Safe HVAC Control via Batch Reinforcement Learning

Hsin-Yu Liu*, Bharathan Balaji†, Sicun Gao, Rajesh Gupta, Dezhi Hong
University of California, San Diego, La Jolla, CA, USA, †Amazon

ABSTRACT

Buildings account for 30% of energy use worldwide, and approximately half of it is ascribed to HVAC systems. Reinforcement Learning (RL) has improved upon traditional control methods in increasing the energy efficiency of HVAC systems. However, prior works use online RL methods that require configuring complex thermal simulators to train or use historical data-driven thermal models that can take at least 10^4 time steps to reach rule-based performance Also, due to the distribution drift from simulator to real buildings, RL solutions are therefore seldom deployed in the real world. On the other hand, batch RL methods can learn from the historical data and improve upon the existing policy without any interactions with the real buildings or simulators during the training. With the existing rule-based policy as the priors, the policies learned with batch RL are better than the existing control from the first day of deployment with very few training steps compared with online methods.

Our algorithm incorporates a Kullback-Leibler (KL) regularization term to penalize policies that deviate far from the previous ones. We evaluate our framework on a real multi-zone, multi-floor building—it achieves 7.2% in energy reduction cf. the state-of-the-art batch RL method, and outperforms other BRL methods in occupants' thermal comfort, and 16.7% energy reduction compared to the default rule-based control.

KEYWORDS

HVAC control, Batch Reinforcement Learning, Deep Reinforcement Learning

1 INTRODUCTION

Buildings account for 28% of the global carbon emissions [56], and HVAC (heating, ventilation, and air conditioning) systems account for the majority of building energy consumption[1]. Modern data-driven algorithms have the potential to improve the energy efficiency of traditional HVAC control algorithms. Here we focus on HVAC control in office buildings.

An office building is typically divided into multiple thermal zones, each of which can be controlled locally with a variable air volume unit. The control policy is based on sensor measurements (temperature, humidity, CO_2, airflow, etc) in the thermal zone. Rule-based control (RBC) method is widely used to control the actuators [51], typically in conjunction with proportional-integral-derivative (PID) controllers [17, 33]. Such controls are interpretable but rely on experience and rules of thumb. It becomes challenging to develop and maintain a fine-grained RBC policy for dynamic environments. RBC is also a reactive algorithm as it changes the control settings based on the current measurements. We can improve the control performance if we can forecast the thermal environment characteristics.

We can predict thermal characteristics based on weather conditions, expected usage, and thermal insulation properties. In model-based approaches, thermal states of the building are simulated with simplified linear models, and methods such as model predictive control (MPC) [1, 4, 38, 39, 43, 65] and fuzzy control [7, 53] are used to improve upon RBC policies. However, based on heating/cooling physics, we know that thermal evolution is non-linear with respect to indoor/outdoor conditions and depends on conditions such as orientation with respect to the sun, use of blinds, and wall insulation properties. Therefore, we can devise more accurate models to improve control performance further. Simulators such as EnergyPlus [10] and TRNSYS [29] have been designed to capture the thermal properties of a building. But designing and calibrating such models for a large building requires significant time and expertise. With advances in sensing technologies and machine learning, data-driven models have become popular in recent research.

Reinforcement learning (RL) methods learn via direct interaction with the environment and thus has been studied extensively [25, 62, 67]. They are categorized into model-based RL (MBRL) [15, 42] and model-free RL (MFRL) [9, 24, 70] algorithms. MBRL requires the use of a simulator such as EnergyPlus [10], TRNSYS [29]. Without the pre-training offline, their nature to take exploratory control actions can cause occupant discomfort. MBRL relies on a thermal model learned from historical data, converges faster than MFRL methods. However, even with MBRL, the initial policy is worse than the existing control policy, and it can take weeks/months to improve and converge [16]. By contrast, batch RL can learn directly from historical data. Previous studies have shown that BRL methods can improve on existing policies [20] by exploiting the behavioral policy to identify actions that maximize the reward over an episode with TD-error update (Q-learning) while ensuring that the chosen actions do not deviate too far from the existing policy so the value estimation is more accurate. Typically, there is a hyperparameter to decide the learning trade-off between Q-learning or behavior cloning. Therefore, batch RL is a more efficient method for deployment when historical data is available. *To the best of our knowledge, BRL methods have not been explored for HVAC control.*

We design a BRL-based solution that effectively learns from available historical data without requiring the use of a simulator or explicit modeling of the HVAC system. Our framework guarantees safe system operations by avoiding random setpoint exploration that could damage the equipment and/or make occupants uncomfortable.

Our main contributions are summarized as follows:

[1] https://www.eia.gov/energyexplained/use-of-energy/commercial-buildings.php

*Corresponding authors.
†Work unrelated to Amazon.

978-1-6654-0967-4/22/$31.00 ©2022 IEEE
DOI 10.1109/ICCPS54341.2022.00023

- We propose and develop our framework, a simulation-free control algorithm for energy reduction and thermal comfort co-optimization. Our framework learns from existing historical data only, without requiring a simulator or complex modeling of the space.
- Our method Batch Constrained Munchausen deep Q-learning outperforms state-of-the-art BRL methods by penalizing policies that deviate too far away from the previous policy. It outperforms existing controls from the first day of deployment.
- We compare our framework with several state-of-the-art BRL methods. Our framework reduces the energy consumption by 16.7% compared to the default control, which is 7.2% improvement over the state-of-the-art, while keeping thermal comfort during the entire evaluation period.

2 BACKGROUND AND RELATED WORK

To the best of our knowledge, there is no previous work studying how to co-optimize HVAC energy consumption and occupants' thermal comfort with a completely simulation-free framework deployed on real multi-zone, multi-floor building.

2.1 Model Predictive Control

MPC methods use a model to forecast the outdoor and indoor conditions and optimize for a sequence of control actions that maximizes the given objective. MPC has been studied by several prior works for HVAC control. Aswani et al. [1] use learning-based MPC to control the room temperature to optimize energy consumption. Beltran et al. [4] use occupancy prediction models derived from occupancy data traces and minimize energy consumption while staying within the comfort bounds of the occupants. Maasoumy et al. [39] propose a model-based hierarchical control strategy that balances comfort and energy consumption. They linearize their thermal dynamics model around its operating point and use an LQR supervisory controller that selects the optimal setpoints for the lower level PID controllers. Privara et al. [43] interconnect building simulation software and traditional identification methods to avoid the statistical problems with data gathered from the real building. Winkler et al. [65] develop a data-driven gray-box model whose parameters are learned from building operational data. Together with weather forecast information, this data is fed into the framework to minimize energy costs while satisfying user comfort constraints.

Overall, the known issues of MPC are that it requires an accurate dynamic model, makes convexity assumptions, and the computation cost of computing each control decision is high [3]. RL solutions have been shown to overcome these limitations and outperform MPC methods [41], and the computation cost of a control decision is low as it only requires a neural network inference.

2.2 Reinforcement Learning

2.2.1 Online RL Methods. Researchers have been studied extensively for HVAC control with online RL methods [25, 62, 67]. Zhang et al. [69] jointly optimize visual comfort, thermal comfort, and energy consumption by training for ~12K days in a simulator. OCTOPUS [15] co-optimizes HVAC, lighting, blinds, and window systems and needs ~3.5K days of training. Valladares et al. [57] co-optimize thermal comfort and indoor air quality requiring ~3K days of training. Nagarathinam et al. [41] train a multi-agent policy by taking into account water-side chiller control, and reducing convergence time to 2 years (~700 days) using domain knowledge-based pruning. DeepComfort [24] uses DDPG [35] to co-optimize thermal comfort and energy consumption with 10^4 hour (~417 days) for training. MBBC [16] compares MBRL and MFRL methods with multi-zone control and shows that at least 10^4 of 15-minute time steps (~100 days) are needed to converge. Zhang et al. [68] train in an online fashion to control airflow and temperature. They also take ~100 days to converge.

All prior works need a simulator or a data-driven model to predict the thermal dynamics. Zhang et al. [70] use A3C [40] on real building deployment with model pre-trained on a simulator. HVACLearn [42] proposes an RL-based occupant-centric controller (OCC) for thermostats using tabular Q-Learning with EnergyPlus simulator. Raman et al. [44] implement Zap-Q [14] with ϵ-greedy exploration and compare the model with MPC methods using EnergyPlus. Lu et al. [37] compare on-policy and off-policy RL models with simulated air-conditioned buildings with data-driven models.

