.

2 APPROACH

2.1 ComfortLearn Overview

ComfortLearn uses empirical data to model occupants and buildings. This allows investigating the impact of occupants' interactions on energy savings and thermal comfort. Figure 1 shows a visual overview of the framework and its three steps. This paper covers only the data collection and pre-processing phase of the ComfortLearn platform. The simulation step, which involves real-building control strategies benchmarking will be evaluated in future

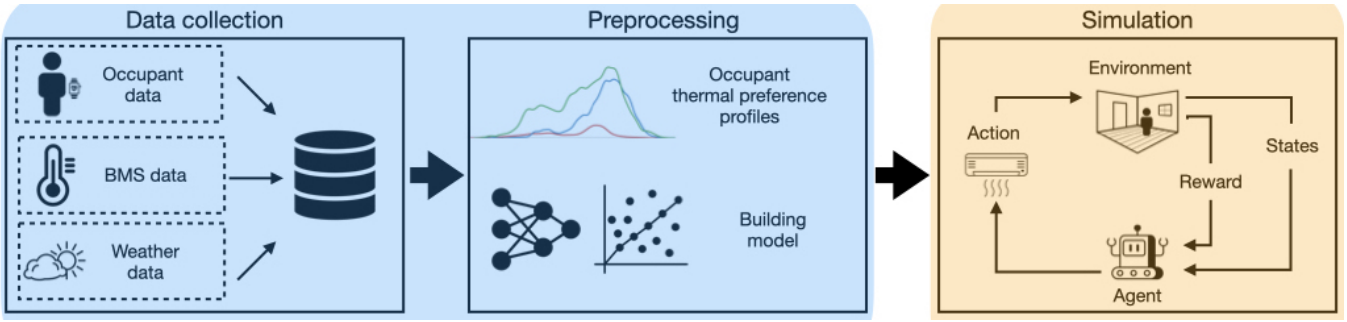


Figure 1: ComfortLearn overview. ComfortLearn comprises three overarching stages: Data collection, preprocessing, and simulation. The first two stages (blue-shaded) are presented in this work, while the third (orange-shaded) is will be covered in a future publication.

work. The platform, which is used as an OpenAI Gym environment, is available on a public GitHub repository: `https://github.com/buds-lab/ComfortLearn`.

2.2 Occupant modelling

Previous researchers mainly only modeled human-building interaction based on their occupancy estimation alone. This approach ignores occupant thermal preferences and relies on assessing comfort using either temperature deltas [5] or ranges [8]. These strategies assume occupants' thermal comfort to be fixed. Inspired by [7], thermal preference profiles of occupants are generated based on empirical density distributions of occupants' 'right-here-right-now' thermal preference votes across various environmental conditions at which they were collected. Figure 2 shows, as an example, a combined empirical density distribution of a dataset comprising approximately 500 data points from 20 people collected over a 6-month period in Singapore [9]. Three distributions, one for each thermal preference type of vote are shown: "Cooler", "No change", and "Warmer". Additional personal and environmental variables beyond the indoor temperature the occupant was exposed to were available in the original dataset. However, given its predominant effect on thermal comfort and ease of measurement in most buildings by the Building Management System (BMS), this feature alone is used to generate the empirical probability density distributions.

Unlike previous work that generates comfort probabilities for each occupant [7], ComfortLearn is not limited by the number of occupants in the dataset nor by one specific dataset. It can generate as many occupants as required by combining all the available thermal preference votes into one empirical density distribution (Figure 2). To generate different occupant profiles, we introduced a parameter that we called *tolerance*. This parameter ranges from 0 to 1 and it is used to filter out participants with a given percentage of "No change" votes. For example, a $tolerance = 0.6$ means that the empirical density distributions are based only on occupants whose "No change" thermal preference votes consist of 60% or less of their overall data (Figure 2b). This parameter does not try to generalise the thermal preference of occupants since their votes are also dependent on the environmental conditions they were exposed to, as well as their clothing and activity levels; it provides a practical way of generating different profiles for simulated occupants. In the context of ComfortLearn, a higher tolerance value means that the occupant has more votes of "No Change" and, therefore, it may be assumed that this participant may be easier to please. Occupants are then assigned an occupancy profile. This can be done using a fixed schedule, e.g., from 8 am to 5 pm or it can be randomly generated.

Finally, to determine the thermal preference of an occupant, the room indoor temperature is used to obtain the thermal preferences probabilities from its respective empirical density distributions (Figure 2). Then, the final vote is obtained by a weighted sample of these three probabilities. This is repeated for all occupants.

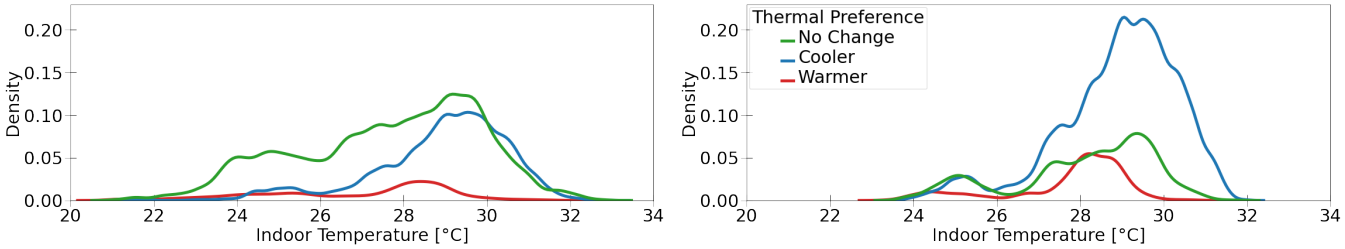
2.3 Building modelling

Related work tend to use pre-simulated buildings or perform co-simulation with either EnergyPlus [5, 6] or Modelica [8]. However, the emphasis of control-oriented models is not to strive for high-fidelity, labor-intensive, and time-consuming models but to rely on approximate models [4]. Jung et al. (2019) have shown that once occupants are modeled with comfort probabilities, the building dynamics can rely on set-point temperature data and its effects on the occupant [7]. Therefore, ComfortLearn uses data measured in real buildings to iterate over the historical values and see the effects on the occupants at time steps of 15 minutes. It is possible, however, to use the historical BMS data and develop a black or gray-box data-driven model of the building or zone(s) if desired. Data-driven models are more suitable when control actions are taken on the building's HVAC system (Step 3, orange shaded region in Figure 1). Still, this step is not considered currently in this work and is left for future studies.

2.4 Evaluation metrics

We used the Exceedance cumulative index to quantitatively assess discomfort [2]. This metric counts all the occurrences when the occupant thermal preference is different from "No change". This value is then normalized by the cumulative duration that the occupant utilizes the space, in periods of 15 minutes, resulting in a value ranging from 0 to 1 where a larger Exceedance value indicates that the occupant is more uncomfortable during the occupied time. The ideal value for occupant comfort should be close to 0. Equation 1 shows the Exceedance calculation.

$$\text{Exceedance} = \frac{\text{Time in uncomfortable conditions}}{\text{Occupied time}} \tag{1}$$



(a) All available data (Thermal preference $tolerance$ = 1.0).

(b) Thermal preference $tolerance$ = 0.6

Figure 2: Empirical thermal preference probability density of occupants for various values of $tolerance$ based on the dataset from [9]. The parameter $tolerance$ ranges from 0 to 1, and it is used to filter out participants with a ratio of "No change" thermal preference $\leq tolerance$. As the $tolerance$ value decreases, other empirical probability densities like the one for "Cooler" (blue curve) expand, meaning occupants with lower tolerance are more difficult to please and vice-versa.

3 CASE STUDY: OCCUPANT PROFILES IN DIFFERENT TEMPERATURE ZONES

We present a real-world case study of ComfortLearn, and its occupants' comfort evaluation under as-is building conditions with no control algorithm, where the building and thermal comfort datasets were collected in Singapore (tropical climate ASHRAE climate zone 1A). Two university classrooms with different ventilation mechanisms from an educational building were chosen. The building's BMS system continuously logged environmental data. We tested our tool over the whole semester period. However, since the results did not vary significantly; we decided to present only the results for a week in the middle of the semester. We analyzed a week of data from 2021-10-18 to 2021-10-22, in the middle of the fall semester, and time steps of 15 minutes are chosen.

The thermal comfort dataset used in this study is the example dataset described in Section 2.2. This dataset was chosen since is one of the largest longitudinal studies available in terms of the number of participants and labeled data points for each participant. In addition, it was collected in the same climate as where the chosen building is located. To make the data more homogeneous, only the thermal comfort responses given "indoors" and under the same activity "sitting" were considered.

For each simulation, occupants are given a random occupancy profile. At the beginning of each day, each occupant is randomly shifted to either of the available rooms. Zone 1 had a set-point temperature of 26.5 °C, while Zone 2 of 25 °C. Both zones are equipped with air-conditioning systems but Zone 1 has additionally multiple ceiling fangs. The "enter" and "exit" time for each work day is the same for all occupants. These times were randomly sampled from the following distributions $\mathcal{N}(\mu = 8, \sigma^2 = 2)$ and $\mathcal{N}(\mu = 17, \sigma^2 = 2)$, respectively, with "enter" times being forced to be after 7 am. Each occupant is assumed to stay inside the assigned room between the entry and exit times. Two different tolerance levels were chosen (0.1 and 0.6) for each of 10 occupants ($occu_1, occu_2, ..., occu_{10}$). The simulation is repeated 30 times to account for stochasticity, and the average values and standard deviations are reported.

4 RESULTS

4.1 Comfort bands

Comfort bands are typically used to evaluate advanced controllers based on Model Predicted Control (MPC), or RL [8] following ASHRAE 55 standards [1]. The building chosen is controlled using a fixed dead band around a set-point temperature during work hours, from 8 am to 5 pm on week days. Based on this, we evaluated the percentage of the time, between 8 am and 5 pm, the indoor temperature remained inside the comfort band defined in the BMS. The comfort bands are assumed to be equal to the room set-point temperature ±1°C. In Zone 1, the comfort band is between 25.5 - 27.5 °C (Band 1). This zone has a significant amount of glazing that is fully shaded. In Zone 2, the comfort band is between 24 - 26 °C (Band 2). This zone has a small amount of glazing and more thermal mass. Figure 3 shows temperatures recorded in each zone over the chosen time window and their respective comfort bands. The schedule-based controllers of both Zones kept the indoor temperature within their respective comfort bands for more than 93% of the time between 8 am and 5 pm. After 5 pm, the system turned off in both spaces.

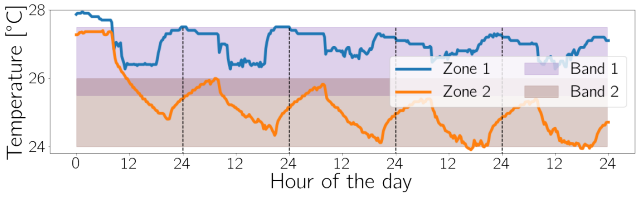


Figure 3: Available zones' indoor temperatures throughout a week at intervals of 15 min. Zone 1 and Zone 2 have a set point of 26.5°C and 24°C with a comfort band of ±1°C (Band 1 and Band 2 respectively).

4.2 Average Exceedance

Figure 4 shows the average daily Exceedance for occupants at a tolerance value of 0.6. Occupants with different tolerance values may benefit from being situated in specific environments. At a tolerance level of 0.6, there is a more evident difference between daily average Exceedance values between the Zones. For this occupant's profile, Zone 2 is a better environment in terms of thermal preference than Zone 1. This may be likely because people with a lower tolerance have a narrower range of indoor temperatures at which they feel comfortable, which fits the traditional air-conditioned environment of Zone 2. Once the tolerance level is increased to 1.0, the daily average Exceedance is consistent across all at around 0.25 for both Zones, potentially because the acceptable range of temperatures makes Zone 1 an equally appropriate place as Zone 2.

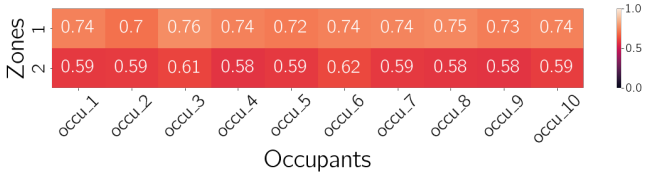


Figure 4: Daily average Exceedance values for each occupant averaged over 30 simulations for occupants with *tolerance* = 0.6. Low values of Exceedance are better.

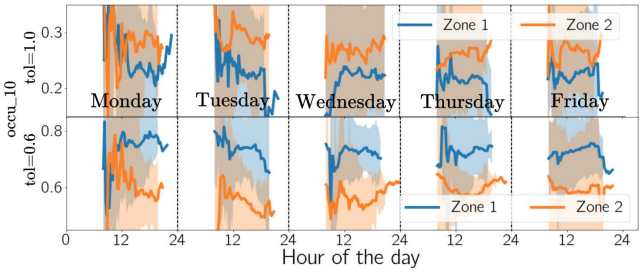


Figure 5: Time-series of Exceedance values throughout a week for a single occupant of *tolerance* 1.0 (top) and 0.6 (bottom). The darker line is the mean over time with a shaded region represented by 2 standard deviations. The lower the better.

4.3 Exceedance over time

A low Exceedance weekly value is a positive result, but discomfort would still be a significant issue if the occupant has sharp changes between low and high Exceedance values throughout the day. Thus, the mean and 2 standard deviations of the Exceedance values, for each time step are calculated for every occupant. Figure 5 shows the same occupant $occu_{10}$ for both tolerances of 1.0 and 0.6. The amplitude of the shaded regions is a consequence of the stochasticity of aggregating the 30 simulation runs, but overall there is little variability. For tolerance 0.6, Exceedance values for Zone 2 (orange shaded region) are lower than the ones for Zone 1 (blue shaded region) (bottom Figure 5) On the other hand, for a tolerance of 1.0, most of the Exceedance values (shaded regions) overlap for Zone 1 and 2 (top Figure 5). These values are consistent with what was found by looking at the overall daily average Exceedance in Section 4.2 where occupants of tolerance 0.6 were better-off in Zone 2 and occupants with tolerance 0.1 have a similar Exceedance value for both Zones.

5 DISCUSSION

We presented ComfortLearn, an OpenAI gym environment for simulating different types of occupants based on empirical data for building controls and thermal comfort. We demonstrated that traditional metrics for comfort in building controls, such as comfort bands, could lead to misleading results. A schedule-based controller appears to comply with its comfort band more than 93% of the time within a work day (8 am to 5 pm) without addressing thermal comfort based on the empirical data, and this is only visible thanks to our methodology and using the Exceedance as a thermal comfort metric. Based on the Exceedance, occupants were comfortable around 75% of their occupied time, much lower if their thermal preference tolerance diminishes. ComfortLearn adds to the need for standardized and customizable environments for building controls and thermal comfort [12]. Unlike other gym environments, this tool and methodology do not rely on co-simulation with other computationally heavy building simulation tools, allowing users to conduct more analyses and experiments. Similar to existing efforts using OpenAI gym framework like CityLearn [10, 11], we believe ComfortLearn can grow and expand within both thermal comfort, and the building controls communities. This tool opens the way for more research regarding intervention and studies centered on the occupant itself, thanks to empirical thermal comfort data, a feature missing from current Occupant-Centric Control (OCC) approaches. Future studies can benefit from the plethora of options and combinations of settings the simulated occupants can take as well as the incorporation of different building control strategies and their effect on them. We envision researchers using the empirical thermal preference labels as *states* and Exceedance calculations as part of the reward definition for RL approaches.

ACKNOWLEDGMENTS

This research was funded by the Republic of Singapore's National Research Foundation through the SinBerBEST2 program.

REFERENCES

[1] American Society of Heating Refrigerating and Air-Conditioning Engineers. Standards Committee. 2013. *Thermal environmental conditions for human occupancy.* Vol. 2013. arXiv: 1011.1669v3 ISBN: 9788578110796.

[2] Sam Borgeson and Gail Brager. 2011. Comfort standards and variations in exceedance for mixed-mode buildings. *Building Research & Information* 39, 2 (April 2011), 118–133. https://doi.org/10.1080/09613218.2011.556345

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. (2016), 1–4. https://doi.org/10.1021/am3026129 arXiv: 1606.01540 ISBN: 2200000006.

[4] José A. Candanedo, Charalampos Vallianos, Benoit Delcroix, Jennifer Date, Ali Saberi Derakhtenjani, Navid Morovat, Camille John, and Andreas K. Athienitis. 2022. Control-oriented archetypes: a pathway for the systematic application of advanced controls in buildings. *Journal of Building Performance Simulation* (2022). https://doi.org/10.1080/19401493.2022.2063947 Publisher: Taylor & Francis.

[5] Bingqing Chen, Zichen g Cai, and Mario Bergés. 2019. Gnu-RL : A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy. In *BuildSys '19 Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ACM (Ed.). New York, NY, USA, 316–325.

[6] Xianzhong Ding, Wan Du, and Alberto Cerpa. 2019. OCTOPUS: Deep reinforcement learning for holistic smart building control. In *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.* 326–335. https://doi.org/10.1145/3360322.3360857

[7] Wooyoung Jung and Farrokh Jazizadeh. 2019. Comparative assessment of HVAC control strategies using personal thermal comfort and sensitivity models. *Building and Environment* 158, April (2019), 104–119. https://doi.org/10.1016/j.buildenv.2019.04.043 Publisher: Elsevier.

[8] Thomas Schreiber, Sören Eschweiler, Marc Baranski, and Dirk Müller. 2020. Application of two promising Reinforcement Learning algorithms for load shifting in a cooling supply system. *Energy and Buildings* 229 (2020), 110490. https://doi.org/10.1016/j.enbuild.2020.110490 Publisher: Elsevier B.V.

[9] Federico Tartarini, Stefano Schiavon, Matias Quintana, and Clayton Miller. 2022. Personalized thermal comfort models using wearables and IoT devices. *Submitted for Publication* (2022).

[10] Jose R Vazquez-Canteli, Sourav Dey, Gregor Henze, and Zoltan Nagy. 2020. CityLearn: Standardizing Research in Multi-Agent Reinforcement Learning for Demand Response and Urban Energy Management. figure 1 (2020). http://arxiv.org/abs/2012.10504 arXiv: 2012.10504.

