Energy & Buildings 285 (2023) 112878

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

Mitigating an adoption barrier of reinforcement learning-based control strategies in buildings

Aakash Krishna G.S. [a,1], Tianyu Zhang [a,1], Omid Ardakanian [a,*], Matthew E. Taylor [a,b]

[a] Department of Computing Science, University of Alberta, Edmonton T6G 2E8, Canada
[b] Alberta Machine Intelligence Institute (Amii), Edmonton T5J 3B1, Canada

ARTICLE INFO

Article history:
Received 25 November 2022
Revised 16 January 2023
Accepted 6 February 2023
Available online 11 February 2023

Keywords:
Multi-agent reinforcement learning
Population diversity
Off-policy policy evaluation
Sim-to-real transfer

ABSTRACT

Reinforcement learning (RL) algorithms have shown great promise in controlling building systems to minimize energy use, operational cost, and occupant discomfort. RL agents learn a control policy by interacting with the physical or simulated environment that represents building systems, occupants, and the outside world. Yet, a large amount of data is needed to learn a near-optimal control policy in the physical building, which requires months or years to collect. Moreover, an agent's performance while training can be quite poor, causing occupant discomfort and additional costs. Learning in simulation does not have such real-world impacts, but differences between buildings, and indeed between simulation and physical buildings, potentially lead to poor performance when a policy learned in simulation is deployed in a physical building. This paper addresses part of the sim-to-real problem by training on one set of simulated (source) buildings and then deploying to a novel simulated (target) building. This approach significantly reduces the training cost of RL on the target building by 1) learning a large number of policies on prototype buildings, 2) evaluating these policies on historical data obtained from the target building's environment and selecting the best ones according to the evaluation result, and 3) using the best policies to control the target building while continuing to learn. The proposed approach involves learning a diverse population of control policies using a novel diversity-induced RL algorithm, and policy clustering, evaluation, and selection techniques. Three case studies show our approach assigns policies to the target building that outperform the default controller by 4.0–30.4%, without sacrificing thermal comfort. Similarly, they outperform policies that are learned only on the target building (i.e., without transfer) by 24.9–74.9% and 16.2–72.2% before and after 500 months of training, respectively.

1. Introduction

Safe and optimal control of buildings has received increasing attention in the past few decades due to two main reasons. First, building operation and construction are responsible for almost one-third of total global final energy consumption and 15% of direct carbon emissions [1]. Second, people spend around 90% of their time indoors [2], hence their health and well-being largely depend on the operation and Indoor Environmental Quality (IEQ) performance of buildings. Despite the growing adoption of sensors and monitoring systems in buildings, controls are still simple, reactive, and must be customized for every building based on its type, floor plan, and occupancy schedule. Existing building controls — in particular Heating, Ventilation and Air Conditioning (HVAC) control systems — do not take full advantage of time-series emitted by the sensors and fail to strike a balance between energy use and IEQ factors, such as thermal comfort [3]. As a result, buildings are not as comfortable as they could be, consume a significant amount of energy, and produce excessive carbon emissions.

Many efforts have been made to date to make building controls proactive and adaptive to occupant and grid needs. Specifically, a wide range of model-based, data-driven, and learning-based control strategies have been proposed in the literature [4,5], yet none of these strategies can be applied to all types of buildings in the building stock. Model Predictive Controls (MPC) are not widely adopted because accurate weather and thermal models, and occupancy schedules are not readily available for many buildings. Although these models may be available for select buildings, they cannot be used to control other buildings due to the diversity in building design and construction. Identifying these models using data collected from the building and its environment has its own challenges (e.g., lack of sufficient excitation [6]), and does not work

[*] Corresponding author.