Online RL methods, either model-free or model-based, rely on exploration of the state-action space to improve the control policy. Model-free approaches are particularly data inefficient (months to years of convergence time), and therefore, require the use of a simulation model to learn a policy that can be practically deployed. But deploying such policies to a real building requires careful calibration of the simulation model, which is prohibitively time-consuming and expensive. Model-based methods are comparably data-efficient and can use a thermal dynamics model trained with historical data. However, even these methods require weeks to months of real-world interaction for convergence. The initial control policy performance is considerably worse than the existing rule-based policy [16, 41], and becomes a large impediment to adoption. To setup an EnergyPlus model, we need building-specific information, such as materials used to construct the building, that require consulting blueprints. Even after modeling with such details, a separate calibration step is required to ensure the accuracy of the model. Whereas for our reward function model, we used standard heat transfer equations and already available sensor data from the building management system. The reward function can be reused in a new building, whereas EnergyPlus will require redoing the work again. Without a model to simulate airflow, we use the readings and set points from the building management system. These are standard data points available in modern buildings, and our method can be reused as is in other buildings.

2.2.2 Offline RL Methods. Offline methods have not been explored much yet in the building controls domain. GNU-RL [9] implements behavior cloning for HVAC control. In contrast to behavior cloning [52], where the agent simply learns to copy the behavioral policy with an ML model, the BRL method is able to learn from the existing data with Q-update and compensate for the lack of diversity in the buffer by perturbing the selected action with a perturbation network. BRL maximizes the values returned by selecting policy that improves upon the existing policy, rather than imitating it.

Previously, Ruelens et al.'s works focus on electricity cost optimization [49], demand response [48], and energy efficiency of heat

pump [50] using fitted Q-iteration (FQI [46]). Vazques et al. [58, 59] balance comfort and energy consumption of a heat pump using FQI. Yang et al. [66] implement Batch Q-Learning for low exergy buildings. The closest work to ours is Wei et al.'s work [63], where they control airflow using offline training using a modified Q-learning algorithm, where they clip and shrink the reward value. Unlike our method, the experiments are done in simulators, do not control zone temperature setpoint, and only consider temperature as a proxy for thermal comfort.

Algorithms such as FQI, Batch Q-learning, and Wei et al.'s DQN heuristic are all based on pure off-policy algorithms. Fujimoto et al. [23] show that off-policy methods exacerbate the extrapolation error in a completely offline setting. The errors occur because the Q-network is trained on historical data but exploratory actions yield policies which are different from the behavioral ones. They propose Batch Constrained Q-learning (BCQ) [23] which restricts selected actions to be close to those in the historical data and outperforms prior approaches. BCQ uses a Variational AutoEncoder (VAE) [28] to reconstruct the predicted actions given current states according to existing data.

BCQ is designed for complete offline, off-policy learning to penalize policies that are far from the behavioral policies in the replay buffer. We build on top of the BCQ algorithm to further constrain new policy to be close to the previous one. We enforce this constraint through Kullback-Leibler (KL) divergence between the learned policy and historical policy [61]. We show that our algorithm performance is more stable than BCQ in our real building evaluation.

We use the existing dataset as the prior experience, since the rules are made by domain experts, its behavioral policy is *safe* cf. random initialized online policy. In this paper, we focus on the performance of the algorithm in the initial days (one week) of deployment and leave the long-term performance evaluation as future work.

3 DESIGN OF OUR FRAMEWORK

3.1 BRL-based Control Framework Setup

As shown in Fig. 1, we first obtain historical data and process them into replay buffer containing the transitions tuples. At each time step, the BRL model will randomly sample a mini-batch from the replay buffer and train the target networks with the transitions sampled. Periodically (according to the *eval_freq* in Alg. 1), we evaluate the trained agent's policy (the *select_action* function in Alg. 2) on real building zones to observe the states from our system's readings and calculate the reward. The average rewards over time are shown in Fig. 5.

We use the episodic formulation as this is the standard procedure in BRL literature [23, 32, 36]. In our formulation, the episode ends if the predicted thermal comfort is out of the thermal comfort range, i.e. absolute value larger than 0.5. Therefore, the agent is trained for an arbitrarily long episode length as long as it does not impact comfort. If we use a fixed episode length such as 24 hours, the agent will optimize for that period. We use a time step of 9 minutes because that is the data-writing period for our building management system. We choose the minimum possible time step to minimize system response time and reduce any discomfort to occupants.

We represent the agent and its environment as a Markov Decision Process (MDP) defined by a tuple, $M_{\mathcal{B}} = (\mathcal{S}, \mathcal{S}', \mathcal{A}, P, R, \gamma)$, where $\mathcal{A}$ is the action space in the batch $\mathcal{B}$, $\mathcal{S}$ is the state space, $\mathcal{S}^{'}$ is the arriving state space where $\forall s' \in \mathcal{S}'$ corresponds to $s \in \mathcal{S}$ at a certain time step t such that $s = s_t$, $s' = s_{t+1}$. $P(s'|s, a)$ is the transition distribution, $R(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. The goal of our BRL model is to find an optimal policy $\pi^*(s) = \text{argmax}_{a\ s.t.(s,a)\in\mathcal{B}}\ Q^{\pi}_{\mathcal{B}}(s, a)$, which maximizes the expected accumulative discounted rewards.

More specifically, we have the following:

- *State*: We use the following attributes for the RL process to evaluate the policy: indoor air temperature, actual supply airflow, outside air temperature, and humidity. These states include the features needed for thermal comfort estimation s_t^{TC} and those that represent the responses of actions as RL states s_t^{RL}.
- *Action*: We control two important parameters, namely, zone air temperature setpoint (a_t^{ZNT}) and actual supply airflow setpoint (a_t^{Sup}). Both are in continuous space and the action spaces are normalized in the range of $[-1, 1]$.
- *Environment*: Real building HVAC zones across three different floors. Every room is a single HVAC zone, and all these rooms are used as lab space and work office.
- *Reward*: We monitor the thermal states of the space as well as the thermal comfort index predicted by a regression model, and then make control decisions with the actions selected by the BRL model. Our reward function penalizes high HVAC energy use and discourages a large absolute value of the thermal comfort index, which indicates discomfort to occupants. Our reward function at time step t is:

$$R_t = -\alpha|TC_t| - \beta P_t, \tag{1}$$

where α, β are the weights balancing between 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 [2]. The DRL agent co-optimizes HVAC energy reduction and occupants' thermal comfort.

3.2 Thermal Comfort Prediction

As we need to calculate the thermal comfort level as required by our reward function, we adopt the widely used predicted mean vote (PMV) [19] measure as our thermal comfort index. In this metric, there are degrees of satisfaction, ranging from −3 (cold) to 3 (hot), where PMV within the range from −0.5 to 0.5 is considered thermal-comfortable.

We adopt the ASHRAE RP-884 thermal comfort data set [13] and train a simple gradient boosting tree (GBT) model [27] to predict the thermal comfort by taking the current thermal states given by our building system in real-time. We show the evaluation of the effectiveness in Fig. 2 with such a simple GBT-based thermal comfort index.

3.3 Batch Reinforcement Learning for Control

We take a BRL-based method, namely, batch-constrained deep Q-learning (BCQ) [23] as our foundation and make improvements

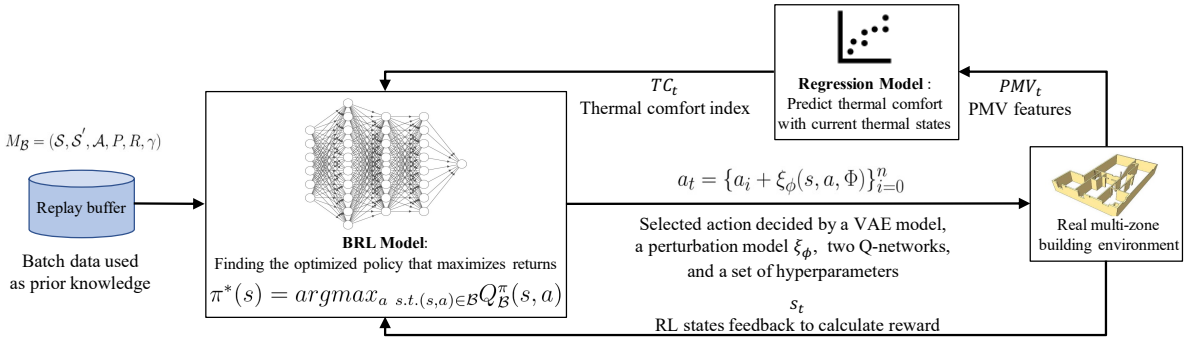

Figure 1: Overview: Our batch reinforcement learning model selects actions that co-optimize thermal comfort for occupants and energy consumption of HVAC system.

on it. BCQ is a pure offline, off-policy RL method that avoids the extrapolation errors induced by the incorrect value estimation of out-of-distribution actions selected out of existing dataset.