[11] José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. 2019. CityLearn v1.0: An OpenAI gym environment for demand response with deep reinforcement learning. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 356–357. https://doi.org/10.1145/3360322.3360998 ISBN: 9781450370059.

[12] David Wölfle, Arun Vishwanath, and Hartmut Schmeck. 2020. A Guide for the Design of Benchmark Environments for Building Energy Optimization. *BuildSys 2020 - Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2020), 220–229. https://doi.org/10.1145/3408308.3427614 ISBN: 9781450380614.

Deep Reinforcement Learning with Online Data Augmentation to Improve Sample Efficiency for Intelligent HVAC Control

Kuldeep Kurte
Oak Ridge National Laboratory, USA
kurtekr@ornl.gov

Kadir Amasyali
Oak Ridge National Laboratory, USA
amasyalik@ornl.gov

Jeffrey Munk
National Renewable Energy Laboratory, USA
jeff.munk@nrel.gov

Helia Zandi
Oak Ridge National Laboratory, USA
zandih@ornl.gov

ABSTRACT

Deep Reinforcement Learning (DRL) has started showing success in real-world applications such as building energy optimization. Much of the research in this space utilized simulated environments to train RL-agent in an offline mode. Very few research have used DRL-based control in real-world systems due to two main reasons: 1) sample efficiency challenge—DRL approaches need to perform a lot of interactions with the environment to collect sufficient experiences to learn from, which is difficult in real systems, and 2) comfort or safety related constraints—user's comfort must never or at least rarely be violated. In this work, we propose a novel deep **R**einforcement **L**earning framework with online **D**ata **A**ugmentation (RLDA) to address the sample efficiency challenge of real-world RL. We used a time series Generative Adversarial Network (TimeGAN) architecture as a data generator. We further evaluated the proposed RLDA framework using a case study of an intelligent HVAC control. With a ≈28% improvement in the sample efficiency, RLDA framework lays the way towards increased adoption of DRL-based intelligent control in real-world building energy management systems.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Hardware** → **Power and energy**.

KEYWORDS

deep reinforcement learning, data augmentation, intelligent HVAC control, demand response, building energy

ACM Reference Format:
Kuldeep Kurte, Kadir Amasyali, Jeffrey Munk, and Helia Zandi. 2022. Deep Reinforcement Learning with Online Data Augmentation to Improve Sample Efficiency for Intelligent HVAC Control. In *Third ACM SIGEnergy Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM) (RLEM '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3563357.3566168

RLEM '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3566168

1 INTRODUCTION

Reinforcement Learning (RL) is a branch of Machine Learning (ML) that is concerned with learning optimal policies through reward maximization [14]. It is a trial and error method, in which RL-agent interacts with the environment through actions, collects data through observations, and improves the policy. The contemporary Deep Learning (DL) based RL algorithms such as Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), etc. have demonstrated to be effective in simulated environments where RL-agent can perform a large number of interactions with the environment before it learns the optimal policy. In real-world applications such as building energy management, RL-agent may not have an opportunity to perform a large number of interactions with the environment. This is called the sample efficiency challenges of real-world RL. Additionally, the comfort and safety-related constraints must never or rarely be violated. This means RL-agent needs to be very cautious during the exploration while performing random actions in real-world settings. Due to these reasons: 1) sample efficiency challenges, and 2) comfort and safety-related constraints, the uptake of DRL-based controls in real-world problems is still limited.

A similar trend has been observed in the research of DRL-based building energy optimization. For instance, the research in [16], [4], [10], and [6] used simulated environments of buildings and HVAC systems to train RL-agent to obtain optimal HVAC control policy. In similar research, [2], [1] used simulation environments of buildings and water heaters for training RL-agent to control water heater in a cost-efficient way. The more recent interesting approaches in this direction are RL-based energy-efficient data center [12], energy-efficient personalized thermal comfort in office buildings [18], and multi-agent multi-objective RL for controlling residential appliance scheduling [11]. These approaches used either energy plus or a custom build simulated environments for training RL-agent.

In the United States, more than 50% of buildings' energy use is attributed to HVAC [7]. Therefore, a significant energy saving can be achieved by intelligently controlling the HVAC system. To achieve this, wide-scale adoption of such intelligent systems in real-world is necessary. However, very few of the above works demonstrated RL's capability in real-world energy management applications. Most of these algorithms fall under the model-free RL category that does not assume any prior knowledge of the

environment and learn optimal policy only by interacting with the environment. Hence such approaches need a large number of interactions with the environment to obtain a good policy. This is difficult in real-world settings due to the aforementioned challenges of the real-world setting.

In this work, we proposed a DRL framework with online Data Augmentation (RLDA) capability to address the sample efficiency challenges of real-world building energy management, specifically HVAC control. DRL setup uses a replay memory that stores the agent's experiences in terms of a set of tuples comprising of current state (s_t), action (a_t), reward (r_{t+1}), and next state (s_{t+1}), i.e. $<s_t, a_t, r_{t+1}, s_{t+1}>$. A set of tuples stored consecutively in the replay memory represents a trajectory that the agent has traversed through the environment. We use a time series Generative Adversarial Network (TimeGAN, [17]) architecture to generate synthetic trajectories and use them to augment the replay memory with synthetic tuples. In the current RLDA setup, we invoke TimeGAN on day 10 which then uses all the tuples from the replay memory until day 10 and generates the synthetic tuples. Further, DRL training will use synthetic tuples along with real experiences. We compared the average cumulative energy cost of 50 executions of operating HVAC using RLDA and conventional DRL with the cumulative energy cost of a fixed setpoint baseline. The average cumulative energy cost of conventional DRL crossed baseline's cumulative energy cost on the 39th day whereas the proposed RLDA crossed the baseline's cumulative energy cost on the 28th day. This shows ≈28% improvement in the sample efficiency. This result provides experimental evidence that performing data augmentation in an online fashion during DRL training is a potential way to address the sample efficiency challenges of the real-world RL.

Contributions: The contributions of this paper are: 1) We proposed a DRL framework with a TimeGAN-based online data augmentation module to address the sample-efficiency challenges of the real-world RL. 2) We provide experimental evidence that the proposed RLDA framework improves the sample-efficiency of DRL training.

The rest of this paper is structured as follows: Section 2 describes the proposed RLDA framework. It provides the details of TimeGAN architecture and how we utilized it to generate synthetic experiences. Next, Section 3 discusses the experimental setup and results showing the sample-efficiency improvement achieved by the RLDA framework. Finally, Section 4 concludes the paper and summarizes the future work that needs to be done in this direction.

2 DEEP REINFORCEMENT LEARNING WITH ONLINE DATA AUGMENTATION FRAMEWORK

2.1 RL preliminaries

RL is a powerful paradigm for solving control optimization problems such as HVAC control of building energy management. RL-agent interacts with the environment through actions and state. Environment evolves to the next state (s_{t+1}) in the response to the action. The objective of RL algorithms is to obtain an optimal policy (π^*) that dictates what action (a_t) to take in a current state s_t. For instance, the setpoint to set for a given current indoor condition. The optimal policy maximizes the cumulative discounted reward, G_t (refer Eq. 1).

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \tag{1}$$

RL-agent keeps a record of state-action value called Q-value, $Q(s_t, a_t)$. It dictates how good it is to take an action a in a state s. Q-learning is a model-free RL algorithm that learns optimal Q-value by iteratively executing the Q-learning equation (refer Eq. 2).

$$\begin{aligned} Q_{t+1}(s_t, a_t) &\leftarrow Q_t(s_t, a_t) \\ &+ \eta(r_{t+1} + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)) \end{aligned} \tag{2}$$

2.2 Deep-Q-Network

The Q-learning algorithm works well for discrete states and actions, where maintaining a Q-table is easy. In the case where state or actions are continuous, Q-table can not be maintained. DRL algorithms such as Deep-Q-network (DQN) address this problem by using neural networks to approximate Q-table [13]. Figure 1 shows the architecture of DQN. It consists of two networks: 1) evaluation network—approximates Q-value of the current state and action, and 2) target network—approximates the Q-value of the next state and action. DQN-agent interacts with the environment and stores the experiences of tuples in a replay memory. During training, a batch of tuples is randomly drawn from the replay memory and a loss is calculated using the Q-value of the current state and action, and the target Q-value. This loss is used to update the weights of the evaluation network. The weights of the evaluation network are copied to the target network periodically (say at every ΔT training steps).

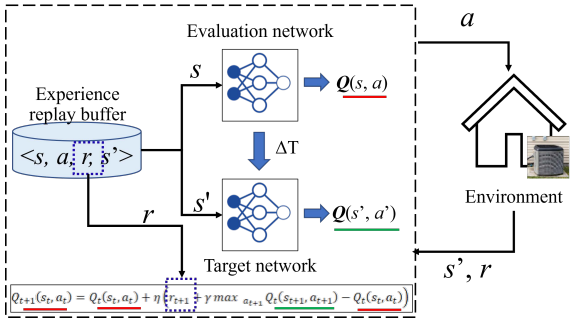


Figure 1: DQN-architecture.

2.3 TimeGAN

TimeGAN [17] is a variant of Generative Adversarial Network (GAN [8] for time-series data generation task. It was shown to demonstrate excellent performance on a time-series data augmentation task by Jinsung Yoon et al., in 2019 [17]. In building energy domain, GAN was used to generate energy use and load characteristics for multiple buildings [3], to create uncertainty-infused synthetic profiles of building performance [9], and to generating realistic building electrical load profiles [15]. Figure 2 shows TimeGAN architecture

which includes autoencoder, GAN, and supervisor architectures. Autoencoder learns the temporal dynamics in the hidden dimensions. The generated features in the hidden dimension by both encoder ($h_{1:T}$) and generator ($\tilde{h}_{1:T}$) are used in a supervised learning fashion to train the generator to capture the temporal dynamics of the real data, i.e. $p(X_t|X_{1:t-1})$. In this way, the generative model of TimeGAN learns the temporal correlations as well as relationships among features. The generator and discriminator architectures of GAN compete with each other. On one hand, the generator aims to generate synthetic time-series samples that can achieve real feature distributions. On the other hand, the discriminator aims to distinguish whether a given time-series sample is synthetic or not. Zhang et al., 2022 [19] demonstrated the use of TimeGAN for data augmentation for improving heating load prediction of the heating substation.

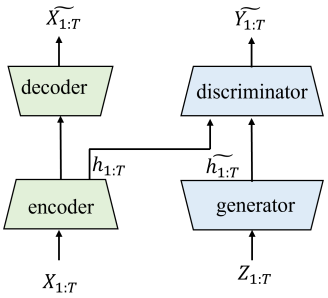


Figure 2: TimeGAN architecture.

2.4 RLDA framework

The experience replay memory stores tuples $<S_t, A_t, R_{t+1}, S_{t+1}>$ where the consecutive tuples are correlated. It represents the trajectory traversed by an RL-agent through the environment. Since these experiences are stored sequentially, they can be treated as a time-series of tuples. In a real-world setting, we want RL-agent to learn an optimal policy with a limited number of samples. In this work, we propose an online data-augmentation approach to generate synthetic experiences during training. We use TimeGAN to generate synthetic trajectories of experiences. During training, the data augmentation process is invoked after Δt_a time period, e.g. 10 days. The data augmentation will use the tuples of experiences collected until Δt_a time from replay memory. Further, a set of partial trajectories of tuples of a particular sequence length (say 16) are produced to train TimeGAN. TimeGAN uses these partial trajectories and learns to generate similar partial trajectories of tuples of the same sequence length. After this process, the synthetic tuples from these generated trajectories are used along with the real tuples during DQN training.

3 EXPERIMENTAL RESULTS

3.1 Simulation setup

We evaluated the performance of the proposed RLDA framework in a simulated environment. We trained a DQN model using weather from TMY3 Knoxville, TN, USA data for July and August months in a cooling mode. We used a grey-box model that simulates the

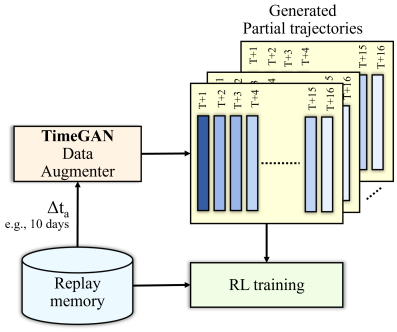


Figure 3: Proposed RLDA framework.

thermal response of a 2-story unoccupied research house located in Knoxville, TN, USA. A particle swarm optimization algorithm was used to train the simulation model parameters. More details about simulation model development are described in [5]. The simulation model training and validation results are presented in [10].

3.2 DQN training

We formulated the HVAC control problem as a Markov Decision Process (MDP) which involves defining state and action spaces, and reward function.

State space: We used features such as time of the day, indoor temperatures of both the zones, outdoor temperature, and Time Of Use (TOU) electricity price as a state. In the TOU price, the peak price is \$0.25 during 2:00–8:00 p.m. and the off-peak price is \$0.05.

Action space: Actions consist of a set of setpoints, one for each zone, i.e. 70°F, 74°F. It indirectly generates AC's ON/OFF behavior based on the current indoor temperature.

Reward function: The reward function used in this work is $RF = -Energy\_cost$.

3.3 Real-world RL setting and constraints

RL algorithms use multiple episodes of training when trained in a simulated environment. However, in real-world RL problems, it is not possible to perform multiple episodes. In fact, RL-agent is deployed in the environment, continues to interact with the environment, and learns the optimal policy. Moreover, in a real-world setting, RL-agent may get very little chance to explore the environment. To mimic this real-world setting in simulation, we put the following additional constraints:

- DQN training is carried out only for one episode and we performed 50 independent repetitions of the training to captures the effects of random actions and random initialization of agent.
- RL-agent is allowed to explore for a short period at the beginning of the training period. This is controlled by a parameter called Exploration Rate Factor (*ERF*). For example, *ERF*=0.01 allows RL-agent to take random actions during the first 1% of the total training iterations.
- We invoke the data augmentation on day 10.

Various DQN and TimeGAN parameters in the RLDA framework and their values used during training are shown in Table 1.

Table 1: DQN and TimeGAN parameters used

Parameter	Value
Episodes	1
Simulation step (Δt_s)	1 min
Control step (Δt_c)	15 min
Learning rate	0.01
Optimizer	Adam
Reward decay (γ)	0.9
ϵ-greedy value	0.99
Target replacement iterations	200
Initial steps	1440
Batch size	64
Experience replay memory size	100,000
Exploration Rate Factor (ERF)	0.01
[Lower Threshold (LT), Upper Threshold (UT)]	[72°F, 74°F]
Data augmentation day (Δt_a)	day 10
TimeGAN sequence length	16
TimeGAN training iterations	10000

3.4 Evaluation criteria

We used a fixed setpoint baseline of 74°F. Further, we ran RL and RLDA for 62 days of July and August months. Here we refer DQN algorithm without data augmentation as RL. We repeated this training 50 times. During training, we recorded the cumulative energy cost to capture the learning progress. These cumulative energy costs from 50 repetitions, were used to calculate average cumulative energy cost. We then compared the day when average cumulative costs of both RL and RLDA have crossed the baseline. This was used to calculate the improvement in sample-efficiency, i.e. reduction in the data used observed by RLDA over RL.

3.5 Results

Figure 4 shows the comparison of the average cumulative energy cost of 50 repetitions of RLDA and RL training with the baseline's cumulative energy cost. The ribbon plot shows the range of 25% quantile and 75% quantile of cumulative energy cost of 50 repetitions. Figure 4(b) shows the zoomed-in plot of the blue rectangle in Figure 4(a). We observed that RLDA's average cumulative energy cost crossed the baseline on day 28, and RL's (i.e. without data augmentation) average cumulative energy cost crossed the baseline on day 39. This provides us with experimental evidence that the proposed online data augmentation module clearly provided benefits and allowed RL-agent to learn faster, i.e. ≈11 days earlier than the RL without any data augmentation. This shows ≈28% improvement in the sample-efficiency.

3.6 Limitation of the current work

We identified the following limitations of the present work:

- The synthetic trajectories of tuples may contain bad tuples which need to be filtered, which is missing in the current RLDA framework. In the future, we plan to integrate a module to filter the bad tuples based on some quantitative criteria.
- The TimeGAN's training is computationally expensive. Currently, with ≈960 tuples (10 days of tuples), the sequence length of 16, and 10,000 training iterations, it takes an average of ≈50 minutes for training. This can be accelerated in the future using GPUs and distributed training.

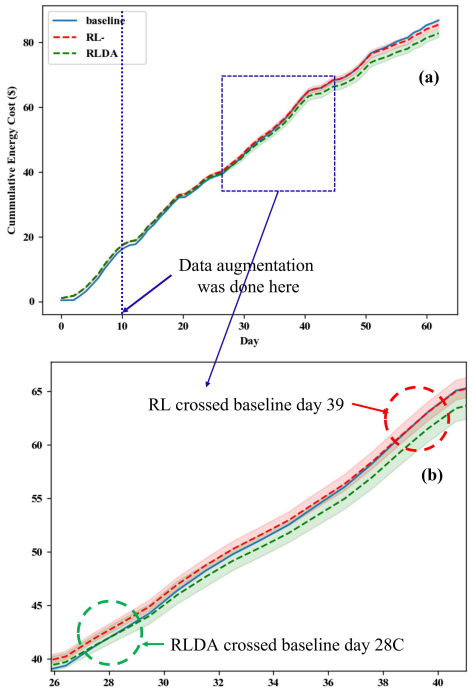


Figure 4: Cumulative cost comparison of RLDA and RL.