E-mail addresses: krishnag@ualberta.ca (A. Krishna G.S.), tzhang6@ualberta.ca (T. Zhang), ardakanian@ualberta.ca (O. Ardakanian), matthew.e.taylor@ualberta.ca (M.E. Taylor).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.enbuild.2023.112878
0378-7788/© 2023 Elsevier B.V. All rights reserved.

for newly constructed buildings where trend or log data is scant. Learning-based controls, such as policies learned by interacting with the building systems via a Reinforcement Learning (RL) algorithm, require less customization in general, but they perform poorly in the early stage of training when the decision-making agent is exploring a large state-action space. Previous work shows that learning a reasonable, near-optimal policy for a complex multi-zone building may take several weeks or even months [7,8]. While the policy can be learned by interacting with a simulated building (e.g., the digital twin of the building) to reduce the high-qualitytraining cost, it requires prior knowledge of the building model, suffering from the same scalability issue as MPC. If the policy is learned in the physical building, its sub-optimal operation in the early stage of training can be overly costly and its low IEQ performance may cause excessive discomfort for the occupants. This is one of the primary adoption barriers of RL-based controls.

In this paper, we investigate how to mitigate this key adoption barrier. The first step is to build a library of control policies, each trained on a real or simulated building, using *environment* and *policy diversity* [9]. We observe that some policies in the policy library when transferred to a novel environment, i.e., a thermal zone in the target building, perform markedly better than policies that have been trained on that building for several months. To efficiently identify these policies, we borrow ideas from Neural Architecture Search (NAS) and Off-Policy Policy Evaluation (OPE) to evaluate the policies in the policy library using a small batch of log data (e.g., just a few weeks worth of data) from the target building. The log data is collected while the target building is controlled by a default controller, e.g., a rule-based or reactive controller. We show that the proposed policy evaluation approach gives us a reliable estimate of how these policies might perform on the target building, thereby enabling us to assign a subset of them to zones in that building. Our approach entails policy clustering, selective ranking, and eventually transferring the best policies to respective zones in the target building. Our contribution is threefold:

- We build a library of RL-based HVAC control policies using a diversity-induced policy gradient method with an augmented loss function. These policies are learned through interaction with a prototype office building.
- We propose a novel two-step method that combines policy clustering and evaluation, and uses the resulting ranking to efficiently identify high-quality policies for the target building among policies in the policy library. This method requires just two weeks of log data collected from the target building.
- Using three buildings in different climates, we evaluate the efficacy of the proposed policy selection and evaluation technique in identifying high-quality policies that, when transferred to a novel building, outperform the default rule-based or reactive controller, even without retraining.

The rest of this paper is organized as follows. We introduce the Markov decision process, a sample-efficient policy gradient RL algorithm, and two kinds of diversity in Section 2, and discuss different approaches to policy evaluation in Section 3. We formulate the control problem in Section 4 and present our methodology in Section 5. The source and target buildings are described in Section 6 and our experiment results are discussed in Section 7. We review the related work on HVAC control including RL-based control techniques in Section 9, and conclude the paper in Section 10.

2. Reinforcement learning

Reinforcement learning (RL) [10] is a learning framework that allows agents to optimize their behavior in an environment through trial and error. It is formulated in terms of a Markov decision process (MDP) which is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ represents the set of states the agent could be in, $\mathcal{A}$ represents the set of all actions the agent can take in the environment, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the stochastic transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discount rate. The goal of an agent is to learn a (stochastic) policy π, which maps the state to a probability distribution over actions, helping the agent choose the action that maximizes its *return*. The return is defined as the discounted reward of the agent $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$.

Proximal Policy Optimization (PPO). PPO is a model-free policy gradient RL algorithm [11]. It is based on the actor-critic architecture which is suitable for continuous control problems. PPO is shown to be effective in numerous RL tasks [12]. This is partly due to the fact that it employs a clipping mechanism that restricts the updates of the policy to a trust region, and that it performs multiple epochs of updates per each sample. The loss of PPO is given by:

$$L_{PPO} \doteq \widehat{\mathbb{E}}\left[\min\left(\rho(s_t, a_t)\widehat{A}_t, \text{clip}(\rho(s_t, a_t), 1-\epsilon, 1+\epsilon)\widehat{A}_t\right)\right], \tag{1}$$