As illustrated in Fig. 1, for each time step t, we obtain state information from the sensors in the building. To only calculate the reward but not updating the models. BCQ first samples a mini-batch of data (the size of the mini-batch is set as a hyperparameter) from the entire set of historical data. Then, it trains a parametric generative model G_ω, a conditional VAE on the batch to model the distribution by transforming an underlying latent space. The encoder $E_{\omega_1}(s, a)$ takes a distribution of state-action pairs and outputs the mean μ and standard deviation σ of a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. A latent vector z sampled from the Gaussian is passed to the decoder $D_{\omega_2}(s, z)$ which outputs an action. The loss function of VAE consists of two parts: reconstruction loss and the KL regularization term.

$$\mathcal{L}_{recon} = \sum_{(s,a)\in\mathcal{B}} (D_{\omega_2}(s,z) - a)^2, z = \mu + \sigma \cdot \epsilon, \epsilon \sim \mathcal{N}(0,1)$$
$$\mathcal{L}_{KL} = D_{KL}(\mathcal{N}(\mu,\sigma)||\mathcal{N}(0,1)),$$
$$\mathcal{L}_{VAE} = \mathcal{L}_{recon} + \lambda\mathcal{L}_{KL}.$$

VAE here aims to produce only actions which are similar to existing actions in the batch given the current state. The purpose of the perturbation model $\xi_\phi(s, a, \Phi)$ is to increase the diversity of seen actions, it adjusts the value of the selected action a in the range of $[-\Phi, \Phi]$. It could compensate for the lack of diversity in the batch data, as a trade-off of inaccurate value estimation. By adjusting the hyperparameters n and Φ, it could behave similarly to behavior cloning with $n = 1$ and $\Phi = 0$, or similarly to traditional Q-learning when $n \to \infty$ and $\Phi \to a_{max} - a_{min}$.

$$\phi \leftarrow \underset{\phi}{\text{argmax}} \sum_{(s,a)\in\mathcal{B}} Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s).$$

At the core of BCQ is the value estimation networks, a pair of Q-networks $Q_{\theta_1}(s, a)$ and $Q_{\theta_2}(s, a)$. By taking a weighted minimum between the two values as a learning target y for both networks. On the other hand, for the actor network, at first, n actions are sampled with respect to the generative model, and then adjusted by the target perturbation model, before being passed to each target Q-networks for updates:

$$y = r + \gamma \max_{a_i} \left[\lambda \min_{j=1,2} Q_{\theta'_j}(s', \tilde{a}_i) + (1-\lambda) \max_{j=1,2} Q_{\theta'_j}(s', \tilde{a}_i)\right], \quad (2)$$

where r is the reward, γ is the discount factor, λ is the minimum weighting in double-Q learning, $\theta_{j=1,2}$ are weights of the two critic Q-networks.

We propose an improvement on the BCQ algorithm, called Batch Constrained Munchausen RL (BCM), that encourages the agent to update policy similarly to the previous one using a regularization term in the Q-update. With respect to other aspects, BCM inherits BCQ's characteristics and acts as an intermediate state of behavior cloning and Q-learning.

The idea of the BCM algorithm is the following: we adopt the regularization term in Munchausen RL (M-RL) [61] which penalizes the policies which deviate far from the previous policy with Kullback-Leibler (KL) divergence [31, 60]. M-RL utilizes the current policy as one of Q-update's learning signals. $KL(\pi_1||\pi_2) = \langle \pi_1, \ln \pi_1 - \ln \pi_2 \rangle$. The other term added in M-RL is the entropy term which penalizes the policies that are too far away from the uniform distribution, where $\mathcal{H}(\pi) = -\langle \pi, \ln \pi \rangle$. In offline settings, this term does not help improve the Q-update since we cannot accurately estimate uniform policy if we have only static data. We do not encourage exploration as the online mode in the original M-RL settings. Our problem is focused on conservative and safe policies exclusively selected from the batch with a small amount of perturbation. It helps to avoid the lack of diversity within state-action visitation in the batch distribution.

$$y = r + \alpha_m \left[\tau_m \ln \pi_{\hat{\theta}}(a_t|s_t)\right]^0_{l_0} + \gamma \max_{a_i} \left[\lambda \min_{j=1,2} Q_{\theta'_j}(s', \tilde{a}_i) + (1-\lambda) \max_{j=1,2} Q_{\theta'_j}(s', \tilde{a}_i)\right], \quad (3)$$

where $\pi_{\hat{\theta}} = softmax(\frac{Q_{\hat{\theta}}}{\tau})$, the target Q after soft clipping in double Q-learning, α_m is the M-RL scaling parameter, τ_m is the entropy temperature parameter, and l_0 is the clipping value minimum, since the log-policy term is not bounded and can cause numerical issues if the policy becomes too close to deterministic. We replace $\tau\ ln\ \pi(a|s)$ by $[\tau\ ln\ \pi(a|s)]^0_{l_0}$, where $[\cdot]^y_x$ is the clipping function. The other added term in original M-RL algorithm is the entropy term which encourages policies to be close to uniform distribution. We do not use it as it is not applicable for offline settings [61]. Once we choose the action using BCM, we adjust the corresponding setpoints through a building operating system (BOS) [30, 64]. The environment reflects the real response of action applied with a time delay d, so our framework waits for d to get data s_t from the sensors. Also, a PMV feature vector PMV_t is fed into the regression model for thermal comfort prediction. According to the

Algorithm 1: HVAC control via our framework

Input : Batch data $\mathcal{B}_f$ for a certain floor f, time horizon T, floor set $\mathcal{F}$, zone/room set $\mathcal{Z}$, and delayed response time d, target network update rate τ, mini-batch size b, max perturbation to selected actions Φ, number of sampled actions n, minimum weighting λ, evaluation frequency $eval\_freq$, M-RL scaling factor $\alpha_m \in [0, 1]$, and entropy temperature parameter τ_m

Output : Reward, next state, and action selected by BCM

Initialize: HVAC Environment Env, RL agent BCM

$d_a = dim(a), d_s = dim(s)$;

for $f \in \mathcal{F}$ **do**

  $BCM_f = BCM(d_s, d_a, \gamma, \tau, \lambda, \phi, \alpha_m, \tau_m)$;

  for $z \in \mathcal{Z}$ **do**

    $0 \leftarrow$ t;

    while $train\_iteration < T$ **do**

      $BCM_f.train(\mathcal{B}_f, b, n)$;

      if $t\%eval\_freq == 0$ **then**

        $s_t^z = Env^z.getThermalState(t)$;

        $TC_t^z = Env^z.getPredictedTC(s_t^z)$;

        $a_t^z = BCM_f.select\_action(s_t^z)$;

        $s_{t+1}^z, r_t^z = Env^z.step(a_t^z, s_t^z, d)$;

        t += 1 ;

      end

    end

  end

end

Algorithm 2: BCM training algorithm

Input : Batch data $\mathcal{B}_f$ for a certain floor f, target network update rate τ, mini-batch size N, max perturbation to selected actions Φ, number of sampled actions n, minimum weighting λ, evaluation frequency $eval\_freq$, M-RL scaling parameter α_m, and entropy temperature parameter τ_m

Output : Updated target networks

Initialize: RL agent BCM, Q networks $Q_{\theta_1}, Q_{\theta_2}$, VAE generative network $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$, perturbation network ξ_ϕ, random parameter $\omega, \phi, \theta_1, \theta_2$, and target networks $Q_{\theta_1'}, Q_{\theta_2'}, \xi_{\phi'}$ with $\theta_1' \leftarrow \theta_1, \theta_2' \leftarrow \theta_2, \phi' \leftarrow \phi$

for $t \leftarrow 0$ **to** T **do**

  Sample mini-batch N transitions (s, a, r, s') from $\mathcal{B}_f$;

  $\mu, \sigma = E_{\omega_1}(s, a), \tilde{a} = D_{\omega_2}(s, z), z \sim \mathcal{N}(\mu, \sigma)$

  $\omega \leftarrow \text{argmin}_\omega \sum(a - \tilde{a})^2 + D_{KL}(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, 1))$

  Sample n actions: $\{a_i \sim G_\omega(s')\}_{i=1}^n$;

  Perturb each action: $\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^n$;

  Set value target y (Eqn.3);

  $\theta \leftarrow \text{argmin}_\theta \sum(y - Q_\theta(s, a))^2$;

  $\phi \leftarrow \text{argmax}_\phi Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s)$;

  Update target networks: $\theta_i' \leftarrow \tau\theta + (1 - \tau)\theta_i'$;

  $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$;

end

prediction of regression model TC_t and the RL states s_t, we calculate the reward using Eq.(1). We repeat this process until reaching the maximum number of time steps T. Details of the HVAC control via BCM algorithm are described in Algorithm 1.