- Here, day 10 was chosen arbitrarily to invoke data augmentation. This needs to be automatically computed on-the-fly based on RL-agent's interactions with the environment.
- The current work uses limited seasonal variation, i.e. only one summer season was consider. To obtain more robust results, we will use a year's worth of simulation.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented a Deep Reinforcement Learning framework with online data augmentation (RLDA) to address the sample-efficiency challenge of the real-world RL. We used HVAC control as our use case. For data augmentation, we used the TimeGAN-based time-series generator. The preliminary results are very promising and showed ≈28% of improvement in sample-efficiency, which shows the applicability of the proposed RLDA framework. More research is needed in this direction. In the future, we plan to implement a module that filters the bad tuples generated by TimeGAN based on some quantitative criteria. Also, we will accelerate TimeGAN's computation through the use of GPUs. Moreover, in this work, day 10 was chosen arbitrarily to invoke data augmentation. More research is required in this direction to identify the perfect time to invoke data augmentation. Also, instead of invoking data augmentation only once throughout the training, it can be invoked multiple times periodically during training. Also, we will perform an year worth of simulation to evaluate TimeGAN's response to the seasonal variation.

ACKNOWLEDGMENTS

This research is sponsored by the Artificial Intelligence Initiative as part of the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

REFERENCES

[1] Kadir Amasyali, Kuldeep Kurte, Helia Zandi, Jeffrey Munk, Olivera Kotevska, and Robert Smith. 2021. Double Deep Q-Networks for Optimizing Electricity Cost of a Water Heater. In *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 1–5. https://doi.org/10.1109/ISGT49243.2021.9372205

[2] Kadir Amasyali, Jeffrey Munk, Kuldeep Kurte, Teja Kuruganti, and Helia Zandi. 2021. Deep Reinforcement Learning for Autonomous Water Heater Control. *Buildings* 11, 11 (2021). https://doi.org/10.3390/buildings11110548

[3] Gaby Baasch, Guillaume Rousseau, and Ralph Evins. 2021. A Conditional Generative adversarial Network for energy use in multiple buildings using scarce data. *Energy and AI* 5 (2021), 100087.

[4] Enda Barrett and Stephen Linder. 2015. Autonomous HVAC Control, A Reinforcement Learning Approach. In *Machine Learning and Knowledge Discovery in Databases*, Albert Bifet, Michael May, Bianca Zadrozny, Ricard Gavalda, Dino Pedreschi, Francesco Bonchi, Jaime Cardoso, and Myra Spiliopoulou (Eds.). Springer International Publishing, Cham, 3–19.

[5] Borui Cui, Jeffrey Munk, Roderick Jackson, David Fugate, and Michael Starke. 2017. Building thermal model development of typical house in US for virtual storage control of aggregated building loads based on limited available information. In *Proceedings of ECOS*.

[6] Yan Du, Helia Zandi, Olivera Kotevska, Kuldeep Kurte, Jeffery Munk, Kadir Amasyali, Evan Mckee, and Fangxing Li. 2021. Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning. *Applied Energy* 281 (2021), 116117. https://doi.org/10.1016/j.apenergy.2020.116117

[7] EIA. 2015. Use of energy explained Energy use in homes (U.S. Energy Information Administration (EIA), 2015). https://www.eia.gov/energyexplained/use-of-energy/homes.php

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[9] Fazel Khayatian, Zoltán Nagy, and Andrew Bollinger. 2021. Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties. *Energy and Buildings* 251 (2021), 111334.

[10] Kuldeep Kurte, Jeffrey Munk, Olivera Kotevska, Kadir Amasyali, Robert Smith, Evan McKee, Yan Du, Borui Cui, Teja Kuruganti, and Helia Zandi. 2020. Evaluating the Adaptability of Reinforcement Learning Based HVAC Control for Residential Houses. *Sustainability* 12, 18 (2020). https://doi.org/10.3390/su12187727

[11] Junlin Lu, Patrick Mannion, and Karl Mason. 2022. A multi-objective multi-agent deep reinforcement learning approach to residential appliance scheduling. *IET Smart Grid* 5, 4 (2022), 260–280. https://doi.org/10.1049/stg2.12068 arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/stg2.12068

[12] Muhammad Haiqal Bin Mahbod, Chin Boon Chng, Poh Seng Lee, and Chee Kong Chui. 2022. Energy saving evaluation of an energy efficient data center using a model-free reinforcement learning approach. *Applied Energy* 322 (2022), 119392. https://doi.org/10.1016/j.apenergy.2022.119392

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[14] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[15] Zhe Wang and Tianzhen Hong. 2020. Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN). *Energy and Buildings* 224 (2020), 110299.

[16] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *Proceedings of the 54th annual design automation conference 2017*. 1–6.

[17] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems* 32 (2019).

[18] Liang Yu, Zhanbo Xu, Tengfei Zhang, Xiaohong Guan, and Dong Yue. 2022. Energy-efficient personalized thermal comfort control in office buildings based on multi-agent deep reinforcement learning. *Building and Environment* 223 (2022), 109458. https://doi.org/10.1016/j.buildenv.2022.109458

[19] Yunfei Zhang, Zhihua Zhou, Junwei Liu, and Jianjuan Yuan. 2022. Data augmentation for improving heating load prediction of heating substation based on TimeGAN. *Energy* (2022), 124919.

Modeling and Simulating Thermostat Behaviors of Office Occupants: Are Values more important than Comfort?

Poorvesh Dongre
Denis Gračanin
Department of Computer Science, Virginia Tech
Blacksburg, VA, USA
{poorvesh,gracanin}@vt.edu

Shiwali Mohan
Saman Mostafavi
Kalai Ramea
Palo Alto Research Center (PARC)
Palo Alto, CA, USA
{smohan,smostafa,kramea}@parc.com

Abstract

Existing literature postulates thermal behaviors primarily to be a consequence of thermal discomfort. However, some thermal behaviors, such as changing the thermostat settings in an office building, can cause thermal discomfort to other occupants in the building. Given this social constraint of thermostat behaviors, our paper uses the human-building interaction (HBI) dataset to estimate the impact of office occupants' personal values on their thermostat-up and thermostat-down behaviors using logistic regression. Our preliminary results show that personal values such as agreeableness significantly impact thermal behaviors above and beyond what can be explained by thermal comfort-related features. Finally, we also develop a data-driven agent-based model (DDABM) to understand the emergence of thermostat behaviors influenced by personal values under various environmental conditions.

CCS Concepts

• **Computing methodologies** → **Model development and analysis**; *Machine learning*; • **Social and professional topics** → *User characteristics.*

Keywords

human-building interaction (HBI), machine learning (ML), agent-based model (ABM)

ACM Reference Format:
Poorvesh Dongre, Denis Gračanin, Shiwali Mohan, Saman Mostafavi, and Kalai Ramea. 2022. Modeling and Simulating Thermostat Behaviors of Office Occupants: Are Values more important than Comfort?. In *BuildSys '22: The 9th ACM International Conference on Systems for Energy-Efficient Built Environments, November 09–10, 2022, Boston, Massachusetts.* ACM, New York, NY, USA, 4 pages.

1 Introduction & Background

The indoor environment in office buildings is usually regulated by facility manager to ensure optimal occupant comfort for all occupants. However, occupants may engage in thermal behaviors such as opening/closing doors and windows to improve their comfort levels. In fully air-conditioned offices, occupants resort to changing their clothing level or thermostat settings. But, adjusting the thermostat settings can cause thermal discomfort to other occupants in the building.

In literature, thermal discomfort is often regarded as the primary driver of thermostat behaviors. However, such behaviors are complex and dynamic and are influenced by several other contextual factors, including ease of control, freedom to reposition, and social constraints [7]. Research shows that occupants in shared spaces rely more on psychological coping mechanisms (e.g., tolerating or ignoring discomfort) rather than making any environmental changes [5]. This because they do not want to cause inconvenience to other occupants. Therefore, in this paper, we use personal values of office occupants as an indicator of their social constraints and assess its impact on thermostat behaviors of office occupants in addition to environmental and comfort-related features. We use the human-building interaction (HBI) dataset by Langevin et al. (2015) [6] for our empirical analysis.

The standard methods for modeling occupant behavior are rule-based, stochastic, and data-driven. Researchers have used linear regression, logistic regression, artificial neural networks (ANN), and reinforcement learning (RL) in the data-driven category. Logistic regression is a standard technique for modeling thermal behaviors in commercial buildings [2]. Examples include learning thermostat set-points in office spaces [4] and modeling personal fan use, heater use, and clothing change of office occupants [6]. Therefore, our research also uses logistic regression to model the thermostat behaviors of office occupants.

The second contribution of this paper is a data-driven agent-based model (DDABM) that simulates the thermostat behaviors learned from our logistic regression analysis. Agent-based modeling (ABM) is a powerful tool for understanding the behavior of complex systems, as it can model unanticipated outcomes arising from interactions between agents. However, traditionally, agents' behavior in an ABM is governed by simple rules that may not reflect the true behavior of populations. We demonstrate how a data-driven behavior prediction model can be incorporated into an ABM for better understanding. Our work is inspired by the DDABM framework proposed by Zhang et al. (2016) used for forecasting residential rooftop solar adoption [9].

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3567404

2 Thermostat Behavior Modeling

2.1 Analyzing Predictors

The HBI dataset [6] used in this research is from a year-long study of 24 U.S. office occupants, that recorded 9 environmental features using data loggers and collected 105 occupant-related features using survey questionnaires. The features used to model thermostat behaviors of office occupants are as below:

(1) Environmental Conditions:
- Indoor Operative Temperature
- Outdoor Ambient Temperature

(2) Personal Characteristics:
- Acceptable Thermal Sensations
- Gender

(3) Personal Values:
- Wishes set point to stay at setting they choose
- Choose set point that is most agreeable to others

The indoor operative temperature (mean of indoor ambient and radiant temperature) and outdoor ambient temperature are in degrees Celsius.

Occupants' acceptable thermal sensations are on the ASHRAE 7-point scale [1]. These are used to categorize the occupants into two acceptability groups. If the acceptable thermal sensation of an occupant is less than that of the group, then the occupant is categorized as 0 (cooler climate preference). Otherwise, the occupant is categorized as 1 (warmer climate preference). Occupants' gender is noted as: 1=Male; and 2=Female.

Occupants' personal values indicate if they choose a set point of their preference or a set point that is more agreeable to others. Personal values are on a survey scale of 0 to 6, with 0 showing strong disagreement and 6 indicating strong agreement with a value. K-means clustering was used to group occupants with similar personal values, and the elbow method with inertia was used to estimate the optimal value of K [3].

The features are entered in logistic regression to get the probabilities of two binary behavior outcomes to account for the occupants' three possible thermostat behaviors. These are labeled as below:

(1) thermostat turned up (label=1), and no action (label=0)
(2) thermostat turned down (label=1) and no action (label=0).

Given the discrete nature of features, all data is normalized to bring it to the same scale before using it in K-means clustering and logistic regression.

2.2 Results

2.2.1 Clustering Personal Values Clustering occupants' based on their personal values revealed four occupant personas as shown in Figure 1: (0) empathetic; (1) balanced; (2) indifferent; (3) egocentric. The x-axis and y-axis represent if occupants choose a set-point of their preference if occupants choose a set-point that is more agreeable to others respectively. In other words, the plots shows the variation of occupants' personal values on a normalized scale.

As the name suggests, empathetic personas are more agreeable to others in choosing a thermostat set-point. Balanced personas are more equitable and will choose a set point acceptable to everyone, including themselves. Indifferent personas have no preference when it comes to choosing a set point. Egocentric personas are motivated to choose a set point of their preference. These observations were used to create a new personal values feature "values cluster" in predicting thermostat behaviors.

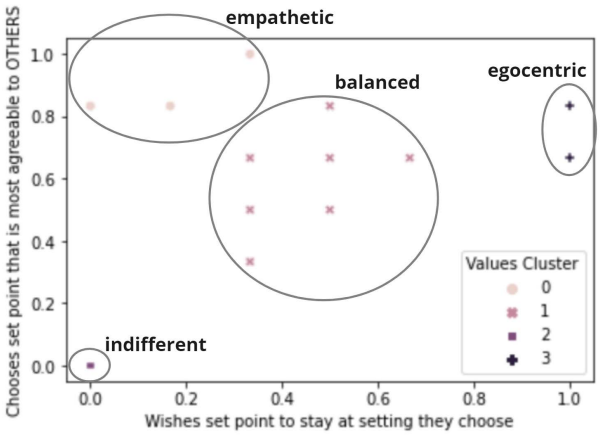


Figure 1: Result from K-means Clustering of Occupants' Personal Values.

2.2.2 Impact of Predictors The results of the logistic regression analysis for thermostat-up and thermostat-down behavior are shown in Table 1 and Table 2, respectively. For both behaviors, there are three regression models, beginning with a model that uses only environmental conditions and personal characteristics, the second using environmental conditions and personal values, and the third using all the predictors combined. The tables show the regression coefficient of each feature and McFadden's R^2 values for each model.

All models (M1, M2, and M3) in Table 1 show that thermostat-up behavior is negatively related to indoor temperature but somewhat positively related to the outdoor temperature. In other words, the indoor temperature has a higher negative impact, and the outdoor temperature has a lower positive impact on thermostat-up behavior. It means that occupants are less likely to increase the thermostat's temperature as indoor temperature increases.

With personal characteristics, the thermostat-up behavior was positively associated with gender in M1 and M3. It means women are more likely to increase the temperature than men. Acceptability groups show negative relations in M1 and positive relations in M3 with thermostat-up behavior. It suggests that the impact of personal characteristics in M1 is getting overridden by personal values in M3. The high impact of personal values on thermostat-up behavior is also observed in M2 and M3. M2 and M3 also have similar accuracy scores but are individually greater than M1. McFadden's R^2 value for M3 is 0.22, highest among the three models, indicating that including personal values offers better explanation of variance in the dataset.

A close to opposite trend is showing with thermostat-down behavior (Table 2). The indoor temperature has a greater positive impact, and the outdoor temperature has a lower negative impact on thermostat-down behavior in all models (M1, M2, M3). It means that occupants are more likely to decrease the thermostat temperature as the indoor temperature increases.

Table 1: Parameter coefficients and McFadden's R^2 values for three logistic regression models (M1 to M3) for thermostat-up behavior.

Category	Parameters	M1	M2	M3
Environmental Condition	Indoor Temperature	-6.57	-8.13	-8.49
	Outdoor Temperature	2.07	1.37	1.80
Personal Characteristics	Acceptability Group	-1.23		1.45
	Gender	0.71		0.38
Personal Values	Own Preference		7.47	
	Others Preference		1.01	
	Values Cluster			8.25
Intercept		2.98	-2.19	-3.37
Accuracy		0.74	0.89	0.87
McFadden's R^2		0.13	0.15	0.22

Table 2: Parameter coefficients and McFadden's R^2 values for three logistic regression models (M1 to M3) for thermostat-down behavior.

Category	Parameters	M1	M2	M3
Environmental Condition	Indoor Temperature	4.57	8.37	8.29
	Outdoor Temperature	-1.96	-2.97	-2.56
Personal Characteristics	Acceptability Group	-2.52		-0.31
	Gender	-1.73		-3.02
Personal Values	Own Preference		5.08	
	Others Preference		-2.99	
	Values Cluster			4.89
Intercept		0.32	-5.81	-5.67
Accuracy		0.78	0.82	0.83
McFadden's R^2		0.06	0.11	0.15

Both the acceptability group and gender were negatively associated with the thermostat-down behaviors, suggesting that men with cooler climate preferences are more likely to decrease the thermostat temperature. The impact of personal values over personal characteristics is also evident in thermostat-down behavior. Occupants who wish to have a set point of their preference are more likely to decrease the thermostat temperature than those who are more agreeable to others. M3 has the highest accuracy and McFadden's R^2 value among the three models for thermostat-down behavior.

3 Data-Driven Agent-Based Model (DDABM)

Existing state-of-the-art literature on simulating occupant thermal behaviors with ABMs uses the predicted mean vote (PMV) model for predicting an agents' behavior. Group-level PMV values are translated to an individual thermal sensation and compared with the agent's sensation to determine a behavioral outcome. The assumption here is that thermal discomfort is the primary motivator of thermal behaviors.

Contrary to that, in our proposed DDABM, maximum likelihood estimates from our logistic regression analysis that includes environmental conditions, personal characteristics, and personal values are used to predict the thermostat behavior of agents. This approach is because our analysis revealed a more significant impact of personal values than thermal comfort-related features in predicting thermostat behaviors.

Table 3: Parameter coefficients and McFadden's R^2 values for logistic regression model of thermal comfort.

Category	Parameter	Model
Environmental Condition	Indoor Temperature	-0.124
	Outdoor Temperature	0.01
Personal Characteristics	Acceptability Group	0.76
	Gender	-
Intercept		2.31
Accuracy		0.56
McFadden's R^2		0.05

Our ABM does estimate the thermal comfort of agents in the simulated environment under varying temperature conditions. However, we use general thermal comfort levels of occupants' measured on a survey scale instead of PMV for thermal comfort predictions. The HBI dataset measured occupants' general thermal comfort levels on a survey scale of 1 to 6, with 1 showing that occupants are very uncomfortable and 6 showing that they are very comfortable. Logistic regression was used to predict whether an agent feels uncomfortable or uncomfortable based on indoor temperature, outdoor temperature, and the acceptability group of agents. General thermal comfort responses between 1 to 4 were labelled 0 (uncomfortable) and those with 5 & 6 were labelled 1 for our binary classification prediction of comfort. Our logistic regression model's low accuracy and fit were another reason not to use it to inform agents' decision-making (Table 3).

We use NetLogo as the interactive development environment (IDE) to develop an ABM that simulates the thermostat behaviors of agents [8]. Environmental conditions are used as inputs to instantiate the simulation. Various agent groups are created based on their personal characteristics and personal values. They randomly spawn in the NetLogo world and interact with the environmental conditions to assess their possible behaviors.

Model M3 from the logistic regression analysis is used to determine the probability of individual agents' thermostat behavior. The probability that an agent takes an action on the thermostat is calculated using a Logit function. If the probability of an agent taking action (thermostat up or thermostat down) is greater than 0.5 then that agent will adjust the thermostat setting, causing the simulated environment's indoor temperature to change accordingly. The agents also interact with the current environmental conditions to report their thermal comfort levels in the ABM. As reported earlier, the thermal comfort model does not influence agents' decision-making because of the underlying hypothesis of this research and low accuracy.

A flowchart showing the process of our data-driven ABM is shown in Figure 2. The ABM continues to loop through this entire process until all agents decide to not take any action on the thermostat.