where $\widehat{A}_t$ is the estimated advantage at time t, $\rho(s_t, a_t)$ is the ratio of the probability under the new and old policies respectively, and *clip* projects this probability ratio onto $[1-\epsilon, 1+\epsilon]$ so it cannot be too far away from 1. The advantage estimate can be calculated from $\widehat{A}_t \doteq G\prime_t^{\pi_{\text{old}}} - V^{\pi_{\text{old}}}(s_t)$, where $G\prime_t^{\pi_{\text{old}}}$ is the discounted reward starting from s_t and running π_{old} for a fixed number of timesteps, and $V^{\pi_{\text{old}}}$ is the state-value function under π_{old}. The ratio $\rho(s, a)$ is defined as $\frac{\pi(a_t|s_t)}{\pi_{old}(a_t|s_t)}$ with π being the learning policy and π_{old} representing the old policy. Finally, ϵ is the hyperparameter controlling the size of the updates by constraining them to a trust region as stated earlier.

Diversity-Induced Reinforcement Learning. The optimal policy learned through a series of interactions with a given environment may perform poorly in a novel environment. To improve generalization, diversity-induced RL algorithms attempt to find a large number of near-optimal, yet diverse policies on the training environment [13]. To enforce diversity, one approach is to modify the reward function such that diversity is encouraged during training [14]. This is referred to as *policy diversity*. Another approach is to train a policy on various environments [15], which is known as *environment diversity*.

3. Policy evaluation methods

The previous section discussed how reinforcement learning can be used to train a policy by interacting with the environment. This section discusses how a policy could be evaluated from environmental data captured by another policy. For example, a learned policy could be evaluated from data collected while a rule-based controler was interacting with an environment.

3.1. Off-policy policy evaluation

Off-policy Policy Evaluation (OPE) concerns estimating the performance of a given decision-making policy, known as the *evaluation policy*, using historical data that may have been generated by a different *behavior policy* (e.g., a proportional controller). We denote the historical data as $\mathcal{D} = \left\{(s_t, a_t, r_t)_{t=1}^{n}\right\}$ where s_t, a_t, and r_t are respectively the state, action taken, and reward received from the environment at t. The most popular OPE methods are based on importance sampling, examples of which are inverse probability weighting (IPW) [16] and self-normalized inverse probability weighting (SNIPW) [17]. In general, SNIPW is shown to be more stable in certain tasks as its value is bounded by the support of the rewards and its variance is smaller than IPW [18]. Given the evaluation policy π_e and the behavior policy π_b that was used to generate the historical data, the value of π_e (i.e., the expected cumulative reward available from each state–action pair) under IPW and SNIPW is defined as follows:

$$\widehat{V}_{\text{IPW}}(\pi_e;\mathcal{D}) \doteq \frac{1}{n}\sum_{t=1}^{n}\rho(s_t,a_t)r_t,$$

$$\widehat{V}_{\text{SNIPW}}(\pi_e;\mathcal{D}) \doteq \frac{\sum_{t=1}^{n}\rho(s_t,a_t)r_t}{\sum_{t=1}^{n}\rho(s_t,a_t)},$$

where $\rho(s,a) \doteq \frac{\pi_e(a|s)}{\pi_b(a|s)}$ is the importance sample ratio, $\mathcal{D}$ denotes the offline dataset from which the trajectory was sampled, and s_t, a_t and r_t respectively represent the state, action taken, and reward received at time step t. The above-mentioned OPE methods assume that actions are discrete, and use rejection sampling to filter the dataset. However, this approach cannot be extended to work with continuous actions as rejection sampling does not work in the continuous setting [19]. To overcome this limitation, Kallus et al. [19] employ kernel density estimation to calculate the value of a policy, which is given by:

$$\widehat{V}_{\text{Kernel}}(\pi_e;\mathcal{D}) \doteq \mathbb{E}\left[\frac{1}{h}K\left(\frac{\text{argmax}_{a_{t'}}\pi_e(a_{t'}|s_t) - a_t}{h}\right)\frac{r_t}{\pi_b(a_t|s_t)}\right].$$

Here K is the kernel function, such as the Gaussian kernel, and h is the bandwidth which is a hyperparameter. When a Gaussian kernel is adopted, we refer to this method as GK.