4 EVALUATION

4.1 Data Collection and Pre-processing

The data we use from all the sensors and control points are recorded every 9 minutes via a BOS. We obtain data of an entire year, from the beginning of July 2017 to the end of June 2018 of fifteen rooms across three different floors in a building. The batch for each floor, or the *buffer*, contains around $200K$ transitions (2F:∼$260K$, 3F:∼$193K$, 4F:∼$172K$), and it might differ from one to another due to varied system maintenance duration throughout the year. Since the rooms on the same side of a floor often share similar thermal dynamics, we thus create batch data for each floor to ensure that the replay buffer reflects each variable air volume (VAV)'s thermal dynamics precisely. We set each room to its maximum occupancy, which is obtained from our campus facility information management system, and in evaluation, we assume full occupancy the entire time. We could easily modify the problem formulation by taking occupancy into account in both our policy and reward function. The airflow CFM (cubic feet per minute) needed is just multiplied by the number of people in the room. However, at this moment we have no occupancy sensor data, so we assume the most strict condition of full capacity. We standardize our actions in a batch to the range of $[-1, 1]$ as a standard procedure in the RL setup. For each action sample a_i, it is converted to z_i such that $z_i = (a_i - \mu)/s$, where μ is mean, s is the standard deviation of the batch. In the replay buffer, there are several main matrices required: action $\mathcal{A}$, state $\mathcal{S}$, next state $\mathcal{S}'$, reward $\mathcal{R}$ (calculated with our thermal comfort prediction model, power consumption, and RL states), index $\mathcal{I}$ (which records the indices as time stamps), and episode terminal status $\mathcal{N}$ (it labels if an episode is terminated or not—in our setting when the predicted thermal comfort metric does not satisfy the criteria, i.e. $|PMV| > 0.5$, the episode is considered as terminated). To summarize, the batch data is a set consisting of the above-mentioned matrices, i.e. $\mathcal{B} = \{\mathcal{A}, \mathcal{S}, \mathcal{S}', \mathcal{R}, \mathcal{I}, \mathcal{N}\}$.

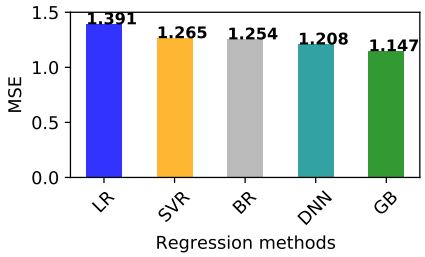

Figure 2: Performance comparison of regression models for predicting thermal comfort based on PMV

We use Intel Xeon Gold 6230 CPUs (2.10GHz) and NVidia Quadro RTX 8000 GPUs with Ubuntu 18.04 OS for our experiments.

4.2 Thermal Comfort Prediction

We compare five different regression models for predicting thermal comfort, namely Linear Regression (LR), Support Vector Regression (SVR), Bayesian Regression (BR), Deep Neural Network (DNN), and Gradient Boosting (GB) (Fig. 2). The input features of the models are

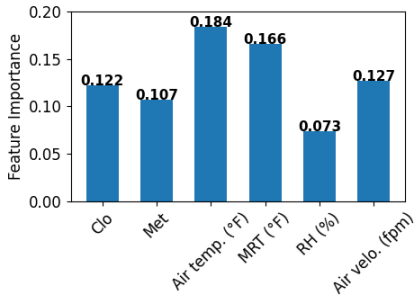

Figure 3: Importance of feature to thermal comfort via mutual information regression analysis. The features are clothing level (Clo), metabolic rate (Met) indoor air temperature (Air temp.), mean radiant temperature (MRT), relative humidity (RH), and air velocity (Air velo.).

zone air temperature, humidity, mean radiant temperature (MRT), air velocity, metabolic rate (Met), and clothing insulation (Clo). We set clothing level as "typical summer indoor clothing". Metabolic rate is set as "typing" where the zones evaluated are all student lab and office spaces. There are in total 30650 data points with complete feature information in the ASHRAE RP-884 thermal comfort data set [13] we adopted for evaluation. All models are trained and tested with 10-fold cross-validation. Hyperparameters optimization is conducted via either grid search or Bayesian optimization. According to Fig. 2, the best model is the gradient boosting tree [27] with an MSE of 1.147, which supports our choice of GB-based model to predict thermal comfort index for the RL reward function. It is reasonable that the gradient boosting method outperforms the deep learning counterpart on tabular data because of selection bias and hyperparameter optimization [54] The MSE metrics reported are averaged with 3 runs for each model.

4.3 Importance of Airflow Control

Few prior works quantitatively study the importance of airflow control in maintaining occupants' thermal comfort. Almost all research focuses on temperature and humidity control for occupants' thermal comfort. Here, we empirically analyze how airflow impacts thermal comfort based on the PMV features.

We conduct analysis via mutual information-based regression. Between two random variables (X, Y), the dependency of these two variables, which is a non-negative value, is calculated as:

$$I(X;Y) = \int_y \int_x p_{(X,Y)}(x,y) log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right) dx\, dy,$$

where $p(X, Y)$ is the joint probability density function of X and Y, and p_X, p_Y are the corresponding marginal density functions. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency [47]. Fig. 3 indicates that air velocity is the second most important factor after air temperature and mean radiant temperature (MRT) (here we approximate MRT with air temperature [12]). Thus, by controlling zone air temperature and airflow (air velocity can be converted to airflow rate with room area), we control the two most important features affecting occupants' thermal comfort.

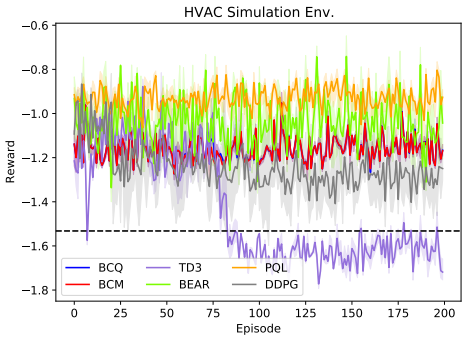

Figure 4: Performance comparison with VAE simulators

4.4 Preliminary Experiments

We first investigate how BRL methods are compared with online RL methods, we compare these BRL methods with the state-of-the-art online RL methods: TD3 [22] and DDPG [35]. Our approach is to build a data-driven simulator environment with two VAEs. The first one is for predicting the RL and thermal comfort states. The second one is to predict the power/energy consumption. These two VAEs function as the thermal states simulator.

We evaluate with 200 episodes and the evaluation frequency is every five time steps. We run each algorithm with three randomly initialized initial conditions in the range of our dataset. As we see in Fig. 4, the solid line is the average of these three runs, and the half-transparent regions indicate the range of these runs. The results show that the performance ranking among these BRL methods: PQL>BEAR>BCQ/BCM>DDPG>TD3. While BRL methods reach a stable state, online RL methods TD3 and DDPG are still exploring new policies, and thus yield a continuously declining performance in a short period of time. These BRL methods (details of PQL and BEAR are elaborated in 4.5) learn exclusively from the batch provide stable, and safe policies. The reason why performance is constant is that in the simulation environment the responses of the system are deterministic which is different from the real building environments. (Fig. 5) In real building systems, the responses are stochastic.