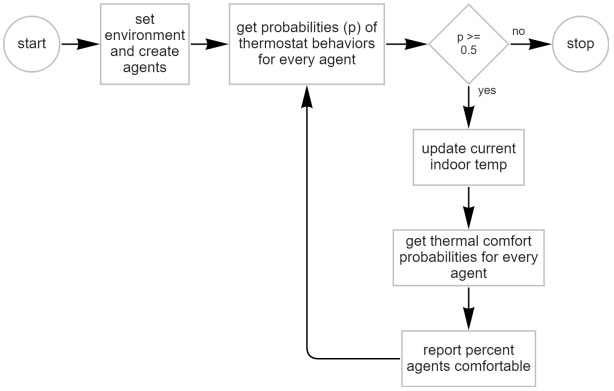


Figure 2: ABM process flowchart.

3.1 Simulation Performance

The first DDABM simulation was performed with 24 agents who had their characteristics and values as learned from the dataset. The indoor and outdoor temperature was set at 20 degrees Celsius. The simulation continued until the indoor temperature achieved a stable value. There were a few fluctuations in the value of indoor temperature due to thermostat behaviors before it achieved a stable value.

The ABM simulation was executed with various indoor and outdoor temperature values. It was observed that the final indoor temperature always drifted to a temperature significantly lower than its corresponding indoor and outdoor temperatures. It can be because the behavior of occupants' who prefer to have cooler indoor environmental conditions for themselves is dominant over other occupants. It was also seen that the final indoor temperature was higher for higher values of initial indoor temperature and outdoor temperature.

The percent comfortable values reported in the ABM remained very low for most simulations even after the final indoor temperature achieved a stable value. It can be due to the low accuracy of the logistic regression model used to predict the thermal comfort of agents.

4 Conclusion

The contribution of this paper is twofold. We use logistic regression to model thermostat behaviors of office occupants and simulate this behavior in NetLogo under various environmental conditions. Indoor & outdoor temperature, acceptable thermal sensations, gender, and personal values were used as predictors of thermostat-up and thermostat-down behaviors. Contrary to existing evidence, our exploration revealed that occupants' personal values motivated their thermostat behaviors more than any other features.

Next, we proposed a preliminary design to demonstrate how the data-driven behavior prediction model can be incorporated in ABM. Unfortunately, the performance of ABM was not realistic, and this could be because of the data amputation techniques used to balance the dataset. Nonetheless, our data-driven approach to create an ABM of thermal behaviors can inform the design of an intelligent thermostat that can learn from the thermostat behaviors of occupants.

In the future, we will collect more data about the contextual factors that affect occupants' thermal behavior to confirm our hypothesis and develop an ABM that simulates realistic thermal behavior under varying environmental conditions and social constraints.

References

[1] ASHRAE. 2020. *Thermal Environmental Conditions for Human Occupancy.* Standard 55. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta.

[2] Salvatore Carlucci, Marilena De Simone, Steven K Firth, Mikkel B Kjærgaard, Romana Markovic, Mohammad Saiedur Rahaman, Masab Khalid Annaqeeb, Silvia Biandrate, Anooshmita Das, Jakub Wladyslaw Dziedzic, et al. 2020. Modeling occupant behavior in buildings. *Building and Environment* 174 (2020), 106768.

[3] Poorvesh Dongre, Asma Aldrees, and Denis Gračanin. 2021. Clustering appliance energy consumption data for occupant energy-behavior modeling. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.* 290–293.

[4] H Burak Gunay, William O'Brien, Ian Beausoleil-Morrison, and Jayson Bursill. 2018. Development and implementation of a thermostat learning algorithm. *Science and Technology for the Built Environment* 24, 1 (2018), 43–56.

[5] Judith Heerwagen and RC Diamond. 1992. Adaptations and coping: occupant response to discomfort in energy efficient buildings. In *Proceedings of the 1992 Summer Study on Energy Efficiency in Buildings.*

[6] Jared Langevin, Patrick L Gurian, and Jin Wen. 2015. Tracking the human-building interaction: A longitudinal field study of occupant behavior in air-conditioned offices. *Journal of Environmental Psychology* 42 (2015), 94–115.

[7] William O'Brien and H Burak Gunay. 2014. The contextual factors contributing to occupants' adaptive comfort behaviors in offices–A review and proposed modeling framework. *Building and Environment* 77 (2014), 77–87.

[8] U. Wilensky. 2016. NetLogo. http://ccl.northwestern.edu/netlogo/. [Last accessed 17 September 2022].

[9] Haifeng Zhang, Yevgeniy Vorobeychik, Joshua Letchford, and Kiran Lakkaraju. 2016. Data-driven agent-based modeling, with application to rooftop solar adoption. *Autonomous Agents and Multi-Agent Systems* 30, 6 (2016), 1023–1049.

Addressing partial observability in reinforcement learning for energy management

Marco Biemann*
Technical University of Denmark
Department of Technology, Management and Economics
Kgs. Lyngby, Denmark
marcob@dtu.dk

Xiufeng Liu
Technical University of Denmark
Department of Technology, Management and Economics
Kgs. Lyngby, Denmark
xiuli@dtu.dk

Yifeng Zeng
Northumbria University
Department of Computer and Information Sciences
Newcastle upon Tyne, United Kingdom
yifeng.zeng@northumbria.ac.uk

Lizhen Huang
Norwegian University of Science and Technology
Department of Manufacturing and Civil Engineering
Gjøvik, Norway
lizhen.huang@ntnu.no

ABSTRACT

Automatic control of energy systems is affected by the uncertainties of multiple factors, including weather, prices and human activities. The literature relies on Markov-based control, taking only into account the current state. This impacts control performance, as previous states give additional context for decision making. We present two ways to learn non-Markovian policies, based on recurrent neural networks and variational inference. We evaluate the methods on a simulated data centre HVAC control task. The results show that the off-policy stochastic latent actor-critic algorithm can maintain the temperature in the predefined range within three months of training without prior knowledge while reducing energy consumption compared to Markovian policies by more than 5%.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Mathematics of computing** → *Markov processes*; **Variational methods**.

KEYWORDS

Reinforcement learning, HVAC control, energy management, POMDP, recurrent neural networks, variational inference.

ACM Reference Format:
Marco Biemann, Xiufeng Liu, Yifeng Zeng, and Lizhen Huang. 2021. Addressing partial observability in reinforcement learning for energy management. In *The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '21), November 17–18, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3486611.3488730

*Also with Norwegian University of Science and Technology.

BuildSys '21, November 17–18, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9114-6/21/11…$15.00
https://doi.org/10.1145/3486611.3488730

1 INTRODUCTION

Smart control in the energy sector is a complex process whose control decisions are influenced by exogenous signals in the form of time-series data, such as ambient temperature, solar radiation, energy prices and energy demand. However, these signals are notoriously difficult to understand and predict, mainly due to their high variance. In recent years, reinforcement learning (RL) in automatic control systems has gained significant attention. RL does not explicitly model these uncertainties but instead learns from the interaction with the environment. Most existing controllers assume the Markov assumption, i.e., the decision making depends only on the current state (not on past states). This assumption is reasonable for chess or Go or for the robot locomotion tasks in OpenAI Gym [5], as their state space consists of positions and velocities of the joints, determining the environment completely. In most real-world control scenarios, the Markov assumption is however inaccurate. To address this, past information is commonly used for decision-making. For image-based observations, Mnih et al. [25] use frame stacking to infer the direction and velocity of moving objects.

In practice, the measurements of the environment are often insufficient to model the system precisely. This applies especially to thermal control applications, as modelling non-equilibrium thermodynamics is challenging. For example, in a data centre, the state information depends on various factors, including CPU load, temperatures of the servers and building materials, dynamics of the airflows in the room, and more. It is not realistic to measure all of these factors, and good approximations are costly. Privacy is an important concern for centralised implementations of multi-agent systems. In these scenarios, only partial information is available, and how to use this limited information for control becomes a challenge.

Such problems can be formulated as a *Partially Observable Markov Decision Process* (POMDP), which is a generalisation of MDP. This formulation can be beneficial for two reasons. History-dependent policies are helpful, as they can recognise the recurrent patterns in the exogenous time-series data. Further, by explicitly formulating that the state is unknown, the agent can learn a latent state representation, which can benefit learning in non-stationary environments.

The idea of maintaining a belief state in partially observable environments in optimal control is due to Åström [1]. It has been studied in the control literature [3, 32] and was revisited recently in RL using variational inference [17]. Another line of work that does not rely on learning explicitly the state rely on recurrent neural networks (RNN) and especially LSTM [15], an architecture addressing vanishing gradients. LSTM was first applied to RL for solving tasks requiring long-term memory [2, 37] and plays a central role in various major achievements of RL in games [18, 27, 35]. Regarding the energy sector, Ruelens et al. [31] highlighted that typical environments in energy management are not fully observable and encoded past observations into an autoencoder. Wang et al. [36] used LSTM in an RL actor-critic algorithm, whereas Zhang et al. [39] used LSTM in a model-based RL algorithm to learn environmental dynamics. Sequence-to-sequence models [6–8] and Bayesian networks [16, 28] were applied to make predictions in model predictive control. Soft actor-critic (SAC) [10] is chosen in this work due to its promising results in energy management [4, 21, 30, 40].

In this paper, we present two approaches based on POMDPs on a simulated HVAC control case study. The first approach changes the network architectures to gated RNNs, taking sequences of past observations as input. The second one consists of inferring the state by learning the belief with variational inference. Both methods demonstrate their effectiveness by obtaining state-of-the-art results in terms of data and energy efficiency.

2 PROBLEM FORMULATION

A *Partially Observable Markov Decision Process* (POMDP) is a generalisation of an MDP. A POMDP is a septuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, p, e, \rho, r)$, where:

- $\mathcal{S}$ is the *state space*, that is all the sufficient and necessary information to model the transitions and rewards;
- $\mathcal{A}$ is the *action space*;
- $\mathcal{O}$ is the *observation space*, corresponding to the measurements available to the agent;
- p are the *state-transition probabilities* of going from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ using action $a \in \mathcal{A}$;
- e are the *emission probabilities* of observing (or measuring) observation $o \in \mathcal{O}$ in state $s \in \mathcal{S}$;
- ρ is the *intial state probability* of starting at state $s \in \mathcal{S}$;
- $r : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$ is the reward function.

The POMDP model reduces to an MDP if $\mathcal{S} = \mathcal{O}$. We assume that all probabilities are unknown to the agent. We define the *history* $h_t = (o_0, a_0, \cdots, a_{t-1}, o_t)$ as the available information at timestep t. We define the probability of trajectories $\tau = (s_t, o_t, a_t)_{t \geq 0}$ by:

$$p_\pi(d\tau) = \rho(ds_0) \prod_{t \geq 0} e(do_t \mid s_t) \pi(da_t \mid h_t) p(ds_{t+1} \mid a_t, s_t)$$

and aim to find a probability distribution π^* that maximises for $\gamma \in (0, 1)$ the following objective:

$$J(\pi) = \mathbb{E}_{\tau \sim p_\pi} \left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \right].$$

Compared with MDP, the policy is conditioned on the whole history h_t, instead of the last observation o_t. That implies that the Q-function $Q_\pi(h_t, a_t)$ needs to be defined over the whole history as well. This is problematic as its dimension grows over time.

3 METHODS

This section will address the above research problem by presenting two distinct approaches, relying on actor-critic methods. The first approach assumes that RNNs parameterise the actor and critic to deal with sequential data. This change can be straightforwardly implemented into standard algorithms, such as Proximal Policy Optimisation (PPO) [33]. The second approach aims to learn a latent representation of the state, that can be used by the actor and critic, instead of the whole history. It starts with an uninformative prior $b_0(s_0)$ and updates the belief $b_t(s_t \mid h_t)$ with new evidence using Bayesian statistics. It is a classical method that has been widely studied, e.g., [1, 20, 32, 34]. This method can lead to more robust policies in non-stationary environments [38] and more explainable results.

3.1 Recurrent neural networks

Standard implementations of actor-critc methods, such as PPO, use separate feed-forward networks for the actor and the critic. Following Jaderberg et al. [19], the architecture is modified so that observations go through an LSTM cell instead. Past observations are encoded into a recurrent state, that is shared between the actor and critic. The long-term memory c_t is beneficial to identify the recurrent patterns of weather data. The advantage of this approach is that the RL algorithm does not have to be modified, except for the architecture and the shorter time horizon, to avoid backpropagating too far through time. Figure 1 describes the neural network architecture of PPO with LSTM used in StableBaselines [14].

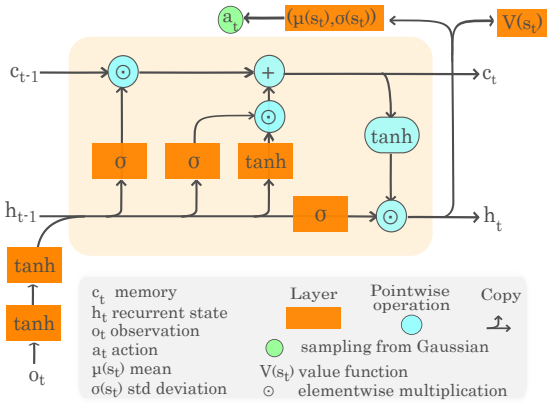
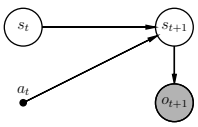


Figure 1: Neural network architecture of PPO with LSTM.

Off-policy methods typically learn the Q-function. The architecture, described in Figure 1 can only be used for the actor; the critic needs to be conditioned on actions as well. Off-policy methods typically use an architecture similar to [13, 29]. A notable difference is that we need to store short trajectories into the replay buffer instead of simple transitions. However, a major concern of LSTM for off-policy methods is its high computational cost, as typically three LSTMs need to be trained [9] for better performance.

3.2 Learning the belief

A *belief* b is a probability measure in $\mathcal{S}$ over states. Given a prior belief b_t and action a_t, we compute the posterior $b_{t+1}(s_{t+1} \mid b_t, a_t, o_{t+1})$, using the new observation o_{t+1}. This can be expressed by the following graphical model:



By applying Bayes' theorem, the belief can, in theory, be updated recursively:

$$b_{t+1}(s_{t+1} \mid b_t, a_t, o_{t+1}) \propto \int_{\mathcal{S}} p(s_{t+1} \mid a_t, s_t) e(o_{t+1} \mid s_{t+1}) b_t(ds_t). \tag{1}$$

By updating the belief this way, we can show that it is a sufficient statistic [3], meaning that the belief summarises all the information from the past required for decision making. In particular, future rewards can be estimated by $Q_\pi(s_t, a_t)$ with $s_t \sim b_t$.

However, the update (1) is intractable as it requires knowledge of the model. Therefore, the belief needs to be updated in an approximate way, which can be done with variational autoencoders (VAE) [22]. A possible way to solve this is the stochastic latent actor-critic (SLAC) algorithm [23], that is a natural generalisation of SAC to POMDPs. Other studies combine VAE with RNNs [12, 17].

We restrict the beliefs to Gaussian distributions, as a posterior of a Gaussian remains Gaussian. The distribution q_φ is an inference model (or encoder), that aims to find a latent representation of the state, given the observed data. The encoder is a neural network updating the state using the most recent data (a_{t-1}, o_t):

$$s_t \sim q_\varphi(\cdot \mid s_{t-1}, a_{t-1}, o_t). \tag{2}$$

The state s_t is what we are interested in, but as we cannot solve (1) exactly, we should ensure that q_φ is a sufficient statistic. That is, the representation s_t needs to encode information generating the observations o_t given by the environment and be updated when an action is taken, so that the next state s_{t+1} explains the next observation o_{t+1}. Therefore, we train a generative model (or decoder) consisting of two networks:

$$\hat{o}_t \sim e_\psi(\cdot \mid s_t),$$
$$\hat{s}_{t+1} \sim p_\psi(\cdot \mid s_t, a_t),$$

where we denote by a hat the observations and states generated by the decoder. As (2) is conditioned on the previous state, we construct a Bayesian network (where $s_0 \sim q_\varphi(\cdot \mid o_0)$ and $\hat{s}_0 \sim \mathcal{N}(0, I)$) to generate a sequence of states $(s_0, \cdots, s_T)$. The networks are updated in order to:

- maximise the likelihood of observations $\sum_{t=0}^{T} \log e_\psi(\cdot \mid s_t)$,
- minimise $\sum_{t=0}^{T} \mathcal{D}_{KL}\left(q_\varphi(\cdot \mid s_{t-1}, a_{t-1}, o_t) \| p_\psi(\cdot \mid s_{t-1}, a_{t-1})\right)$.

The actor $\pi_\theta(a_t \mid s_t)$ and the critics $Q_w(s_t, a_t)$ can be defined and updated the same way as for SAC, conditioned on $s_t \sim q_\varphi$. This approach can also be applied to other off-policy methods. The design used by Lee et al. [23] (and in the experiments) is more complicated and uses a latent variable factorisation for better performance.

SLAC is strongly related to model-based algorithms, such as Dreamer [11]. However, a significant difference is that in SLAC, the environment model is only used in the loss function to infer better states, used by the critic to predict rewards. The predictions of the model are not used by the policy to make decisions or as additional training data.

4 EXPERIMENT

We evaluate the methods based on a classical HVAC control case study simulated with EnergyPlus, whose RL environment was implemented by Moriyama et al. [26] and has been used in [4, 24, 39]. It represents a two-zone medium-sized data centre, whose objective is to reduce energy consumption, while maintaining the indoor temperatures within a predefined range. The observation space consists of the outdoor air temperature, the indoor temperature in both zones and the electricity demand of the servers P_{it} (in kW) and HVAC system P_{hvac}. The actions consist of changing the temperature setpoints of the HVAC system and adjusting the airflow rate in both zones. We used the following reward function:

$$r(s, a) = R_{\text{west}} + R_{\text{east}} - \lambda_P \left(P_{\text{it}} + P_{\text{hvac}}\right),$$

where R_i is the reward obtained when maintaining temperature in zone i in the range. The term R_i is defined as:

$$R_i = \exp\left(-\lambda_1 \left(T_i - T_{\text{tgt}}\right)^2\right) - \lambda_2 \left([T_{\text{min}} - T_i]_+ + [T_i - T_{\text{max}}]_+\right), \tag{3}$$

where $[\text{T}_{\text{min}}, \text{T}_{\text{max}}]$ is the desired range, T_{tgt} is the midpoint of the interval and $[x]_+ = \max(x, 0)$. We used $\text{T}_{\text{min}} = 23\ ^\circ C$, $\text{T}_{\text{max}} = 24^\circ C$, $\lambda_P = 10^{-5}$, $\lambda_1 = 0.5$, $\lambda_2 = 0.1$. As in [26], we used a tighter range in the reward function for better temperature control to further insure that the temperatures lie between $22^\circ C$ and $25^\circ C$. The first term in (3) corresponds to a Gaussian, centered at the desired temperature; the second corresponds to a trapezoid, helping training when the temperature is far away from the target, as the Gaussian would tend too quickly to 0. For more details about the case study, we refer to [4, 26].