In this paper we investigate the use of three OPE methods, namely IPW, SNIPW, and GK, in the building control domain to evaluate policies in the policy library using log data generated by a default controller in the target building. To our knowledge, this is the first time that the OPE methods are used for policy evaluation and transfer learning in buildings.

3.2. Proxy-based policy evaluation

Neural Architecture Search (NAS) has become the standard technique in deep learning to discover the best neural networks among a set of candidate architectures for a given supervised learning task. Since the search space of neural architectures can be extremely large, one cannot possibly evaluate the performance of all architectures. Thus, it is important to devise efficient exploration and lightweight evaluation techniques [20]. Efforts have been made to identify low-cost or zero-cost proxy (ZCP) [21,22] tasks to rank neural networks at initialization (i.e., before training). In gradient-based approaches, a mini-batch of data is used to calculate the gradient of loss for each layer. These gradients are then used to rank neural networks. Gradient-based approaches differ in how the gradients are aggregated but all use the aggregate as a heuristic to predict how the neural network would perform in a task.

Lee et al. [22] introduces a saliency metric, called SNIP, that approximates the change in loss when a connection is removed. This helps identify connections in the network that are important to the given task before training the network, using a mini-batch of data. While SNIP was originally proposed for network pruning, it can be used as a proxy for NAS, based on the observation that a neural network that attains a higher SNIP will perform better in the given task [21]. SNIP is defined as

$$\mathcal{S}_{\text{SNIP}} \doteq \left|\frac{\partial\mathcal{L}}{\partial\theta}\odot\theta\right|,$$

where $\mathcal{L}$ is the loss function of the neural network with parameters θ, and $\odot$ denotes the Hadamard product. Abdelfattah et al. [21] empirically evaluate various ZCP metrics to compare their efficiency in ranking neural networks. They also propose a new metric, called *gradnorm* (GN), which can be used for NAS, and is defined as the sum of the Euclidean norm of the gradients after backpropagating the loss computed from a mini-batch of data.

4. Zero-cost proxies for RL

We consider an HVAC system that consists of one or multiple air handling units (AHUs) and variable air volume (VAV) systems as depicted in Fig. 1. The optimal HVAC control can be cast as a sequential decision making problem where an agent interacts with the building to control various knobs (e.g., actuators in the VAV systems) and receives a reward in return, which is used to learn the control policy. While a single agent can control the entire building (all actuators in AHUs and VAV systems), it prevents the policy from being transferred to a new building that has a different state-action space, e.g., contains more VAV systems. We frame the HVAC control problem in a MARL setting where each agent is responsible for controlling a single zone; a building is controlled by several independent agents, each acting in their respective zone. Our multi-agent MDP is a tuple $(N, \mathcal{S}, \mathcal{A}_{i,i\in\{1,\ldots,N\}}, \mathcal{R}_{i,i\in\{1,\ldots,N\}}, \mathcal{T}, \mathcal{H})$ where.

- N represents the number of agents;
- $\mathcal{S}$ represents the state space observed by all the agents. In our formulation, each agent receives readings of 6 physical or virtual sensors, namely mean temperature ($^\circ C$), mean humidity ($\%$), outdoor temperature ($^\circ C$), solar irradiation (W), binary occupancy state, and hour of the day ($0-23$);
- $\mathcal{A}_i$ denotes the action space for agent i. We define the action of each agent as setting the minimum position of the damper in their VAV system. The minimum damper position is a value in $[0.1, 1]$, where 0 indicates that the damper is closed and 1 indicates that the damper is fully opened. For example, if the agent assigns 0.2 to the minimum damper position, the damper can be opened between 20% and 100%. The AHU control points and all other VAV control points are adjusted by the controller in EnergyPlus, using the *predictive system energy balance* method [23]. This controller makes sure the thermal comfort requirement is satisfied in each zone by adjusting the supply air temperature and/or the reheat coil power. The RL agents do not interfere with this process;
- $\mathcal{R}_i$ is the scalar reward received by agent i when it takes action a that causes a state transition from s to $s\prime$. To incentivize the agents to minimize the HVAC energy consumption, we define the reward of each agent as the energy consumption of the respective VAV system with a negative sign;
- $\mathcal{T}$ is the transition function that defines the transition probability from state s to state $s\prime$. EnergyPlus [23] defines this transition function;
- $\mathcal{H}$ denotes the length of each episode. Even though agents continuously control the VAV systems, we model the control problem as an episodic task so that we can evaluate the policies over a fixed period of time. Specifically, we use one winter month with 15-min time steps to create an episode.