4.5 Baseline Methods

4.5.1 State-of-the-art BRL Methods. After BCQ was proposed, there are several studies outperforming it in the OpenAI Gym [6] simulation environments. We implement these methods as baselines to be compared with BCM.

- Bootstrapping Error Accumulation Reduction (BEAR) [32]: BEAR identifies bootstrapping error as a key source of BRL instability. It is due to bootstrapping of actions that lie outside of the training data distribution. The algorithm mitigates the out-of-distribution action selection by searching over the set of policies that is akin to the behavior policy. BEAR's ultimate goal is to search over the set of policies Π, which shares the same set of values that the random variable can take on as the behavior policy. And its performance is outstanding with the medium-quality static dataset (medium-quality means by training an agent with half amount of time steps cf. expert RL agent/human expert or when the agent is trained to yield half the average return cf. the expert agent).

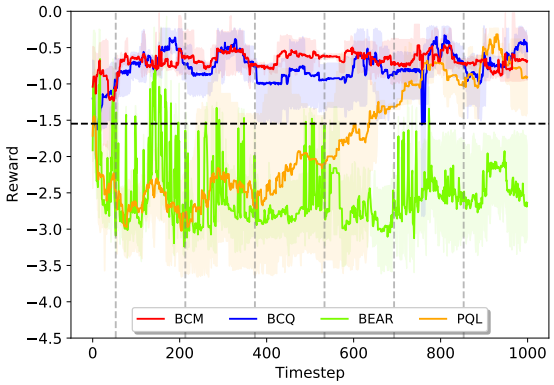

Figure 5: Reward comparison of various algorithms

- Pessimistic Q-Learning (PQL) [36]: While BRL yields a new policy other than those in the batch, it might visit states and actions that are outside the distribution of the batch data. In addition, function approximation with a limited number of samples leads to overly optimistic estimates. PQL thus uses pessimistic value estimates in the low-data regions in the Bellman optimality equation as well as the evaluation back-up. It can yield more adaptive and stronger guarantees when the concentrability assumption does not hold. PQL learns from policies that satisfy a bounded density ratio assumption akin to the on-policy policy gradient methods. The approach of PQL to improve from BCQ's architecture is that they add a state-VAE to predict the arriving state given current state-action pair, filtering state-action distribution $\tilde{\mu}(s, a)$ instead of $\tilde{\mu}(s|a)$. The filtration is implemented by setting a hyperparameter b as the 2^{nd} percentile of the state-VAE Evidence Lower Bound (ELBO). If the ELBO is larger than b then Q-update is executed, otherwise, it is not executed.

4.5.2 Comparison Methodology. We run each algorithm in a single room on each floor in the same week so that outside air temperature (OAT) is the same. For instance, in one week we run our BCM in rooms in the same *stack* on different floors, e.g. 2144, 3144, and 4144, and at the same time a different BRL algorithm, e.g. BEAR, is running in rooms in a different adjacent stack, say, 2146, 3146, and 4146. In each room, we run the algorithm for 1,000 time steps, which is about one week. To reduce performance variations, we evaluate each algorithm in three different rooms (one room from each floor: 2F, 3F, and 4F). These rooms have the same functionality (lab or office spaces) and are of roughly the same size and occupancy capacity. The entire evaluation time of all the experiments is from September 28th to October 19th, 2021.

Appendix A.1 lists the hyperparameters for each method.

4.6 Results and Analysis

4.6.1 Reward Comparison. Fig. 5 shows the evaluation results 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. It shows that BCM outperforms other methods by providing a relatively stable learning curve.

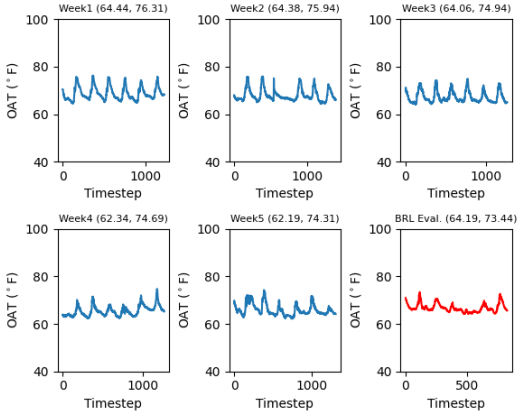

Figure 6: We find the top-5 most similar weeks regarding OAT to our experiment week (last figure) for evaluating energy consumption and thermal comfort.

PQL constrains the Bellman update over state-action pairs that are sufficiently covered by the conditional probability of action given state when generating the data. It adds a state-VAE and a statistical filtration over BCQ's architecture with pessimistic value approximation, which might overkill near-optimal policy that is without enough visitation, however, as time evolves, PQL gradually learns better. BEAR is only guaranteed to outperform BCQ on medium-quality data sets collected from a partially trained policy – a middle ground between optimal policy and random policy. However, in our case, the replay buffers are closer to the data generated with expert policy. This explains the outcomes in a reasonable way. BCQ, as an ablation version of our BCM algorithm, yields a comparable performance as BCM but fails to keep a stable outcome due to the lack of a strong learning signal.

The comparison between algorithms in our experiments is distinct from the results shown in the original papers, where PQL outperforms BCQ and BEAR in two out of the three simulated environments. By contrast, on our real building HVAC system, BCQ provides a more stable and continuously improving performance than the other two BRL methods. This is because all those experiments were conducted in simulation environments where data are effectively unlimited, consequences for poor actions are non-existent, and system dynamics are clean and often deterministic [18]. However, in real-world problems, systems are stochastic and non-stationary. It is not guaranteed that these algorithms would behave the same or similar to simulated cases in these settings.

4.6.2 Energy Consumption and Thermal Comfort Comparison. Outside Air Temperature (OAT) is a key factor affecting zone temperature; therefore, it affects both thermal comfort and energy usage of the HVAC system. It is thus reasonable to compare energy consumption during baseline time periods with the most similar OAT trend to the period during which these BRL methods are evaluated. To do so, we adopt Dynamic Time Warping (DTW) [5] to find historical weeks with similar OATs, as DTW is a widely used method to measure the similarity of time-series data of different lengths. In addition to considering the "shape" of historical OAT, we also consider the mean OAT difference between our experiment time period and historical weeks. In summary, we find historical time periods whose OAT trend is similar *and* with close average weekly

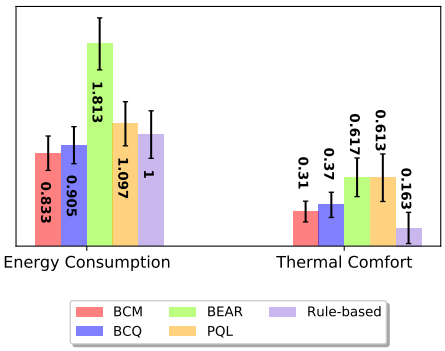

Figure 7: Energy consumption and thermal comfort comparisons among different control methods

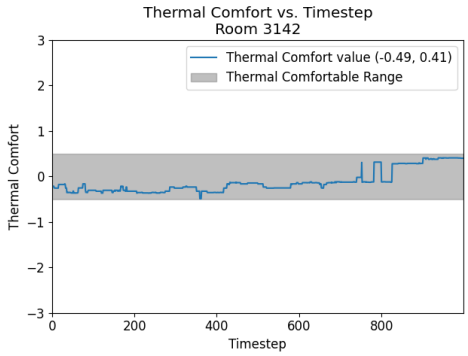

Figure 8: Thermal comfort achieved by our BCM model during evaluation

OAT to our experiment week. Fig. 6 shows an example of historical weeks found using the above metrics. In this figure, a tuple of (*min*, *max*) OAT is labeled on top of each week's OAT data.

Once we have the top-5 weeks with the most similar OAT trend to our experiment period, we compare all methods and estimate energy consumption and thermal comfort.

In Fig. 7, we normalize the historical energy use to one as reference. BCM consumes the least energy compared with other methods. A 16.7% of energy consumption reduction is achieved, and BCQ also outperforms RBC by 9.5%. On the other hand, the occupants' thermal comforts are shown in real average absolute values. The standard deviations (marked as error bars) of all BRL methods are smaller than their historical counterparts.

We also examine the thermal comfort during the entire time period for every experiment and keep track of changes and violations as time evolves. Fig. 8 is an example showing that BCM maintains thermal comfort level persistently during the entire evaluation time period.