The classical algorithms, PPO, PPO-LSTM and SAC, are implemented with the Stable Baselines framework [14] and its original hyperparameters. For SAC-LSTM, we use the architecture from [29]. As the original SLAC architecture is implemented for image observations, it has to be modified for the current case study, following the implementation by Han et al. [12][1]. Given their large influence on the performance of algorithms, we provide details about the used architectures and hyperparameters in supplementary material on our website[2]. We also present there additional figures and results about the experiments done in Section 5.

5 RESULTS AND ANALYSIS

Our discussion will focus on energy consumption and data-efficiency. The ability to maintain the temperature within the desired range

[1]The implementation is available at https://github.com/oist-cnru/Variational-Recurrent-Models. The repository contains implementations of SAC-LSTM, SLAC and their own algorithm SAC-VRM.
[2]https://biemann.github.io/rlem2021

is essential, but all algorithms (except PPO with a feed-forward network) can handle this task within 20 years of training, obtaining similar results (although we observed that SLAC is especially good at this task).

In Figure 2, we compare the power consumption for all algorithms. The algorithms are trained without any prior knowledge. We observe that the models specialised for POMDPs (SLAC, SAC-LSTM, PPO-LSTM) can significantly reduce consumption and outperform the baseline controller implemented into EnergyPlus within one month. The algorithms show similar improvements in terms of temperature control, as shown in Figure 3 for SLAC, and the temperatures lie predominantly in the range after three months. We observed similar results for SAC-LSTM (SAC takes around one year). PPO-LSTM takes a few years until it manages to maintain the temperatures in the range (see Figure 3 for the first episode). It still increases data-efficiency considerably, compared to traditional PPO.

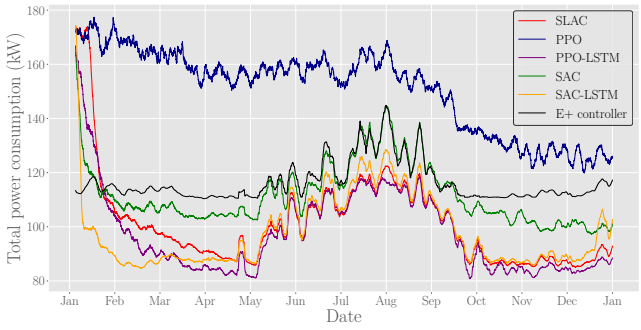


Figure 2: Comparison of algorithms during the first episode

After 20 years of training, we observe in Figure 4 that PPO-LSTM consistently has the lowest consumption, reducing it relative to traditional PPO by 7.8% (and by 20.0% relative to the baseline). SLAC also consistently outperforms the conventional RL algorithms, reducing energy consumption by 5.2% compared to SAC and by 15.3% compared to the baseline. SAC and SAC-LSTM have similar performance in terms of consumption. We observed that SAC-LSTM is better at maintaining temperatures than SAC, but sensitive to hyperparameters and prone to catastrophic forgetting. We found that the other algorithms are robust. This observation and the lower energy consumption suggest that it may be preferable to choose SLAC over SAC-LSTM as a choice of non-Markovian off-policy algorithm.

The significant improvements of SLAC over SAC and PPO-LSTM over PPO in terms of energy consumption, temperature management and data-efficiency suggest that policies that can remember past information may be helpful in stochastic environments. The use of non-Markovian policies can give new insights into the choice between on-policy (PPO) and off-policy (SAC, SLAC) methods, for instance, discussed by [4]. Off-policy methods remain more data-efficient and can reach a good policy quickly, but show only minor improvements after a few episodes. In contrast, PPO-LSTM achieves similar temperature stability after a few episodes while reducing energy consumption significantly.

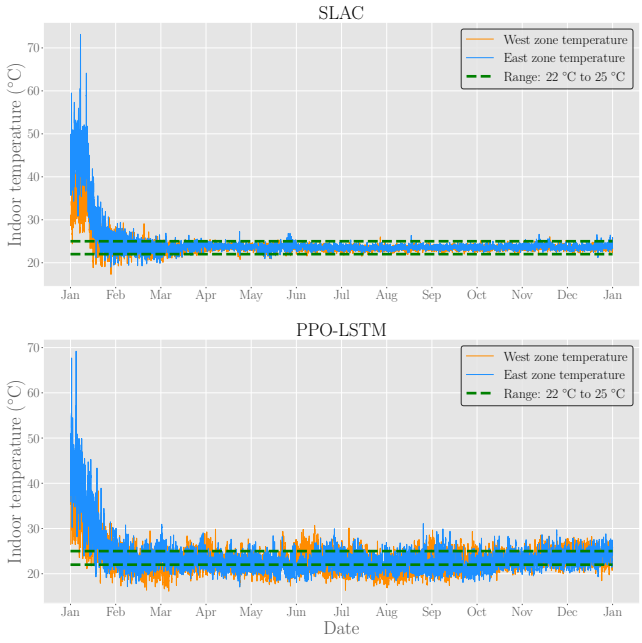


Figure 3: First training episode of SLAC and PPO-LSTM.

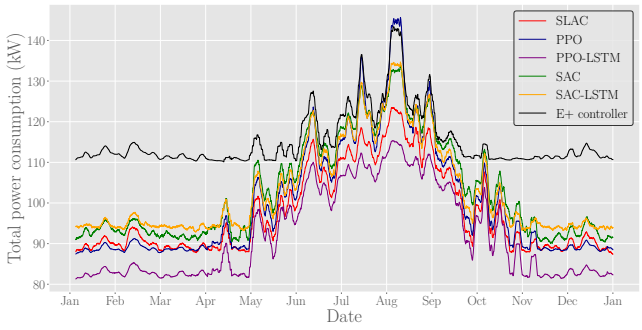


Figure 4: Comparison of algorithms on test location.

6 CONCLUSION

Non-Markovian policies can identify tendencies in the recent past, such as whether temperatures increased in recent hours. This additional insight can allow policies to outperform Markovian policies, suggesting that the formulation of RL in terms of MDP is inaccurate for energy management applications. We found that the SLAC algorithm is more data-efficient and reduces energy consumption, compared to SAC and the baseline respectively. Similarly, the use of an LSTM can improve the results of PPO, and reduce energy consumption significantly. The data-efficiency of SLAC, combined with imitation learning, should close the gap towards training an RL controller directly in the real world.

Future work should investigate whether non-Markovian policies can achieve competitive results with policies using weather or price forecasts as input. An extension of these methods to model-based RL algorithms is natural, as they are based on similar concepts.

REFERENCES

[1] Karl J Astrom. 1965. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Applic.* 10 (1965), 174–205.

[2] Bram Bakker. 2001. Reinforcement Learning with Long Short-Term Memory. In *NIPS*.

[3] Dimitri P Bertsekas and Steven E Shreve. 1996. *Stochastic optimal control: the discrete-time case*. Vol. 5. Athena Scientific.

[4] Marco Biemann, Fabian Scheller, Xiufeng Liu, and Lizhen Huang. 2021. Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control. *Applied Energy* 298 (2021), 117164.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540

[6] Bingqing Chen, Weiran Yao, Jonathan Francis, and Mario Bergés. 2020. Learning a distributed control scheme for demand flexibility in thermostatically controlled loads. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 1–7.

[7] Yize Chen, Yuanyuan Shi, and Baosen Zhang. 2017. Modeling and optimization of complex building energy systems with deep neural networks. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1368–1373.

[8] Matthew J Ellis and Venkatesh Chinde. 2020. An encoder–decoder LSTM-based EMPC framework applied to a building HVAC system. *Chemical Engineering Research and Design* 160 (2020), 508–520.

[9] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. *Proceedings of Machine Learning Research* 80 (2018), 1587–1596.

[10] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*. 1861–1870.

[11] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2019. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*.

[12] Dongqi Han, Kenji Doya, and Jun Tani. 2020. Variational Recurrent Models for Solving Partially Observable Control Tasks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lL4a4tDB

[13] Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, and David Silver. 2015. Memory-based control with recurrent neural networks. *arXiv preprint arXiv:1512.04455* (2015).

[14] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. 2018. Stable Baselines. https://github.com/hill-a/stable-baselines.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Md Monir Hossain, Tianyu Zhang, and Omid Ardakanian. 2021. Identifying grey-box thermal models with Bayesian neural networks. *Energy and Buildings* 238 (2021), 110836.

[17] Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. 2018. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*. PMLR, 2117–2126.

[18] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* 364, 6443 (2019), 859–865.

[19] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=SJ6yPD5xg

[20] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1-2 (1998), 99–134.

[21] Anjukan Kathirgamanathan, Eleni Mangina, and Donal P. Finn. 2021. Development of a Soft Actor Critic deep reinforcement learning approach for harnessing energy flexibility in a Large Office building. *Energy and AI* 5 (2021), 100101. https://doi.org/10.1016/j.egyai.2021.100101

[22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. http://arxiv.org/abs/1312.6114

[23] Alex Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. 2020. Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. *Advances in Neural Information Processing Systems* 33 (2020).

[24] Yuanlong Li, Yonggang Wen, Dacheng Tao, and Kyle Guan. 2019. Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE transactions on cybernetics* 50, 5 (2019), 2002–2013. https://doi.org/10.1109/tcyb.2019.2927410

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[26] Takao Moriyama, Giovanni De Magistris, Michiaki Tatsubori, Tu-Hoa Pham, Asim Munawar, and Ryuki Tachibana. 2018. Reinforcement Learning Testbed for Power-Consumption Optimization. In *Methods and Applications for Modeling and Simulation of Complex Systems*. Springer Singapore, Singapore, 45–59.

[27] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. (2019). arXiv:1912.06680 https://arxiv.org/abs/1912.06680

[28] Nilavra Pathak, James Foulds, Nirmalya Roy, Nilanjan Banerjee, and Ryan Robucci. 2019. A Bayesian Data Analytics Approach to Buildings' Thermal Parameter Estimation. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. 89–99.

[29] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3803–3810.

[30] Giuseppe Pinto, Marco Savino Piscitelli, José Ramón Vázquez-Canteli, Zoltán Nagy, and Alfonso Capozzoli. 2021. Coordinated energy management for a cluster of buildings through deep reinforcement learning. *Energy* 229 (2021), 120725.

[31] Frederik Ruelens, Sandro Iacovella, Bert J Claessens, and Ronnie Belmans. 2015. Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning. *Energies* 8, 8 (2015), 8300–8318. https://doi.org/10.3390/en8088300

[32] Stuart Russell and Peter Norvig. 2002. Artificial intelligence: a modern approach. (2002).

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[34] Richard D Smallwood and Edward J Sondik. 1973. The optimal control of partially observable Markov processes over a finite horizon. *Operations research* 21, 5 (1973), 1071–1088.

[35] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[36] Yuan Wang, Kirubakaran Velswamy, and Biao Huang. 2017. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. *Processes* 5, 3 (2017), 46. https://doi.org/10.3390/pr5030046

[37] Daan Wierstra, Alexander Förster, Jan Peters, and Jürgen Schmidhuber. 2010. Recurrent policy gradients. *Logic Journal of the IGPL* 18, 5 (2010), 620–634.

[38] Annie Xie, James Harrison, and Chelsea Finn. 2021. Deep Reinforcement Learning amidst Continual Structured Non-Stationarity. In *International Conference on Machine Learning*. PMLR, 11393–11403.

[39] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2019. Building HVAC scheduling using reinforcement learning via neural network based model approximation. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 287–296. https://doi.org/10.1145/3360322.3360861

[40] Tianyu Zhang, Gaby Baasch, Omid Ardakanian, and Ralph Evins. 2021. On the Joint Control of Multiple Building Systems with Reinforcement Learning. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 60–72. https://doi.org/10.1145/3447555.3464855

MB^{2}C: Model-Based Deep Reinforcement Learning for Multi-zone Building Control

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

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

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

ABSTRACT

Reinforcement learning has been widely studied for controlling Heating, Ventilation, and Air conditioning (HVAC) systems. Most of the existing works are focused on Model-Free Reinforcement Learning (MFRL), which learns an agent by extensively trial-and-error interaction with a real building. However, one of the fundamental problems with MFRL is the very large amount of training data required to converge to acceptable performance. Although simulation models have been used to generate sufficient training data to accelerate the training process, MFRL needs a high-fidelity building model for simulation, which is also hard to calibrate. As a result, Model-Based Reinforcement Learning (MBRL) has been used for HVAC control. While MBRL schemes can achieve excellent sample efficiency (i.e. less training data), they often lag behind model-free approaches in terms of asymptotic control performance (i.e. high energy savings while meeting occupants' thermal comfort).

In this paper, we conduct a set of experiments to analyze the limitations of current MBRL-based HVAC control methods, in terms of model uncertainty and controller effectiveness. Using the lessons learned, we develop MB^{2}C, a novel MBRL-based HVAC control system that can achieve high control performance with excellent sample efficiency. MB^{2}C learns the building dynamics by employing an ensemble of environment-conditioned neural networks. It then applies a new control method, Model Predictive Path Integral (MPPI), for HVAC control. It produces candidate action sequences by using an importance sampling weighted algorithm that scales better to high state and action dimensions of multi-zone buildings. We evaluate MB^{2}C using EnergyPlus simulations in a five-zone office building. The results show that MB^{2}C can achieve 8.23% more energy savings compared to the state-of-the-art MBRL solution while maintaining similar thermal comfort. MB^{2}C can reduce the training data set by an order of magnitude (10.52×) while achieving comparable performance to MFRL approaches.

CCS CONCEPTS

• **Computing methodologies** → Control methods.

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11…$15.00
https://doi.org/10.1145/3408308.3427986

KEYWORDS

HVAC Control, Model-based Deep Reinforcement Learning, Model Predictive Control

ACM Reference Format:
Xianzhong Ding, Wan Du, and Alberto E. Cerpa. 2020. MB^{2}C: Model-Based Deep Reinforcement Learning for Multi-zone Building Control . In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3408308.3427986

1 INTRODUCTION

Buildings account for 40% of energy usage in the US and 50% of that energy goes to Heating, Ventilation, and Air Conditioning (HVAC) [1]. Rule-based Control (RBC) is widely used to set actuators (e.g., heating or cooling temperature, and fan speed) in HVAC systems [2]. One of the main advantages is that they are easy to understand. However, RBC "rules" are usually set some if-then rules using many times static thresholds based on the rule-of-thumb rules and the experience of engineers and facility managers. They have two fundamental problems: first, they do not scale well with the problem size, as the buildings become larger and more complex, rules must be added; second, they do not handle incomplete or incorrect information very well, an occurrence common in buildings in practice; and finally, they do not necessarily provide a guarantee of optimal control.

Model Predictive Control (MPC) has been widely studied to address these drawbacks by finding optimal control actions based on an analytical building model [3, 4]. Normally, an optimization problem is formulated with the building model and some constraints, and analytic gradient computation is used to optimize over actions and building states simultaneously. However, this often requires convexification of the cost function and first or second-order approximations of building dynamics [5] in order to solve the optimization problem fast and to scale well. As a result, the models used in current solutions are simplified to deal with the parameter-fitting data requirement and computational complexity [3, 4].

Reinforcement Learning (RL) has been widely studied for HVAC control [6–9]. Current solutions mainly adopt Model-Free Reinforcement Learning (MFRL), which learns an optimal HVAC control policy by trial-and-error interactions with a real building. However, MFRL requires a large amount of interactions to converge, e.g., in our experiments, it requires 500,000 timesteps (5200 days) to achieve a high control performance. Although a simulated building model can be used to accelerate the training process, it needs a high-fidelity model, which is hard to calibrate [6, 7]. Recently, Model-Based Reinforcement Learning (MBRL) has been tested for HVAC control to achieve high data efficiency [10]. The HVAC system

dynamics is first learned using a neural network based on historical HVAC data. Based on the learned building dynamics model, an MPC controller tries to find the optimal control action by using a Random Shooting (RS) method [10]. For controlling a single-zone HVAC system, an MBRL-based approach saves approximately 10× training time of the MFRL approach, while achieving comparable performance [10]. However, most of the commercial buildings are multi-zone buildings [11]. In addition to the above scheme not being suitable for multi-zone HVAC systems, MBRL often lags behind the MFRL schemes in terms of control performance (high energy saving while meeting the thermal comfort of occupants).

To overcome these limitations, this paper presents MB^{2}C, a novel MBRL-based HVAC control approach that can achieve both the data/sample efficiency of MBRL and the control performance of MFRL. The design goal of MB^{2}C is to meet the thermal comfort requirements of the occupants while saving as much energy as possible. The energy consumed by a building HVAC system and the thermal comfort of occupants are determined by a set of factors, including current state of all zones, the outdoor weather and the control actions we are about to take (e.g. temperature setpoints). In a multi-zone building, the control actions can be represented as a vector A_s, which is a combination of control actions for all thermal zones. MB^{2}C finds the best A_s from all possible action combinations A_{all} for each control cycle. The best A_s maintains the thermal comfort in its acceptable range for the entire control interval with the lowest energy consumption. MB^{2}C is mainly composed of two parts: (a) a building dynamics model, and (b) an HVAC control algorithm.