Table 1 lists all the state features and the action for each agent. In MARL, each agent aims to learn a policy π_i that maximizes the expected discounted return G_i, given by

$$G_i \doteq \mathbb{E}\left[\sum_{t=0}^{\mathcal{H}}\gamma^t\mathcal{R}_i\left(s_t, \text{argmax}_{a_t\in\mathcal{A}_i}\pi_i(a_t|s_t)\right)\right]. \quad (2)$$

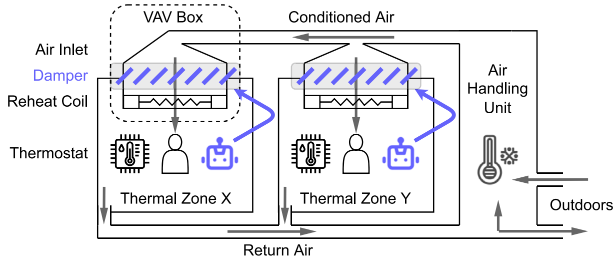


Fig. 1. Illustration of an air loop in a multi-zone building equipped with a forced-air heating and cooling system.

Table 1
State variables and the action of each agent.

State	Zone mean temperature	°C
	Zone mean humidity	%
	Zone occupancy	Binary
	Outdoor temperature	°C
	Solar radiation	W
	Hour of the day	Integer
Action	VAV minimum damper position	%

In our setting, we set $\gamma = 1$ since the length of each episode is finite. Since agents have different rewards, this is a competitive MARL setting. Although this formulation might increase the convergence time, it makes it possible to separately train these agents and transfer a subset of them to another building. We believe this outweighs the drawback of slower convergence.

5. Methodology

We propose a warm-start solution for MARL-based control of the HVAC system that requires only a small amount of historical data collected from the target building. Our methodology has three main parts: 1) *building a library of diverse policies*, 2) *policy selection based on ranking results*, and 3) *policy transfer and retraining*. We first explain how to learn many diverse policies to control the HVAC system of a source building. These policies comprise the *policy library*. We then use a clustering algorithm and various policy evaluation methods to efficiently identify the most suitable policies for controlling the target building using the historical data. After these policies are assigned to the respective zones in the target building to control VAV systems, we retrain them on the target building in an online fashion. The overall workflow is illustrated in Fig. 2.

5.1. Building the policy library

To improve generalization of RL agents that control VAV systems located in individual zones of a building, we build a policy library by taking advantage of both policy diversity and environment diversity, which are defined below. The resulting library includes optimal and near-optimal policies found for a medium office (prototype) building, which is our training environment (aka source building). Since we do not know a priori which of these near-optimal policies could perform better when transferred to a novel target building, we generate a large number of diverse policies to cover a large area of the policy space.

5.1.1. Policy diversity

We augment the loss function of PPO with a diversity loss term, denoted $\mathcal{L}_{diversity}$, as shown below:

$$\mathcal{L}_{augmented} = \mathcal{L}_{PPO} + w\mathcal{L}_{diversity}, \tag{3}$$

where $\mathcal{L}_{PPO}$ is defined in Eq. (1) and w is a hyperparameter that yields a trade-off between reward optimality and policy diversity. When w is zero, we will find the optimal policy, and when it is non-zero, we will find a near-optimal policy that is distinct from the previously learned policies. We define $\mathcal{L}_{diversity}$ as follows:

$$\mathcal{L}_{diversity} = -\frac{\sum_{\pi\prime\in\Pi_{learned}}\sum_{(s,a)\in\text{exp}}\frac{max\left(\frac{\max(\pi(a|s),\pi\prime(a|s))}{\min(\pi(a|s),\pi\prime(a|s))},\bar{\rho}\right)}{\left|G^{\text{exp}}(s)-V^{\pi\prime}(s)\right|}}{\left|\Pi_{learned}\right|}, \tag{4}$$

where π is the behavior policy we are updating, $\bar{\rho}$ is the upper bound on the probability ratio, *exp* is the state–action tuples generated by the behavior policy in the current episode and stored in the replay buffer, $G^{exp}(s)$ is the cumulative reward of this trajectory starting from the state s, $V^{\pi\prime}(s)$ is the estimated state value of state s under a previously learned policy, and $\Pi_{learned}$ is a set of previously learned policies from which the behavior policy should differ. According to these definitions, $|G^{\text{exp}}(s) - V^{\pi\prime}(s)|$ represents the estimation bias of a learned policy given the current trajectory. The estimation bias is high when the previously learned policy $\pi\prime$ disagrees with the experience gained under the behavior policy. The probability ratio, i.e., $\frac{\max(\pi(a|s),\pi\prime(a|s))}{\min(\pi(a|s),\pi\prime(a|s))}$, measures the differences between the behavior policy π and a previously learned policy $\pi\prime$. Thus, $\mathcal{L}_{diversity}$ will be smaller (i.e., the behavior policy is considered more distinct from the learned policies) when the behavior policy estimates the action probabilities differently from the set of policies learned so far and state value estimates are reliable as the learned policies agree with the behavior policy.Note that we encourage policy diversity using a different approach than the population diversity introduced in [14]. The diversity loss term we added to PPO's loss function was originally proposed in our prior work [9]. We slightly change its definition here to ensure that the action probability ratio is bounded at all times. This is achieved by clipping the probability ratio at $\bar{\rho}$.

5.1.2. Environment diversity

Adding small perturbations to the training environment can improve generalization of RL algorithms [15]. We incorporate environment diversity by installing blinds to cover all windows in the training environment. This technically adds a new training environment. Furthermore, we update the occupancy pattern of each zone to remove the time intervals when a zone becomes unoccupied (e.g., lunchtime) during core business hours. This gives two more training environments, bringing the number of training environments to four. Diverse policies are learned by interacting with these four environments.

5.2. Policy selection

Once the policy library is constructed, we can assign policies from this library to the zones in the target building. One way to do this is to estimate the performance of each policy in the target environment using a small amount of historical data collected from the target environment under some behavior policy, e.g., the existing rule-based controller. But since the size of the policy library can be quite large, it is costly to estimate the performance of every policy. To address this problem, we propose a policy clustering method that groups similar policies in the library. We can then sample a few policies from each cluster and evaluate their performance using the policy evaluation methods discussed in Section 3.1 and 3.2. This allows us to approximate the performance of other policies that belong to the same cluster. We describe these steps below.