4.7 Sensitivity Analysis

4.7.1 Perturbation to Action. In our main evaluation, we used $\Phi = 0.05$, which is the parameter controlling the degree of perturbation applied to selected actions. To inspect how perturbation impacts the performance of BCM, we evaluate two different values of 0.1 and 0.2 for Φ. The result in Fig. 9 indicates that for $\Phi = 0.1$, on average, does not yield a higher reward than $\Phi = 0.05$. For $\Phi = 0.2$,

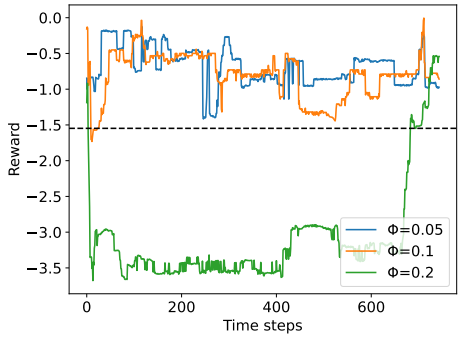

Figure 9: Effect of perturbation to selected actions

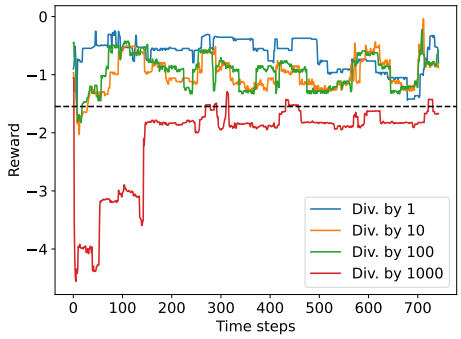

Figure 10: Effect of buffer data size

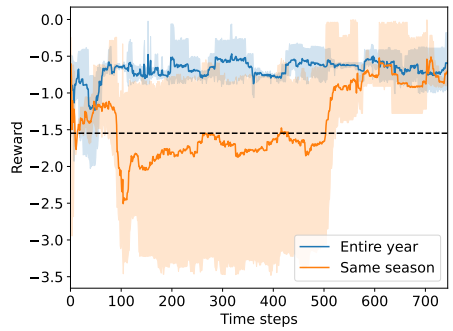

Figure 11: Same Season vs. Entire Year

it cannot learn efficiently until around 700 time steps due to too large the range of action spaces to select from. In our buffer, there is enough diversity since it is extracted with an entire year of data. Thus, we choose $\Phi = 0.05$ in our main experiment.

4.7.2 Amount of Data. We randomly sample data points by a fraction of $\{\frac{1}{10}, \frac{1}{100}, \frac{1}{1000}\}$ and evaluate rooms on the same floor in the same week to observe the impact. Fig. 10 shows the information loss from smaller buffer data. E.g. for the $\frac{1}{1000}$ one, it hardly reaches the average of the original buffer. For the $\frac{1}{10}$ and $\frac{1}{100}$ cases, they show comparable performances but have difficulties being consistent.

4.7.3 Diversity of Batch Data. Originally, we use the thermal states of a set of rooms/zones from an entire year as our batch data. Intuitively, a replay buffer containing data from the same season as

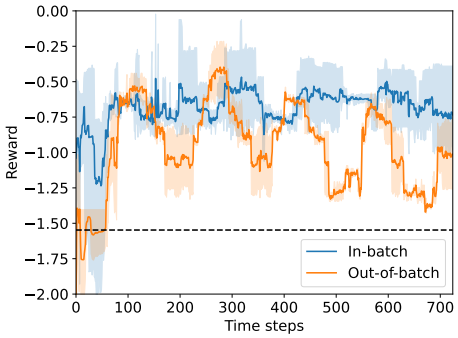

Figure 12: Out-of-batch (OOB) vs In-batch

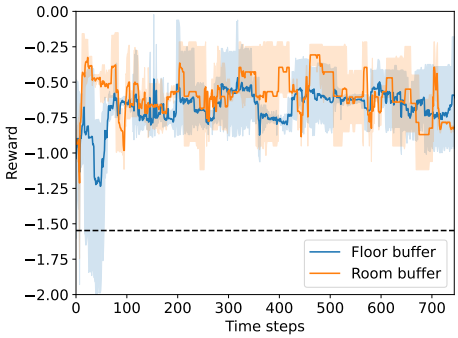

Figure 13: Room Batch vs Floor Batch

our evaluation period might be more suitable because of the similar seasonal weather condition. Thus, we use only the data from the same season as an ablation.

Fig. 11 shows that a batch of the entire year's data produces better performance than only using the same season. A narrower distribution of state-action visitation in a single season cannot update the Q-value as accurately as an entire year's data could. Incorrect Q-value estimation would lead to a lower return. In summary, it is essential to ensure enough state-action visitation diversity in the batch data, in order to estimate the value more accurately.

4.8 Generalization Experiments

4.8.1 In-batch/Out-of-batch Experiment. To examine the generalization of the BRL model, we test the learned policy on rooms where no data exist in the batch. Fig. 12 shows that out-of-batch (OOB) rooms cannot select proper actions to compensate for the OAT fluctuation during the week. The reward curves follow the OAT trend periodically, with clear peaks and valleys. This is reasonable since different zones might respond differently under the same VAV control action, due to the thermal dynamics in the HVAC and distance from VAV to zones.

4.8.2 Room-specific/Floor-specific Experiment. We validate if a room-specific policy is needed. Thus, we use room-specific batch data as our expert policy and evaluate these same rooms. However, in Fig. 13 we observe that although both floor and room models yield consistent outcomes above the average. It is better to use a specific room buffer for a better fit of the room/zone thermal dynamics.

5 CONCLUSION AND FUTURE WORKS

Our simulator-free, multi-zone, BRL-based framework uses existing data as prior knowledge to learn the optimal policy without setting up complex, parameterized simulators. It saves energy compared with the default rule-based control method and maintains thermal comfort. To the best of our knowledge, our work is the first to improve and implement state-of-the-art BRL methods on real building HVAC control. We hope our research encourages domain experts to adopt BRL for real-world problems.

To further improve our control framework, we will update our building operation system to achieve a more frequent data writing rate. This way, we could train the model for the same number of time steps in a shorter time, hereby faster convergence of model. In addition, we will include rooms of different functionality, e.g. conference room, individual office, study area, in our evaluation to create a more generalized model for HVAC control. Also, we could expand the action spaces by including chilling system control and economizers for more comprehensive optimization.

For methodology improvement, we plan to further investigate model-based method in offline mode. Which uses dynamic models to generate a model buffer, then the model buffer is also used to update the BRL model.

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] Anil Aswani, Neal Master, Jay Taneja, David Culler, and Claire Tomlin. 2011. Reducing transient and steady state electricity consumption in HVAC using learning-based model-predictive control. *Proc. IEEE* 100, 1 (2011), 240–253.
[2] 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.
[3] Farinaz Behrooz, Norman Mariun, Mohammad Hamiruce Marhaban, Mohd Amran Mohd Radzi, and Abdul Rahman Ramli. 2018. Review of control techniques for HVAC systems—Nonlinearity approaches based on Fuzzy cognitive maps. *Energies* 11, 3 (2018), 495.
[4] Alex Beltran and Alberto E Cerpa. 2014. Optimal HVAC building control with occupancy prediction. In *BuildSys*. 168–171.
[5] Donald J Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series.. In *KDD workshop*, Vol. 10. Seattle, WA, USA:, 359–370.
[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
[7] Francesco Calvino, Maria La Gennusa, Gianfranco Rizzo, and Gianluca Scaccianoce. 2004. The control of indoor thermal comfort conditions: introducing a fuzzy adaptive controller. *Energy and buildings* 36, 2 (2004), 97–102.
[8] CDC. 2003. Guidelines for Environmental Infection Control in Health-Care Facilities. https://www.cdc.gov/infectioncontrol/guidelines/environmental/background/air.html.
[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 *BuildSys*. 316–325.
[10] 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.
[11] Hui Dai and Bin Zhao. 2020. Association of the infection probability of COVID-19 with ventilation rates in confined spaces. In *Building simulation*, Vol. 13.
[12] Megan Dawe, Paul Raftery, Jonathan Woolley, Stefano Schiavon, and Fred Bauman. 2020. Comparison of mean radiant and air temperatures in mechanically-conditioned commercial buildings from over 200,000 field and laboratory measurements. *Energy and Buildings* 206 (2020), 109582.