Our building dynamics model employs an ensemble of environment-conditioned neural networks. We use a neural network model that takes the current state of the building and the action to perform as input, and outputs a prediction of the next state of the building. To capture model uncertainty, we design a novel weighted ensemble learning algorithm that aggregates the results of multiple building dynamics models by dynamically adjusting the weight of each model according to their accuracy. We also adopt an environment-conditioned neural network architecture by separating the action-depended state items (e.g., zone temperature) and the environment-related state items (e.g., outside temperature), since the latter cannot be actuated by control actions.

Based on a learned building dynamics model, a flexible way to solve the control optimization problem is a shooting method that samples stochastic action trajectories for a number of incoming time-steps [12]. An action trajectory is a set of actions for incoming H time-steps. Every time, H time-steps are evaluated, but only the first action will be executed at the next time-step. For example, RS has been used in the latest MBRL-based HVAC control solution [10], which entails sampling candidate actions from a uniform distribution. However, RS is insufficient to find the best action trajectory, because randomly-shot action trajectories may not include it. We adopt Model Predictive Path Integral (MPPI) control method, which has shown promising performance in robotics control [13]. MPPI derives an optimal control action as the first action of a noise-weighted average over sampled control action trajectories by changing the initial control input and variance of the sampling distribution. We customize MPPI control for building HVAC control under the MBRL-based framework with the best parameter setting.

We implement MB^{2}C in Tensorflow, an open-source machine learning library in Python, with a 3-layer neural network as the building dynamics model and an MPPI-based control algorithm. We study the performance of MB^{2}C and compare it with benchmark methods by controlling a building of five thermal zones. We conduct a variety of simulations in EnergyPlus for evaluation. Extensive simulations reveal that MB^{2}C outperforms the latest model-based DRL method by 8.23% in total energy consumption of the building, without scarifying thermal comfort. Compared with the model-free DRL approach, we reduce the training convergence time by 10.52×, more than an order of magnitude improvement.

2 RELATED WORK

Model Predictive Control for HVAC. MPC solves an optimal control problem iteratively over a receding time horizon. [3] proposed an MPC approach for HVAC control, which minimizes energy use while satisfying occupant comfort constraints. A very recent MPC work, OFFICE [4], proposed a novel MPC framework that optimally manages the trade-off between energy cost and quality of comfort to the building users. OFFICE uses a gray-box approach, where a parametrized first-principled model is used, and the parameter of the model are dynamically learned and updated over time. In our case, we use a black-box approach, where the neural network learns from scratch the relationships between inputs and outputs in the system. Also, the MPC controller used is also different. While OFFICE uses an interior-point method based on a derivable function to find the optimal solution, we use an MPPI controller, which uses sample noise for the exploration around the default values as a search mechanism to find the best optimization solution.

Model-free DRL for HVAC control. Reinforcement Learning has been applied to many areas [14–20]. In particular, MFRL techniques have demonstrated the potential optimal HVAC controls. In MFRL schemes, the agent learns the policy by extensively trial-and-error interaction with the environment. [9] leveraged RL to calculate thermostat set-points to balance between occupant comfort and energy efficiency. [7] implemented and deployed a DRL-based control method for radiant heating systems in a real-life office building. A holistic building control accounting for HVAC, lighting, window opening and blind inclination was studied using branching dueling Q-network (BDQ) in [6]. However, practical application of RL was limited by its sample complexity, i.e. the long training time required to learn control strategies, especially for tasks associated with a large state-action space. Gnu-RL [21] adopted a differentiable MPC policy, which encodes domain knowledge on planning and system dynamics, making it both data-efficient and interpretive. However, they assumed that dynamics of a water-based radiant heating system can be locally linearized. The assumption worked for the problems they considered, but it may not extrapolate to more complex problems like ours.

Model-based DRL for HVAC control. To reduce sample complexity, researchers have adopted model-based deep reinforcement learning for HVAC control [10]. In this work, they proposed an MBRL approach that learns the system dynamics using a neural network. Then, they adopt MPC using the learned system dynamics to perform control with RS method. MBRL method works well when the action and state dimension is low, like single-zone building. They often cannot achieve the final performance as model-free

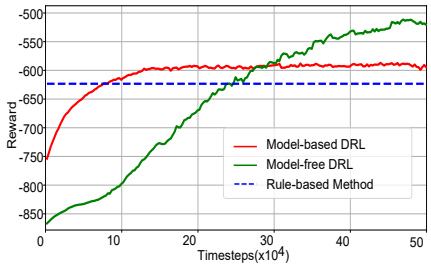


Figure 1: Convergence time and the achieved reward.

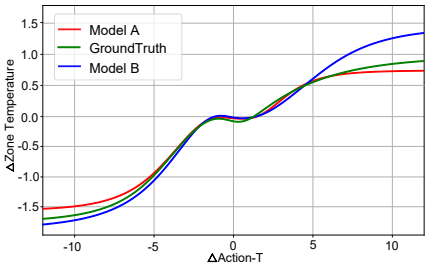


Figure 2: Uncertainty of the building dynamics model.

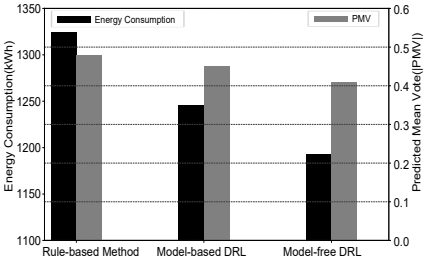


Figure 3: Random shooting in the model-based DRL method.

method when they are applied to high state and action dimensions of multi-zone buildings.

3 MOTIVATION

To understand the performance of a state-of-the-art MBRL method [10], we perform a set of simulations in EnergyPlus for a building with five zones. All system settings are the same as [10], except the state and action dimension is higher for the five-zone building, i.e. a multi-zone building instead of a single zone. We also implement a simple MFRL-based method, Proximal Policy Optimization (PPO) [22], for comparison in this preliminary evaluation. Thermal comfort is measured by PMV [23], which should be controlled within the range (-0.7~0.7). The simulations are conducted with weather data for the month of January. The building is 463 m^2 in Fresno CA. It has windows in all four facades and glass doors in south and north facades. The south-facing glass is shaded by overhangs. For our 5-zone building, the state dimension is 37, including indoor air temperature, humidity, PMV, energy consumption for each zone and related outdoor environmental parameters; and the action dimension is 10, including cooling and heating set points for each zone.

Experiment results. Figure 1 shows the energy-saving performance of model-based and model-free DRL control method with $50{\times}10^4$ time-steps of training data. The reward means the energy-saving performance under the reasonable thermal comfort that is defined in Section 4.2.4. We evaluate the accumulated reward every 2976 time-steps (one month). The performance of the rule-based method is a straight line, because its reward does not change as the weather data and building environment are deterministic.

From Figure 1, we can see that model-based DRL and PPO method need $7.5{\times}10^4$ and $23.75{\times}10^4$ time-steps to behave better performance than rule-based method. For converge time, the model-based method needs $11.5{\times}10^4$ and the PPO method needs $50{\times}10^4$ time-steps. The model-based method is 4.38× more data-efficient than PPO method. However, in the long run, the model-free method eventually outperforms the model-based method. It's easy to see that the model-free method is a trial and error method and the performance increases when using more training data. However, in this case, our model-based method cannot achieve the same performance as model-free method as the training data increases. The model-based method performs well when the action and state dimension is low (e.g., 9 in [10]). However, both the building dynamics model and the control method may not be efficient when the state and action dimension is high, like 47 in our 5-zone building.

Challenge 1 - Model Uncertainty. Neural network models may have epistemic uncertainty, due to the lack of sufficient data to uniquely model the underlying system [24–27]. In an MBRL-based HVAC control system, a building dynamics model predicts the next state of the building, given the current state (e.g. current zone temperature) and a control action (e.g. actuators' temperature set-points). Even a small bias of the building dynamics model may significantly impact the decision of the controller [25, 26]. We conduct an experiment to study this uncertainty of the existing building dynamics model. We use 8000 historical data points to train the model, and 2000 data points for testing.

Figure 2 shows the predictive zone temperature as a function of the action performed. The x-axis shows the temperature differential between the supply temperature (action) and the zone temperature at time t, and the y-axis shows the temperature differential between the zone temperature after and before actuation. The figure depicts the predicted temperatures of two neural network models and the ground truth. These two models have the same architecture and are trained with the same training data, but their training processes start with different initialization states. In the middle region of Figure 2, we have sufficient data, since most of the actions in the historical data do not change the state sharply. In this region, both models can accurately predict the next state. However, when the actions intend to change the state much, we do not have sufficient data for training, and the performance of the two models diverges.

Challenge 2 - Controller Effectiveness. RS generates N independent random action sequences $\{a_t, ...a_{t+H-1}\}$, where each sequence $A_i = \{a_0^i ... a_{H-1}^i\}$ for $i = 1...N$ is of length H action. Given a reward function $r(s,a)$ that defines the task, and given future state predictions $\hat{s}_{t+1} = s_t + f_\theta(\hat{s}_t, a_t)$ from the learned dynamics model f_θ, the optimal action sequence A_{i^*} is selected as the one with the highest predicted reward: $i^* = \arg\max_i R_i = \arg\max_i \sum_{t'=t}^{t+H-1} r(\hat{s}_{t'}, \hat{a}_{t'})$.

Figure 3 studies the energy consumption and thermal comfort of three HVAC control methods, including a rule-based method, a model-based method and a model-free method. To eliminate the impact of model uncertainty for the model-based method, we use the ground-truth states of the building as the results of the building dynamics model (i.e. perfect future state prediction). From the Figure 3, we can see that all three methods can meet the requirement of thermal comfort with same level of PMV value (0.48, 0.45, 0.41). The energy consumption of the model-based method is 4.70% higher than the model-free method. It is caused by RS control, because

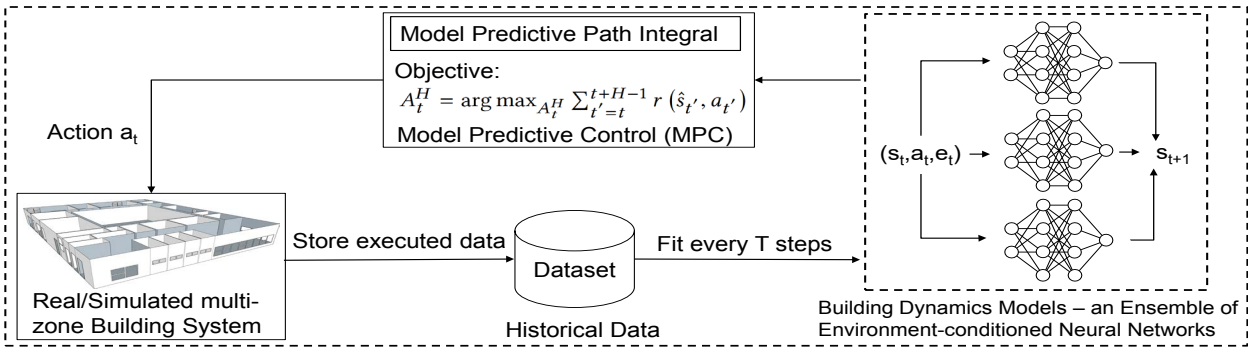


Figure 4: Overall of the Proposed Building Energy Control Framework

the building dynamics model used in the model-based method is perfect in this experiment.

Based on the previous observations, our main goal is to overcome the drawbacks of model uncertainty and controller effectiveness and find a method that is able to match the high performance of model-free methods while having the sample/data-efficiency of model-based methods.

4 DESIGN OF MB^{2}C

In this section, we describe the design of MB^{2}C, including model-based DRL for a multi-zone building control, the building dynamics model and its training details, online control action planning and in-situ update of the building dynamics model.

4.1 MB^{2}C Overview

Figure 4 shows the overview of MB^{2}C as a model-based DRL control approach [26] for multi-zone building HVAC systems. At a high level, MB^{2}C includes two key components, i.e., a building dynamics model and a Model Predictive Path Integral (MPPI) based controller. Our building dynamics model is built by an Ensemble of multiple Environment-conditioned Neural Networks (ENN). It takes the current state of the building HVAC system and a specific control action as input, and outputs the next state of the building HVAC system. Based on the historical data, we train the building dynamics model as a supervised learning process. With the trained building dynamics model, our MPPI-based controller can evaluate different control actions and find the best control action for next time step, which meets the thermal comfort requirement with minimal energy consumption.

When we deploy the system in a building, MB^{2}C executes the best control action by setting corresponding actuators every control cycle. At the same time, we accumulate building data traces, i.e., the next HVAC state determined by the current HVAC state and the executed control action. With the newly collected building traces, we can perform in-situ updating of the building dynamics model periodically (e.g., every week) with a sliding window of 2-months to improve its accuracy, as the seasonality of the data changes during the year. One iterative training process takes 25.32 minutes to finish using a laptop with Intel 4-core i7-6700 CPU and Nvidia GTX 960M GPU, and it can be performed in parallel when the current model is being used in the building; thus, the overhead of the iterative training process does not impact the usage of MB^{2}C in real buildings.

4.2 Model-Based Deep Reinforcement Learning for Multi-zone Building Control

We extend the current MBRL-based method to multi-zone building HVAC control, including the design of those key components.

4.2.1 Preliminaries for DRL. The goal of reinforcement learning is to learn a policy that maximizes the sum of future rewards. At each time step t, the controller is in state $s_t \in S$, executes some action $a_t \in A$, receives reward $r_t = r(s_t, a_t)$, and transitions to the next state s_{t+1} according to some unknown dynamics function $f : S \times A \rightarrow S$. The goal at each time step is to take the action that maximizes the discounted sum of future rewards, given by $\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})$, where $\gamma \in [0, 1]$ is a discount factor that prioritizes near-term rewards. Note that performing this policy extraction requires knowing the underlying reward function $r(s_t, a_t)$ that we use for planning actions under the learned model.

In model-based reinforcement learning, a model of the dynamics is used to make predictions, which is used for action selection. Let $f_\theta(s_t, a_t)$ denote a learned discrete-time dynamics function, parameterized by θ, that takes the current state s_t and action a_t and outputs an estimate of the next state at time $t + \Delta t$. We can then choose actions by solving the following optimization problem:

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

In other words, we will pick the action sequence that maximizes the discounted sum of reward of future H time-steps. In practice, it is often desirable to solve this optimization at each time step, execute only the first action from the sequence, and then re-plan at the next time step with updated state information. Such a control scheme is often referred to as model predictive control (MPC), and is known to compensate well for errors in the model.

4.2.2 State Design. The state is what the building dynamics model takes as input for the next prediction step. In this study, we separate the state into 2 parts: (a) the building state (s_{ti}), which are the state variables that change with our control actions; and (b) the environment state (e_{ti}), which are the state variables that do not change with our control actions.

Building State (s_{ti}) The building state vector that changes over time t for the ith zone consists of the following items: indoor air temperature(°C), indoor air relative humidity (%), PMV, heating energy consumption (kWh) and cooling energy consumption (kWh).

Environment State (e_{ti}) The environment state vector that changes over time t for the ith consists of the following items: outdoor air temperature (°C), outdoor air relative humidity (%), diffuse solar radiation (W/m^2), direct solar radiation (W/m^2), solar incident angle (°), wind speed (m/s), wind direction and occupancy flag (0 or 1). The occupancy flag is an indicator to detect whether there are people in the ith zone, and it is the only element in the vector that changes per zone.

Taking our 5-zone building as an example, the state dimension is 37 including the building, and environment state variables.

4.2.3 Action Design. The action vector (a_{ti}) shows the actuation variables used by the controller to control the building state (s_{ti}). The action state vector that changes over time t for the ith zone consists of the following items: cooling temperature set-point and the heating temperature set-point (both in °C). Given the current state (s_{ti} and e_{ti}) and action (a_{ti}), we want the controller to find the most suitable action combinations ($a_{(t+1)i}$) for all the zones to balance energy consumption and thermal comfort metrics. The action dimension is 10 in our five-zone building.

4.2.4 Reward Design. The reward function controls the optimization parameters that want to be maximized when the agent performs an action (a_{ti}) to transition from the building state s_{ti} to $s_{(t+1)i}$. Both thermal comfort and energy consumption should be incorporated. The reward function is defined as follows:

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

where E is heating and cooling energy consumption for each zone, we use Fanger's formula for the Predictive Mean Vote (PMV) [23] to estimate comfortable temperature bounds for the "standard" occupant within the current seasonal conditions, as defined by ASHRAE standard 55 [28]. The maximum high/low end of the comfort range for Class C environments has PMV values of +/- 0.7. ρ is used to balance the relative importance between energy consumption and thermal comfort. We use ρ = 4 during occupied periods and 0.1 during unoccupied periods since the range of human comfort and energy consumption is different during occupied and unoccupied periods. The reward evaluates the actions to meet the requirement of thermal comfort of all the occupants in the building. N is the number of zones. In the following sections, we will remove the i index for each zone to simplify the notation.

4.3 Learning the Building Dynamics

We require a parameterization of the building dynamics model that can cope with high-dimensional state and action spaces, and the complex dynamics of a multi-zone building. Therefore, we represent the dynamics function $\hat{f}_\theta(s_t, a_t)$ as a multi-layer neural network, parameterized by θ. This function outputs the predicted change in state that occurs as a result of executing action a_t from state s_t, over the time step duration of Δt. Thus, the predicted next state is given by: $\hat{s}_{t+1} = s_t + \hat{f}_\theta(s_t, a_t)$. While choosing too small of a Δt leads to too small of a state difference to allow meaningful learning, increasing the Δt too much can also make the learning process more difficult because it increases the complexity of the underlying continuous-time dynamics.

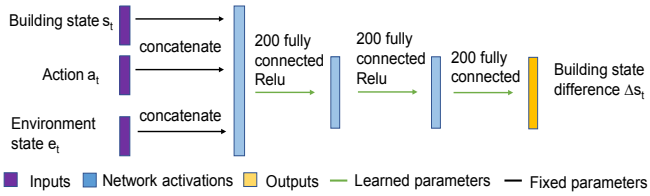


Figure 5: Environment-conditioned neural network for our Building Dynamics Model.

4.3.1 Environment-conditioned Neural Network Architecture. We define a neural network model $\hat{f}_\theta(s_t, a_t)$ for the building dynamics. In order to make the model achieve both good predictive accuracies and tractable computational optimization, we propose a simple and highly effective method for incorporating environment information. We formulate an environment-conditioned dynamics model $\hat{f}_\theta(s_t, a_t, e_t)$ that takes as input not only the current building state s_t and action a_t, but also the current environment state e_t. The model architecture is shown in Figure 5. The building state vector s_t, the action vector a_t and the environment state vector e_t are concatenated together and then are passed through two hidden layers and a final output layer. As opposed to a straightforward outputting of all the related states (building and environment), we produce a prediction of building state difference $\Delta\hat{s}_t$. This reduces the burden of the model to learn the changes in the environment that are not necessary. We provide the ground truth value for environment state, e.g., weather data and occupancy [21].

4.3.2 Weighted Ensemble Learning. As prior work [25, 26] has shown, capturing epistemic uncertainty in the network weights is important in model-based RL, especially with high capacity models that are liable to over-fit to the training set and extrapolate erroneously outside of it. To solve epistemic uncertainty, we propose a weighted ensemble learning algorithm, which approximates the posterior $p(\theta|D)$ with a set of M models, each with parameters θ_i. For deep models, it is sufficient to simply initialize each model θ_i with a different random initialization θ_i^0 and use different batches of data D_i at each training step.

We have M environmental-conditioned models. The input for all M models is the same and it includes the building and environment states and actions. To evaluate the performance of each model, we calculate the mean square error (MSE) of the past C timesteps (4 in our case) for each model compared to the ground truth for N states using Equation 3.

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

We introduce a temporal discount factor ϕ (0.9 in our case) that is used to evaluate how important past model error to the current model error. The temporal discount factor is a value between 0 and 1 since recent prediction cases are more important to the performance of current prediction. After we have the MSE for each model of past C timesteps, we first normalize the MSE to 0-1 scale. $Norm(x)$ is a normalization process, i.e., $Norm(x) = (x - x_{min})/(x_{max} - x_{min})$. Then we calculate the weight ratio W for all models by Equation 4.

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

The sum of all model's weight is 1. After that, we leverage Equation 5 to predict the next state.

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

This allows our method to dynamically adjust the weights in aggregating the M models (M=5 in our case) during the prediction. As the states result in unequal prediction accuracy, our method is more robust against this variance.

4.4 Training the Building Dynamics Model

In this section, we illustrate how we pre-process training data, and train the proposed ENN model.

4.4.1 Data Collection. We collect the training dataset $D(s_t, a_t, s_{t+1})$ by executing the rule-based controller at each time step, and recording the resulting data $\tau = (s(0), a(0), s(1), a(1), ..., s(T-2), a(T-2), s(T-1)))$ of length T. We note that these data are very different from the data the controller will end up executing when planning with this learned dynamics model and a given reward function $r(s_t, a_t)$ (Section 4.5), showing the ability of model-based methods to learn from off-policy data.

4.4.2 Data Preprocessing. We slice the collected data $\{\tau\}$ into training data inputs (s_t, a_t) and corresponding output labels $s_{t+1} - s_t$. In building HVAC control, states can be temperature, humidity ratio, energy consumption, etc. These measurements have various ranges and the weights of the losses will be different if we feed the raw values directly to train the neural network model. Thus, we subtract the mean of the states/action and divide by the standard deviation $x' = \frac{x-\bar{x}}{\sigma(x)}$, where x stands for state or action.

4.4.3 Training the ENN Dynamics Model. ENN model consists of an ensemble of models. To make sure the models behave differently on the same dataset D, we randomly initialize model parameter $\theta_1, \theta_2, ..., \theta_M$ for all the dynamics models and use different batches of data D at each training step. We train the dynamics model $\hat{f}_\theta(s_t, a_t)$ using stochastic gradient descent [29] by minimizing the Mean Square Error (MSE) between predicted delta observation and ground truth delta observation as follows:

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

We use 5-year weather data from Fresno, CA and Chicago, IL for the ENN model training and a completely different one-year for testing in this study. We provide the ENN model with ground truth information on future environment state, i.e. weather and occupancy [21]. In our implementation of ENN, we use the Adam optimizer [30] for gradient-based optimization with a learning rate of 10^{-3}. We train the ENN model with a batch size of 512 and a discount factor $\gamma = 0.99$. The number of epochs is 40. Each dynamics model consists of a neural network of two fully-connected hidden layers of size 200 with relu being nonlinear and a final fully-connected output layer. The weights and biases are initialized using the Xavier initialization process [31]. The number of samples for MPC controllers (RS, CEM, and MPPI) is 1000. The control cycle (timestep) is 15 minutes that is widely used in classic HVAC control [32]. We achieve convergence by 4.75x10^4 time-steps as explained in Section 5.3.1.

4.5 Online Control Action Planning

In our method, we use online planning with MPC to select actions via our model predictions. Given the building state s_t at time t, the prediction horizon H of the MPC controller, and an action sequence $a_{t:t+H} = \{a_t, ..., a_{t+H}\}$, the proposed ENN model $\hat{f}_\theta(s_t, a_t)$ produces a prediction over the resulting data $s_{t:t+H}$. At each time step t, the MPC controller applies the first action a_t of the sequence of optimized actions $A_t^H = \arg\max_{A_t^H} \sum_{t'=t}^{t+H-1} r(\hat{s}_{t'}, a_{t'})$. We adopt the MPPI control method [13] to compute the optimal action sequence.

Model Predictive Path Integral (MPPI) Controller. MPPI control method has been applied to autonomously control a vehicle and get good performance. MPPI is an importance-sampling weighted algorithm and considers an update rule that more effectively integrates a larger number of samples into the distribution update. As derived by recent model-predictive path integral work [13], this general update rule takes the following form for time step t, from each of the K predicted trajectories:

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

Where ω is the importance-sampling weight for each trajectory and ϵ is the noise for exploration. The action for timesteps t of $(i+1)th$ trajectory is the sum of the action for timesteps t of ith trajectory and the noise-weighted average over sampled trajectories.

As shown in the algorithm 1, an initial control sequence is done either by initializing the input buffer with zeros or by using a secondary controller such as rule-based method and using its inputs as the initial control sequence. We first sample H noise from a normal distribution. Then, we compute K trajectories for H finite horizon with Brownian motion. For each trajectory generated, a cost is computed and stored in memory (line 2-7).

In model predictive control, optimization and execution take place simultaneously: a control sequence is computed, and then the first element of the sequence is executed. This process is repeated using the un-executed portion of the previous control sequence as the importance sampling trajectory for the next iteration. In order to ensure that at least one trajectory has non-zero mass (i.e., at least one trajectory has a lowest cost), we subtract the minimum cost of all the sampled trajectories from the cost function (line 9). Note that subtracting by a constant has no effect on the location of the minimum. In the second loop, we get the noise weighted average over K sampled trajectories (lines 10-11). The third loop computes an optimal input sequence using least cost of the trajectories for H finite horizons (lines 12-13). The top of the stack value is given to the actuators (line 14). After that, the whole input control sequence is left shifted by 1 (lines 15-16). To maintain the length of buffer, a_{init} is appended to the input control sequence (line 17). The states are then updated from the ENN model.

4.6 Putting It All Together

We summarize the working flow of MB^{2}C as follows. We first gather historical dataset D using a rule-based policy and randomly initialize model parameter $\theta_1, \theta_2, ..., \theta_M$ for ENN. Then we train the ENN model using this dataset by Equation 6. Finally, we deploy the learned ENN model and our MPPI controller in the real building for HVAC control.

Algorithm 1: MPPI Controller

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

$s_0 \leftarrow GetStateEstimate()$;
for k = 0,1,...,K -1 **do**
  s $\leftarrow s_0$;
  Sample noise $\varepsilon^k = \{\epsilon_0^k, \epsilon_0^k, ...\epsilon_{H-1}^k\} \sim \mathbb{N}(\mu, \sigma)$;
  for t = 1,...,H **do**
    $s_t \leftarrow \hat{f}_\theta(s_{t-1}, a_{t-1} + \epsilon_{t-1}^k)$;
    $Cost(\varepsilon^k)$ += $-reward$ defined by equation 2 ;
$\beta \leftarrow min_k[Cost(\varepsilon^k)]$;
$\eta \leftarrow \sum_{k=0}^{K-1} exp(-\frac{1}{\lambda}(Cost(\varepsilon^k) - \beta))$;
for k = 0,1,...,K -1 **do**
  $\omega(\varepsilon^k) \leftarrow \frac{1}{\eta} exp(Cost(\varepsilon^k) - \beta)$;
for t = 0,1,...,H -1 **do**
  $a_t^* = a_t + \sum_{k=1}^{K} \omega(\varepsilon^k)\epsilon_t^k$;
SendToActuators(a_0);
for t = 0,1,...,H -1 **do**
  $a_{t-1} = a_t$;
$a_{t-1} = Initialize(a_{t-1})$;

For one control execution, we first obtain the current building state from sensors (e.g., zone temperature from a temperature sensor). After that, the best action sequence is sampled by MPPI controller with H horizon and the state is propagated by ENN model by solving the optimization problem defined in Equation 1. We execute the first action of the optimal action sequence in the building by setting corresponding actuators.

When MB^{2}C is running in the building, we can also collect building operation data, which is composed of control action execution records $D(s_t, a_t, s_{t+1})$, including current state, control action, and next state. We add the newly collected data into a sliding window for two months of data and train the ENN model from scratch again. We use a sliding window to adapt to the seasonality of the data, especially weather data. We randomly divide the training data set into a set of batch and update the weight through forward and backward propagation by feeding the data into the model. This process is called one epoch training after traversing all the batch of data. We will repeat this process for multiple epochs (40 in our current implementation) until the model converges. This is an iterative in-situ updating process to improve the accuracy of our building dynamic model.

5 EVALUATION

In this section, we conduct a variety of experiments in EnergyPlus to evaluate the performance of MB^{2}C and three baselines by a set of performance metrics.

5.1 Platform Setup

Building Example and its Dynamics Model in EnergyPlus In this work, we evaluate the performance of MB^{2}C in a building of 463 m^2 at Fresno, California. It is a single floor rectangular building of 5 thermal zones- 4 exterior zones, 1 interior zone. There are windows on all 4 facades. The HVAC system is single duct terminal reheat, which is composed by an Air Handler Unit (AHU) and Variable Air Volume (VAV) boxes. The AHU includes a fan, heating and cooling coils that can change the air's temperature. The VAV boxes take this pre-conditioned air from the main duct, heat it if necessary, and control the airflow provided to each zone.

Since we cannot conduct control experiments in the real building, we leverage a building model in EnergyPlus version 8.6 and conduct simulations with Typical Meteorological Year 3 (TMY3) weather data. In our implementation, the AHU set-point is set by default EnergyPlus control logic, and we only control the heating and cooling set-point in the VAV box.

EnergyPlus has been widely used to evaluate the HVAC control algorithm [6, 7, 10, 21]. There are four reasons why we choose EnergyPlus. First, we do not have one real building that allows us to conduct experiments. MB^{2}C could be deployed in a real building after we finish the ENN model training. Second, it is convenient to generate enough historical training data of rule-based method to train the ENN model. Third, in order to compare with a model-free DRL, we need a significant training data set to train these models since MFRL is not sample efficient. In our case, we need 5200 days (14+ years) of training data, which is unreasonable to obtain from real buildings. Finally, it is easy for us to evaluate the performance of different control algorithms under different locations, seasons and weather profiles.

MB^{2}C System Components As shown in Figure 4, MB^{2}C system includes two main parts: the building dynamics model ENN and the MPPI controller. We also need to store the newly collected building operation data for in-situ update of the building dynamics model. All these three components are all implemented in Tensorflow, which is an open-source machine learning library in Python. We use the building control virtual testbed (BCVTB) [33] for establishing a connection between EnergyPlus and MB^{2}C. We execute the control action by setting the temperature to a specific set point for each zone of our EnergyPlus building model during each control cycle.

5.2 Experiment Setting

We train ENN model based on the weather data from two different cities, Fresno, CA and Chicago, IL due to their distinct weather characteristics. The weather data for Fresno has intensive solar radiation and large variance in temperature, while Chicago is classified as hot-summer humid continental with four distinct seasons.

We compare MB^{2}C with the three baselines. We execute these four control methods to control the building HVAC system using the same weather data for simulation.

Rule-based Method: We implement a rule-based method according to our current campus building control policy for training data generation and comparison evaluation. We assign different zone temperature set-points. Each zone has a separate heating and cooling set-point. The heating set-point is set to 70 °F, and the

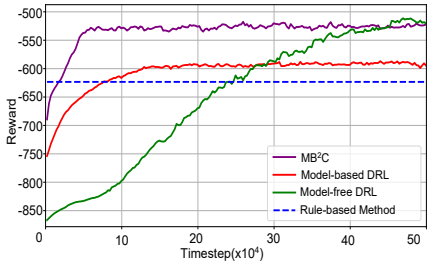


Figure 6: MB^{2}C Achieves both Data-Efficiency and High Performance.

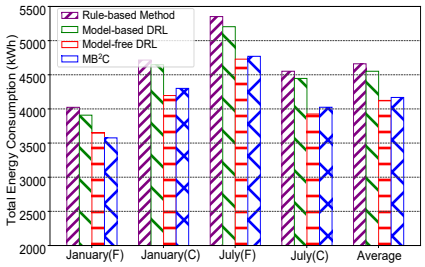


Figure 7: Energy Consumption of MB^{2}C and the Other Baselines.

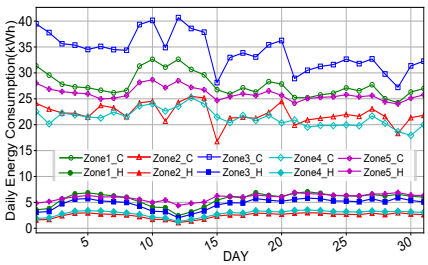


Figure 8: Daily Energy Consumption for Five Zones.

cooling set-point to 74 °F during the warm-up stage. The cooling set-point is limited between 72°F and 80°F, and the heating set-point is limited between 65°F and 72°F.

Model-free DRL: We implement Proximal Policy Optimization (PPO) [22] that is the default reinforcement learning algorithm at OpenAI because of its ease of use and good performance.

Model-based DRL with RS: For the conventional model-based method, we implement the deterministic neural network to model the building dynamics and RS method to choose the heating and cooling setpoints [10].

5.3 Experiment Results

We compare MB^{2}C with the above baselines by a set of performance metrics, including convergence analysis, energy efficiency and thermal comfort. We also study the performance of MB^{2}C, including its daily energy consumption for each zone, the performance gain of its key components, and its parameter setting.

5.3.1 Convergence Analysis. We first study the data efficiency of MB^{2}C and the other three baselines. For this study, we do not limit ourselves to a sliding window of two months for MB^{2}C, since the MFRL method requires copious amount of training data. Figure 6 shows that the accumulated reward of four control methods in each episode during a training process. One episode contains the data collected in one month, corresponding to 2976 time-step. We calculate the reward function every timestep. The reward in Figure 6 is the accumulated reward of one episode, i.e., the sum of the rewards of 2976 time-steps. From the results in Figure 6, we see that the episode reward increases and tends to be stable as the number of training episodes increases. When the episode reward does not change much, it means that we cannot do further to improve the learned control policy and thus the training process converges.

As indicated in Figure 6, MB^{2}C behaves better than rule-based method after the 1.75×10^4 time-steps. In this stage, the ENN model is first learned from off-line historical data. Then it can be deployed into real buildings and leverages the MPPI controller for exploration to further improve its performance. The model-based DRL and model-free DRL need 7.5×10^4 and 23.75×10^4 time-steps to behave better than rule-based method. MB^{2}C achieves 4.28× and 13.57× more data-efficient than model-based DRL and model-free DRL.

For convergence time, MB^{2}C converges faster than both model-based DRL and model-free DRL. MB^{2}C needs 4.75×10^4 and model-based DRL needs 11.5×10^4 time-steps. The model-free DRL needs 50×10^4 timesteps. MB^{2}C is 2.4× and 10.52× data-efficient than model-based DRL and model-free DRL with the same performance as model-free DRL.

5.3.2 Energy Efficiency. Figure 7 depicts the energy consumption results of four control methods. The results reveal that MB^{2}C saves 10.65% and 8.23% energy on average, compared with the rule-based method and model-based DRL. Compared with model-free DRL, MB^{2}C achieves comparable performance. MB^{2}C reduces the energy consumption of HVAC by modeling the complex building dynamics accurately and finding better heating and cooling setpoints.

We can also find that for different seasons and cities, the energy consumption is different. In Fresno, the building consumes 4770.04 kWh in July which is 33.39% more energy than that in January which consumes 3576.07 kWh. The reason is that in July, the outdoor air temperature range at Fresno is 15°C ~ 42°C. We have to keep cooling in daylight. However, in January, the outdoor air temperature range at Fresno -1°C ~ 18°C. This means that we can use outside air that is already in best range of thermal comfort to save energy.

In Chicago, the building consumes 4300.47 kWh in January that is 6.86% more energy than in July, because the weather is cold and the outdoor air temperature range in Chicago is -20°C ~ 15°C. In July, the outdoor air temperature range at Merced and Chicago is similar, 15°C ~ 42°C and 15°C ~ 40°C respectively. But the energy consumption in Fresno is 18.53% higher than the energy in Chicago. The reason is that the average day and night temperature difference for each day is larger than Chicago.

5.3.3 Thermal Comfort. Table 1 presents the average PMV value for all five zones in January and July under Fresno and Chicago weather data. All four control methods can maintain the PMV value in the desired range (-0.7~0.7) for most of the time. The average violation rate of model-based method is 1.97%, which is a little higher than the other three methods, because the controller tries random actions and some of the actions may lead to bad thermal comfort. MB^{2}C achieves a low average violation rate by leveraging more accurate ENN model and more effective MPPI controller.