[13] Richard J De Dear. 1998. A global database of thermal comfort field experiments. *ASHRAE transactions* 104 (1998), 1141.

[14] Adithya M Devraj, Ana Bušić, and Sean Meyn. 2019. Zap Q-Learning-A User's Guide. In *2019 Fifth Indian Control Conference (ICC)*. IEEE, 10–15.

[15] Xianzhong Ding, Wan Du, and Alberto Cerpa. 2019. OCTOPUS: Deep reinforcement learning for holistic smart building control. In *BuildSys*. 326–335.

[16] Xianzhong Ding, Wan Du, and Alberto E Cerpa. 2020. MB2C: Model-Based Deep Reinforcement Learning for Multi-zone Building Control. In *BuildSys*. 50–59.

[17] Anastasios I Dounis, M Bruant, M Santamouris, G Guarracino, and P Michel. 1996. Comparison of conventional and fuzzy control of indoor air quality in buildings. *Journal of Intelligent & Fuzzy Systems* 4, 2 (1996), 131–140.

[18] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. 2019. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901* (2019).

[19] Povl O Fanger et al. 1970. Thermal comfort. Analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering*. (1970).

[20] 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).

[21] Xiaohan Fu, Jason Koh, Francesco Fraternali, Dezhi Hong, and Rajesh Gupta. 2020. Zonal Air Handling in Commercial Buildings. In *BuildSys*. 302–303.

[22] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *ICML*. PMLR, 1587–1596.

[23] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *ICML*. PMLR, 2052–2062.

[24] 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.

[25] Mengjie Han, Ross May, Xingxing Zhang, Xinru Wang, Song Pan, Da Yan, Yuan Jin, and Liguo Xu. 2019. A review of reinforcement learning methodologies for controlling occupant comfort in buildings. *Sustainable Cities and Society* 51 (2019), 101748.

[26] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *AAAI*.

[27] 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.

[28] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[29] SA Klein. 1976. University of Wisconsin-Madison Solar Energy Laboratory. *TRNSYS: A transient simulation program. Eng. Experiment Station* (1976).

[30] Andrew Krioukov, Gabe Fierro, Nikita Kitaev, and David Culler. 2012. Building application stack (BAS). In *BuildSys*. 72–79.

[31] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[32] 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).

[33] Geoff J Levermore. 1992. Building energy management systems. (1992).

[34] Yuguo Li, Hua Qian, Jian Hang, Xuguang Chen, Ling Hong, Peng Liang, Jiansen Li, Shenglan Xiao, Jianjian Wei, Li Liu, et al. 2020. Evidence for probable aerosol transmission of SARS-CoV-2 in a poorly ventilated restaurant. *MedRxiv* (2020).

[35] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[36] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. 2020. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202* (2020).

[37] Siliang Lu, Weilong Wang, Chaochao Lin, and Erica Cochran Hameen. 2019. Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884. *Building and Environment* 156 (2019), 137–146.

[38] 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.

[39] 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.

[40] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*. PMLR, 1928–1937.

[41] Srinarayana Nagarathinam, Vishnu Menon, Arunchandar Vasan, and Anand Sivasubramaniam. 2020. MARCO-Multi-Agent Reinforcement learning based COntrol of building HVAC systems. In *e-Energy*. 57–67.

[42] June Young Park and Zoltan Nagy. 2020. HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points. In *e-Energy*. 434–437.

[43] Samuel Prívara, Zdeněk Váňa, Dimitrios Gyalistras, Jiří Cigler, Carina Sagerschnig, Manfred Morari, and Lukáš Ferkl. 2011. Modeling and identification of a large multi-zone office building. In *2011 IEEE International Conference on Control Applications (CCA)*. IEEE, 55–60.

[44] 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.

[45] REHVA. 2021. REHVA COVID19 Guidance v4.1. https://www.rehva.eu/fileadmin/user_upload/REHVA_COVID-19_guidance_document_V4.1_15042021.pdf.

[46] Martin Riedmiller. 2005. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *ECML*. Springer, 317–328.

[47] Brian C Ross. 2014. Mutual information between discrete and continuous data sets. *PloS one* 9, 2 (2014), e87357.

[48] 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.

[49] 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.

[50] 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.

[51] Jyri Salpakari and Peter Lund. 2016. Optimal and rule-based control strategies for energy flexibility in buildings with PV. *Applied Energy* 161 (2016), 425–436.

[52] Caude Sammut. 2010. *Behavioral Cloning*. Springer US, 93–97.

[53] AB Shepherd and WJ Batty. 2003. Fuzzy control strategies to provide cost and energy efficient high quality indoor environments in buildings with high occupant densities. *Building Services Engineering Research and Technology* 24, 1 (2003).

[54] Ravid Shwartz-Ziv and Amitai Armon. 2021. Tabular Data: Deep Learning is Not All You Need. *arXiv preprint arXiv:2106.03253* (2021).

[55] Muthusamy V Swami and Subrato Chandra. 1987. Procedures for calculating natural ventilation airflow rates in buildings. *ASHRAE final report FSEC-CR-163-86, ASHRAE research project* (1987), 130.

[56] IEA UN. 2020. Global status report for buildings and construction (2019). *Available at https://www. gbpn. org/china/newsroom/2019-global-status-report-buildings-and-constr uction. Access date* 15 (2020).

[57] William Valladares, Marco Galindo, Jorge Gutiérrez, Wu-Chieh Wu, Kuo-Kai Liao, Jen-Chung Liao, Kuang-Chin Lu, and Chi-Chuan Wang. 2019. Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm. *Building and Environment* 155 (2019), 105–117.

[58] 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.

[59] José Vázquez-Canteli, Stepan Ulyanin, Jérôme Kämpf, and Zoltán Nagy. 2018. Adaptive multi-agent control of HVAC systems for residential demand response using batch reinforcement learning. (2018).

[60] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. 2020. Leverage the average: an analysis of KL regularization in reinforcement learning. In *NeurIPS*.

[61] Nino Vieillard, Olivier Pietquin, and Matthieu Geist. 2020. Munchausen reinforcement learning. *arXiv preprint arXiv:2007.14430* (2020).

[62] Zhe Wang and Tianzhen Hong. 2020. Reinforcement learning for building controls: The opportunities and challenges. *Applied Energy* 269 (2020), 115036.

[63] 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.

[64] Thomas Weng, Anthony Nwokafor, and Yuvraj Agarwal. 2013. Buildingdepot 2.0: An integrated management system for building analysis and control. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. 1–8.

[65] Daniel A Winkler, Ashish Yadav, Claudia Chitu, and Alberto E Cerpa. 2020. Office: Optimization framework for improved comfort & efficiency. In *IPSN*. IEEE, 265–276.

[66] 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.

[67] Liang Yu, Shuqi Qin, Meng Zhang, Chao Shen, Tao Jiang, and Xiaohong Guan. 2020. Deep Reinforcement Learning for Smart Building Energy Management: A Survey. *arXiv preprint arXiv:2008.05074* (2020).

[68] 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.

[69] Tianyu Zhang, Gaby Baasch, Omid Ardakanian, and Ralph Evins. 2021. On the Joint Control of Multiple Building Systems with Reinforcement Learning. (2021).

[70] Zhiang Zhang and Khee Poh Lam. 2018. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *BuildSys*.

A APPENDIX

A.1 Experiments Details

A.1.1 Parameters. For researchers to better reproduce our results, we provide the hyperparameters used in our experiments. For most of the models, we follow their default settings unless otherwise recommended. We do not fine-tune the hyperparameters of the BRL algorithms and use the reported values in the literature [23, 32, 36], and we keep the architecture of actor-critic networks for a fair comparison. Modifying the architecture or any detail of implementation might lead to a large difference in performances. [26] In PQL, we scale the maximum state VAE training steps according to the ratio of PQL's MuJoCo buffer size to our building buffer size. For all the network architectures we follow the original setups. The details of the hyperparameters are listed in Table 1.