5.3.4 Daily Energy Consumption for Five Zones. We analyze the daily energy consumption of MB^{2}C for five zones in July at Fresno. As shown in Figure 8, we record the heating energy and cooling energy for each zone per day. The top five hollow line symbols record the trend of cooling energy for five zones respectively. The bottom five solid line shows the trend of heating energy for five zones respectively. The energy spent by the third zone is higher

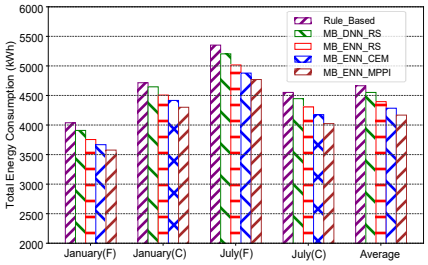


Figure 9: Energy Decomposition.

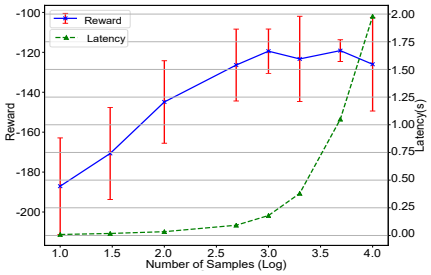


Figure 10: Samples of MPPI Controller.

Figure 11: Horizon of MPPI Controller.

Table 1: Thermal Comfort Statistical Results for Rule-based, Model-based, Model-free and MB^{2}C Schemes

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

than the other zones, because the third zone is south-oriented and the sunlight hits into that zone most of the time.

We also see that both heating and cooling occurs in some days, because the day and night temperature difference is large. In the daylight, the average outdoor temperature is 38°C, and thus we need more energy for cooling. However, at night, the average outdoor temperature is 15°C, and thus we need some heating air to meet the minimum requirement of thermal comfort (in our simulations we assume an office-like environment with students working at night sometimes).

5.3.5 Performance Decomposition. We implement three versions of MB^{2}C with different control methods, i.e., RS (MB_ENN_RS), CEM (MB_ENN_CEM) and MPPI control method (MB_ENN_MPPI). We also compare with the rule-based method and the existing model-based DRL method (MB_DNN_RS).

For MB_ENN_CEM, we implement (Cross-entropy method) CEM [34] controller that begins as the RS method and does this sampling for multiple iterations $m \in \{0...M\}$ at each time step. The top J highest-scoring action sequences from each iteration are used to update and refine the mean and variance of the sampling distribution for the next iteration. After M iterations, the optimal heating and cooling actions are selected to be the resulting mean of the action distribution.

Figure 9 demonstrates the energy consumption of these four methods in two different months and at two different places (Fresno and Chicago). Compared with the rule-based method, MB_DNN_RS can only save 2.42% energy. When the building dynamics model in MB_DNN_RS changed to proposed model (MB_ENN_RS), 3.34% more energy can be saved, which illustrates the efficiency of proposed model. When we change the RS method to CEM method and MPPI method with the proposed model, 2.39% and 4.89 % more energy can be saved that illustrates efficiency of the MPPI controller.

5.3.6 Parameter Setting. MB^{2}C has two important parameters that may influence its performance.

The Number of Samples in the MPPI Algorithm. Figure 10 illustrates the performance of the MPPI controller as the number of sample trajectories is changed. We run MPPI controller with ground truth model to investigate the effect of different number of trajectories (10, 30, 100, 500, 1000, 2000, 5000, 10000). We ran 10 times to calculate the mean and standard reward for each number of trajectories. From Figure 10, we can see that the reward increases quickly as we increase the number of trajectories before 1000 trajectories (power of 3 in the figure). Then it increases slowly after 1000 trajectories, indicating that it is enough for the MPPI algorithm to converge. We also calculate the latency for making one action selection under different number of trajectories. we can see that the latency increases exponentially when trajectories increase. Thus we choose 1000 as the number of trajectories by considering the best reward and lower latency trade-off.

The Length of Horizon in the MPC Process. The horizon refers to the number of steps to look ahead in the MPC process. We investigate the effect of different length of Horizon H in Algorithm 1 to the performance of MPPI Controller. From Figure 11, we can see that the reward increases as the length of H increases and achieves the highest reward when H is 20. Then the reward decreases when we continue increasing the length of H. The reason is that small horizon results in more greedy actions that may not consider future dynamics. Large horizon produces worse actions since the prediction errors aggregate as the horizon becomes larger. We choose 20 for the horizon, which balances the prediction errors and action performance with short latency.

6 DISCUSSION

Building Model Calibration. Currently, we are leveraging the existing five-zone building model in EnergyPlus to evaluate all the existing control methods. We have not done the calibration for this building model since we have no historical operation data of that building. The buildings implemented in EnergyPlus are based on first principles thermodynamical models, so we expect

this model to be similar in performance to a real building. Moreover, it is reasonable to compare all the control methods based on the same building model implemented in EnergyPlus as ground truth. So, for the evaluation done in the paper, we believe this is a fair comparison to test the relative performance of different schemes for "a particular building". If the proposed MB^{2}C was to be deployed in a real building, we would first need to learn the dynamics model from the existing historical data from a real building. Then we deploy the model in the real building for control. If we were to do simulations to test MB^{2}C before real deployment, we need to develop a calibrated EnergyPlus model that matches the target building [6, 7].

Occupancy and Weather Model. In MB^{2}C, we provide the ground-truth value of weather and occupancy for ENN dynamics model. MB^{2}C might be a bit more optimistic since we assume perfect prediction for the weather and occupancy. The errors in prediction may impact controller performance. However, we believe the performance will not significantly deviate from actual results considering model prediction errors. First is that the existing occupancy and weather prediction model [3, 4, 35, 36] show very small prediction error. Second is that MPPI controller outputs the optimal trajectory over the planning horizon. MPPI only takes the first optimal action and re-plans at the next time step based on new observations. This efficiently avoids compounding model error over time.

7 CONCLUSIONS

This paper proposes MB^{2}C, a novel model-based DRL HVAC control system for multi-zone buildings. We develop a new building dynamics model as an ensemble of multiple environment-conditioned neural network models. We also adopt a model predictive path integral control method to perform HVAC control. We compare the performance of MB^{2}C with the rule-based, and state-of-the-art model-based and model-free DRL schemes. The results show that MB^{2}C can achieve 10.65%, 8.23% energy savings on the former and comparable performance with the later, while maintaining (and sometimes even improving) thermal comfort of occupants. Perhaps more importantly, we can achieve this by significantly reducing the training set required by an order of magnitude (10.52× less).

8 ACKNOWLEDGMENTS

We would like to thank anonymous reviewers and our shepherd for their constructive comments and helpful suggestions. This material is based upon work partially supported by the National Science Foundation under grants #CCF-2008837, and a 2020 Seed Fund award from Tecnológico de Monterrey & CITRIS and the Banatao Institute at the University of California.

REFERENCES

[1] Ltd. DR International.2012. 2011 building energy data book. https://openei.org/doe-opendata/dataset/buildings-energy-data-book.
[2] Jyri Salpakari and Peter Lund. Optimal and rule-based control strategies for energy flexibility in buildings with pv. *Applied Energy*, 161:425–436, 2016.
[3] Alex Beltran and Alberto E Cerpa. Optimal hvac building control with occupancy prediction. In *ACM BuildSys*, 2014.
[4] Daniel A Winkler, Ashish Yadav, Claudia Chitu, and Alberto E Cerpa. Office: Optimization framework for improved comfort & efficiency. In *ACM/IEEE IPSN*, 2020.
[5] Narendra N Kota, John M House, Jasbir S Arora, and Theodore F Smith. Optimal control of hvac systems using ddp and nlp techniques. *Optimal Control Applications and Methods*, 17(1):71–78, 1996.
[6] Xianzhong Ding, Wan Du, and Alberto Cerpa. Octopus: Deep reinforcement learning for holistic smart building control. In *ACM BuildSys*, 2019.
[7] Zhiang Zhang and Khee Poh Lam. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *ACM BuildSys*, 2018.
[8] Zoltan Nagy, June Y Park, and J Vazquez-Canteli. Reinforcement learning for intelligent environments: A tutorial. *Handbook of Sustainable and Resilient Infrastructure*, 2018.
[9] June Young Park and Zoltan Nagy. Hvaclearn: A reinforcement learning based occupant-centric control for thermostat set-points. In *ACM e-Energy*, 2020.
[10] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. Building hvac scheduling using reinforcement learning via neural network based model approximation. In *ACM BuildSys*, 2019.
[11] Siddharth Goyal and Prabir Barooah. A method for model-reduction of non-linear thermal dynamics of multi-zone buildings. *Energy and Buildings*, 2012.
[12] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE ICRA*, 2018.
[13] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *IEEE ICRA*, 2017.
[14] Tong Wu and Jorge Ortiz. Towards adaptive anomaly detection in buildings with deep reinforcement learning. In *ACM BuildSys*, 2019.
[15] Bharathan Balaji, Sunil Mallya, et al. Deepracer: Autonomous racing platform for experimentation with sim2real reinforcement learning. In *IEEE ICRA*, 2020.
[16] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, and Rajesh K Gupta. Aces: Automatic configuration of energy harvesting sensors with reinforcement learning. *ACM TOSN*, 2020.
[17] Zhihao Shen, Wan Du, Xi Zhao, and Jianhua Zou. Dmm: fast map matching for cellular data. In *ACM MobiCom*, 2020.
[18] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. Deepapp: A deep reinforcement learning framework for mobile application usage prediction. In *ACM SenSys*, 2019.
[19] Zhi Cao, Honggang Zhang, Yu Cao, and Benyuan Liu. A deep reinforcement learning approach to multi-component job scheduling in edge computing. In *IEEE MSN*, 2019.
[20] Miaomiao Liu, Xianzhong Ding, and Wan Du. Continuous, real-time object detection on mobiledevices without offloading. In *IEEE ICDCS*, 2020.
[21] Bingqing Chen, Zicheng Cai, and Mario Bergés. Gnu-rl: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. In *ACM BuildSys*, 2019.
[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
[23] Poul O Fanger et al. Thermal comfort. analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.*, 1970.
[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.
[25] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *NeurIPS*, 2018.
[26] 2019 Sergey Levine. Model-based reinforcement learning. http://rail.eecs.berkeley.edu/deeprlcourse/.
[27] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *CoRL*, 2020.
[28] A. Standard. Standard 55-2004-thermal environmental conditions for human occupancy. *ASHRAE Inc*, 2004.
[29] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[31] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.
[32] Mesut Avci, Murat Erkoc, Amir Rahmani, and Shihab Asfour. Model predictive hvac load control in buildings using real-time electricity pricing. *Energy and Buildings*, 2013.
[33] Michael Wetter. Co-simulation of building energy and control systems with the building controls virtual test bed. *Journal of Building Performance Simulation*, 2011.
[34] Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, and Pierre L'Ecuyer. The cross-entropy method for optimization. In *Handbook of statistics*. Elsevier, 2013.
[35] Claudia Chiţu, Grigore Stamatescu, and Alberto Cerpa. Building occupancy estimation using supervised learning techniques. In *IEEE ICSTCC*, 2019.
[36] Grigore Stamatescu, Alex Beltran, and Alberto Cerpa. Data-driven comfort models for user-centric predictive control in smart buildings. In *ACM BuildSys*, 2016.

OCTOPUS: Deep Reinforcement Learning for Holistic Smart Building Control

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

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

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

ABSTRACT

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

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

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

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

CCS CONCEPTS

• **Computing methodologies** → Control methods.

KEYWORDS

HVAC control, deep reinforcement learning, energy efficiency

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

1 INTRODUCTION

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

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

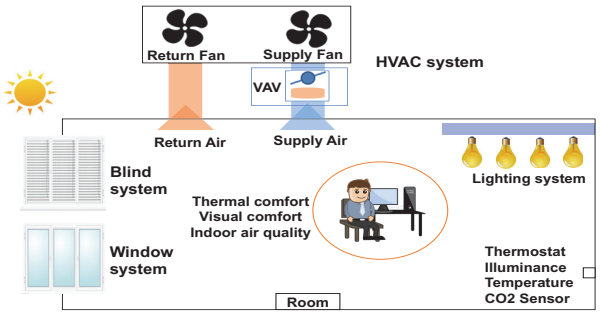


Figure 1: Four Subsystems in a Typical Building.

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

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

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

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

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

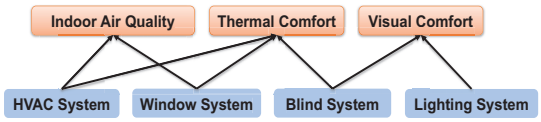


Figure 2: Relationship between Four Subsystems and Three Human Comfort Metrics

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

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

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

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

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

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

2 RELATED WORK

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

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

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

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

3 MOTIVATION

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

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

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

4 DESIGN OF OCTOPUS

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

4.1 OCTOPUS Overview

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

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

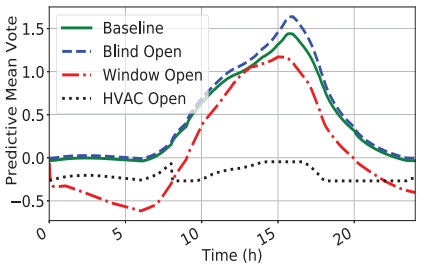


Figure 3: Thermal Comfort, PMV

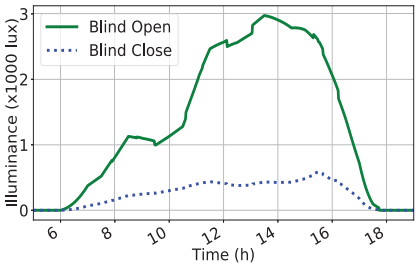


Figure 4: Visual Comfort, Illuminance

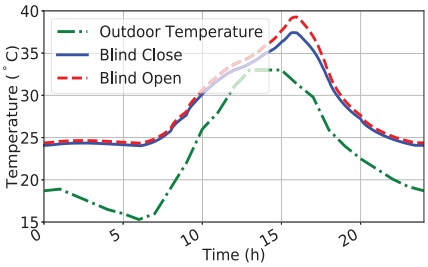


Figure 5: Temperature Effect

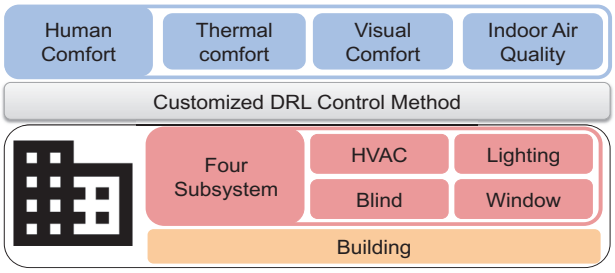


Figure 6: OCTOPUS architecture

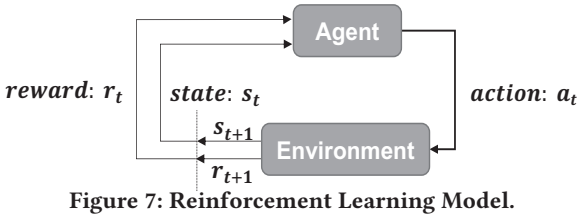


Figure 7: Reinforcement Learning Model.

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

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

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

4.2 DRL-based Building Control

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

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

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

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

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

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

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

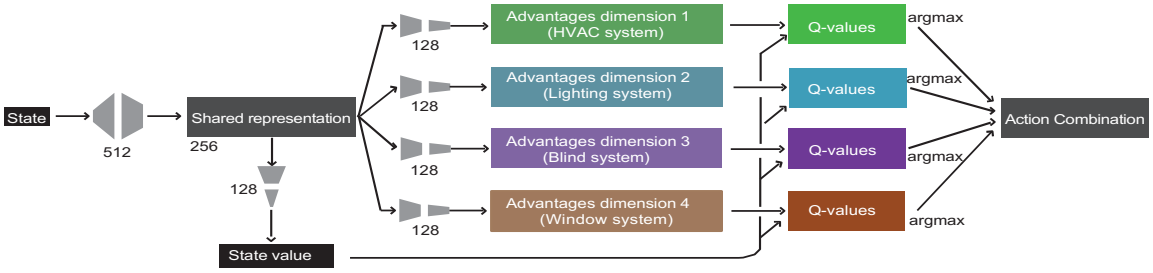


Figure 8: The Specific Action Branching Network Implemented for the Proposed BDQ Agent

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

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

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

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

4.3 Branching Dueling Q-Network

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.4 Reward Calculation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5 IMPLEMENTATION OF OCTOPUS

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

5.1 Platform setup

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

5.2 Rule Based Method

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

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

Table 1: Model Calibration Parameters

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

Table 2: Modeling Error after Calibration

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

5.3 HVAC Modeling and Calibration

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

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

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

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

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

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

5.4 OCTOPUS Training

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

6 EVALUATION

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

6.1 Experiment Setting

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

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

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

6.2 Human Comfort

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

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

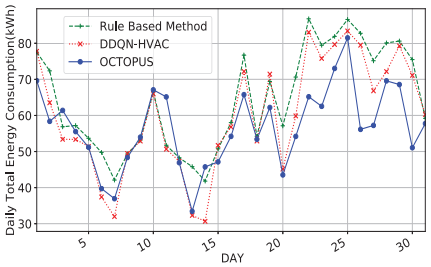


Figure 9: Daily Energy Consumption of Control Methods.

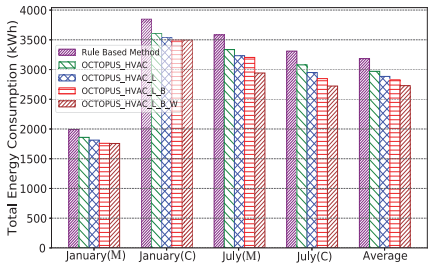


Figure 10: Performance Contribution of Each Subsystem.

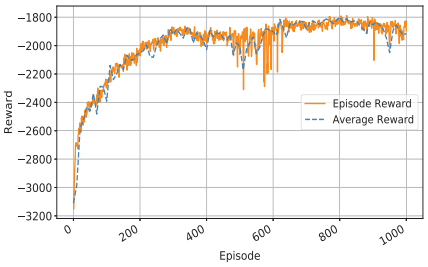


Figure 11: The Convergence of OCTOPUS.

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

6.3 Energy Efficiency

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

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

6.4 Performance Decomposition

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

6.5 Convergence of OCTOPUS training

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

7 DISCUSSION

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

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

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

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

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

8 CONCLUSIONS

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

REFERENCES

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