Table 1: Hyperparameter Settings of evaluated methods

	BCM	BCQ	BEAR	PQL
γ	0.99	0.99	0.99	0.99
N	100	100	100	100
τ	0.005	0.005	0.005	0.005
λ	0.75	0.75	0.75	0.75
Φ	0.05	0.05	–	0.1
α_m	0.9	–	–	–
τ_m	0.03	–	–	–
clip value min.	-1	–	–	–
backup	–	–	–	Q-max
QL noise	–	–	–	0.15
b percentile	–	–	–	2
max state VAE trainstep	–	–	–	$2e4$
Policy update version	–	–	0	–
MMD matching # samples	–	–	5	–
MMD sigma	–	–	20	–
Kernel type	–	–	Laplacian	–
Lagrange threshold	–	–	10	–
Distance type	–	–	MMD	–

γ: discount factor, N: mini-batch size, τ: target network update rate, λ: minimum weighting between two Q-networks, Φ: max perturbation on action, α_m: Munchausen scaling term, τ_m: entropy temperature, clip value min.: minimum clipping value on Munchausen term

A.1.2 Data Monitored. In the evaluation processes, we monitored all the states as time series to observe if there is any abnormality. Also, to inspect how BRL methods optimize the target objectives.

As shown in Fig. 14, it is an example of how the thermal comforts of historical weeks vary under rule-based control. Apparent periodic patterns are observed which follow the OAT trends during the week. It indicates that RBC cannot compensate the OAT variations as BRL method (Fig. 8).

In Fig. 15, it shows the time series of the states observed during BRL evaluation. Our BRL method BCM keep zone air temperature setpoint (ZNT StPt) in a narrow range stably, thus, keep the zone air temperature readings (ZNT) in a reasonable range to maintain thermal comforts while no constraints are applied, which is different from online RL methods where the range of actions are constrained by human experts as hard rules.

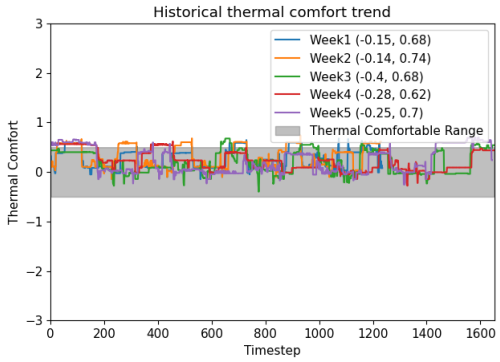

Figure 14: An example of historical thermal comfort trends in top-5 similar OAT weeks

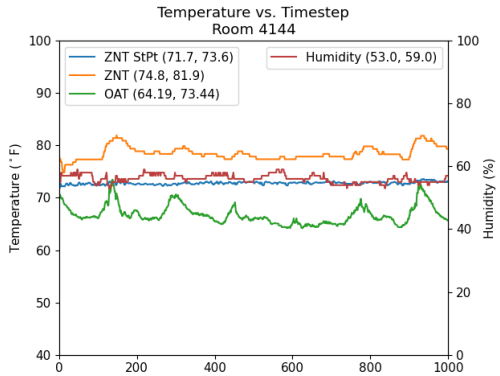

Figure 15: States in BCM evaluation week

A.2 Experiment with safe minimum airflow

A.2.1 Motivation. Indoor environment and indoor gatherings present a disease spreading risk as virus-laden aerosol lingers in indoor air for hours at high concentrations [34] rather than being quickly dispersed and destroyed through UV (sun)light outdoors. Accumulated exposure to viral load over time is an important risk determinant for an individual to be infected [11]. In the context of the current pandemic caused by the spreading of the SARS-CoV-2 virus that causes COVID-19 disease, many efforts are underway to control its spread for the public healthcare system to maintain its capacity and reduce fatalities. We believe that a well-designed operation of the HVAC system can be a critical means to reduce the likelihood of spreading events by appropriately directing airflows. HVAC societies such as ASHRAE and REHVA have recommended high rates of air circulation and an increased fraction of fresh air. This is typically measured by air changes per hour, or ACH, in a given enclosed space or the entire building. ACH is computed by the air volume added to or removed from space in an hour divided by the total volume of the space [55]. For air impurities removed by fresh air, unit ACH is then a time constant that represents the rate of dilution in infectious particles caused by the introduction of fresh-air [11]. ACH is increased primarily by increasing the ratio of fresh air and the speed of airflow supplied to a given space. Typically, commercial buildings are designed to achieve ACH levels of 3-5 whereas more sensitive areas in hospital settings could be as high as 12 ACH [8]. Achieving a substantially high ACH level in a typical office building

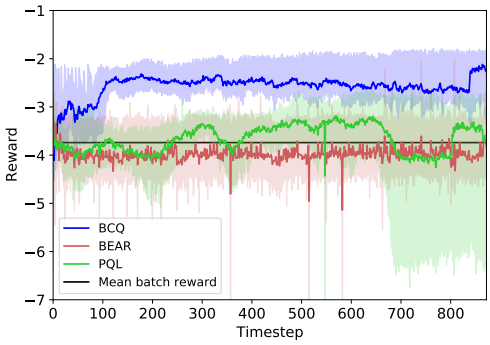

Figure 16: Reward comparison (considering safe airflow)

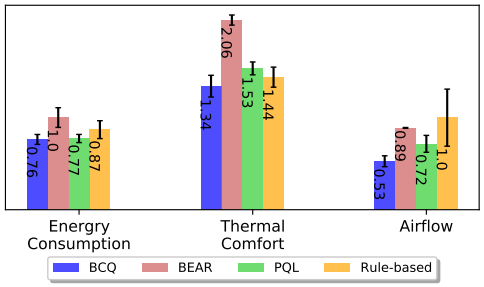

Figure 17: Energy, thermal comfort, and airflow comparison

is challenging due to the cooling capacity of the equipment [21], and thus in our study, we seek to fulfill a minimum safe airflow requirement.

A.2.2 Safe Airflow Level Guidelines. Various guidelines have been issued by ASHRAE[2], CDC[3], and the European union REHVA[4] on building operation to lower the risk of getting infected by the respiratory disease of the occupants through the air during the COVID-19 pandemic. These guidelines provide detailed recommendations regarding multiple aspects of building operation and share much in common, including, but not limited to, use of high-rating minimum efficiency reporting value (MERV) filters and/or UV-C lighting to treat the return air, 24/7 HVAC operation, no use of recirculated air (i.e. use 100% outside air), increased air change (ACH) rate during occupancy.

While comprehensive, these recommendations are difficult to implement altogether, if not completely impossible. The effects of these measures and their implications on the building systems with respect to energy consumption and occupants' thermal comfort still largely remain unclear to practitioners and residents. In our work, we maintain a safe airflow level in the zones we evaluate by requiring a minimum of 21.19 CFM per person (10L/s per person) [45] airflow in a space, which satisfies ASHRAE's, REHVA's, and CDC's requirements.

A.2.3 Experiment Results. In Fig. 16, we compare several state-of-the-art BRL methods as we did in our main experiments. The minimum safe airflow is calculated with the people occupied in the room, where we assume full occupancy.

[2] https://tinyurl.com/yy8f5faq
[3] https://tinyurl.com/y9lczbwp
[4] https://tinyurl.com/yy8nzlmj

The state, action, environment setups are all the same as our main experiments. Except for the reward function at time step t is calculated with the following equation:

$$R_t = -\alpha ReLU(|TC_t| - TC_c) - \beta s_t^{Sup} - \delta ReLU(A_{min}^{safe} - s_t^{Sup}), \quad (4)$$

In Eq.(4), α, β, δ are the weights balancing between different objectives and could be tuned to meet specific goals, TC_t is the thermal comfort index at time t, TC_c is the requirement on thermal comfort, 0.5, and s_t^{Sup} is the supply airflow at the time t, and we assume each room is fully occupied, leading to a constant A_{min}^{safe} for each room based on the ACH requirement and number of people at full occupancy. The *ReLU* (Rectified Linear Unit) activation function is used here to penalize any thermal comfort index that is out of the comfortable range and any airflow value that is lower than minimum safe airflow.

The results are run with two stacks of rooms per algorithm. And each stack of runs lasts approximately a week. The experiment result motivates us to improve from BCQ, since it outperforms the others in the real HVAC environments. The buffer is the same as our main experiments with an entire year of records. And the evaluation time period is from June 1st to June 14th, 2021.

To further analyze the improvements of the target objectives, respectively. Fig. 17 shows the comparison of energy consumption, thermal comfort, and airflow readings. In this figure, RBC value of each category is normalized as one. We could observe that in summer OAT weeks, BRL methods could save more energy compared with the results of our main experiments where evaluation is done in the Fall. BCQ is with a 24 percent of energy reduction cf. RBC due to a more efficient policy control with a more stable airflow and thermal comfort, as the error bars shown in the figure.