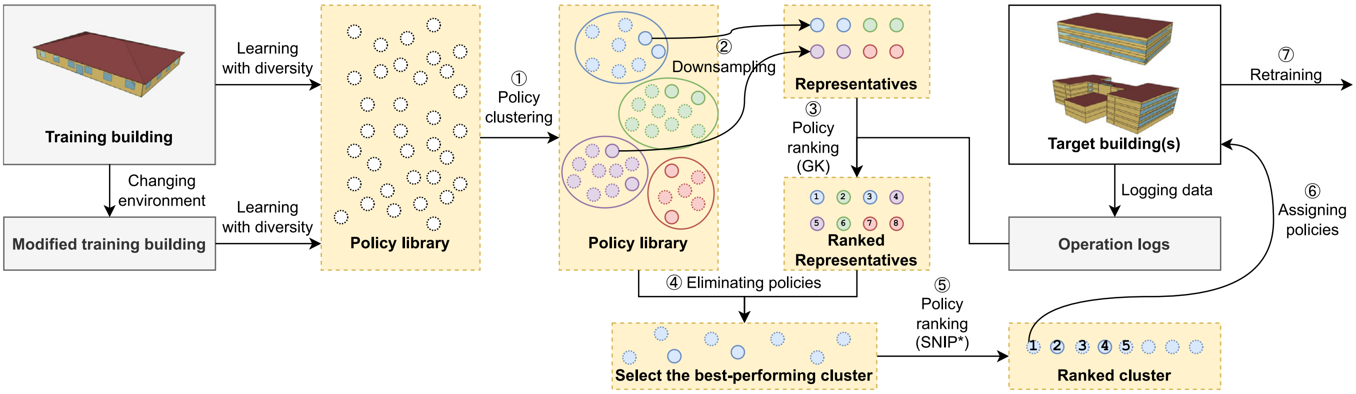


Fig. 2. Schematic overview of the proposed methodology where circled numbers show different steps of the methodology.

5.2.1. Policy representation and clustering

The next step is to identify policies that might have similar performance in the given task. Since we do not know the performance of each policy in the target environment, we cluster policies according to their behavior in the training environment(s). We represent each policy in the policy library using a feature vector of length m. This vector is constructed by sampling $m-1$ states from the distribution of states visited when the policy was being learned in the training environment, and appending the initial state of the target environment. We then use the actions that would be taken from these m states under this policy to obtain the feature vector of length m. We set m to 10 in this study. Given the policy representation in an m dimensional space, we use K-Means to cluster all policies in the policy library. The elbow method is used to determine the number of clusters. Specifically, we keep increasing the number of clusters starting from one cluster and calculate the *inertia* of the current clustering result. The inertia is defined as the sum of the squared differences of all samples from the respective cluster center. We stop when the inertia starts decreasing linearly. After the clusters are formed, we select n representative policies from each cluster. This includes the policy that is closest to the cluster center and $n-1$ randomly selected policies from that cluster. The closet policy to the cluster center is picked as it may represent the average performance of the cluster in the training environment(s), and the other randomly picked policies increase our confidence in the evaluation result. We set n to 5 in this study.

5.2.2. Ranking policies using historical data

Recall that $\mathcal{D}$ contains the historical data collected from the target building under the behavior policy π_b which can be the existing rule-based controller. To rank policies, we use importance sampling to estimate the value of the evaluation policy π_e from a trajectory sampled from $\mathcal{D}$. We adopt the three OPE methods introduced in Section 3.1. The first two are IPW and SNIPW that assume the action space is discrete, we discretize the action space using the Freedman Diaconis estimator [24]. The third OPE method is GK. We use a Gaussian kernel with a bandwidth of 0.3. The GK method can work with continuous action spaces.Apart from the OPE methods, we also test two ZCP methods, namely GN and SNIP, both of which are introduced in Section 3.2. As previously mentioned, in RL, calculating the loss is not possible without deploying the policy on the target building. We overcome this limitation by re-weighting the rewards obtained in the offline dataset using the importance sample ratio $\rho(s,a)$ defined in Section 3.1. Concretely, we sample a trajectory from $\mathcal{D}$ and re-weigh the rewards as follows:

$$\hat{r}_t = \rho(s_t, a_t) r_t,$$

where $a_t, s_t, r_t \sim \mathcal{D}$ are respectively the action taken, state, and reward received at time step t. We limit the size of $\mathcal{D}$ to 15 days of historical data. By replacing the actual rewards with the re-weighted rewards in the trajectory $\mathcal{D}$ we get a modified trajectory that acts as a proxy for having deployed the evaluation policy on the target environment. This proxy trajectory can then be used to calculate $\mathcal{L}_{PPO}$, defined in (1). The loss is then backpropagated to calculate the gradients for each layer, which are used by the gradient-based ZCP methods (GN and SNIP). For clarity, we distinguish GN and SNIP methods that use the proxy trajectory by referring to them as GN* and SNIP*, respectively. Section 7.2 compares the efficiency of the five policy ranking methods, namely SNIP*, GN*, IPW, SNIPW, GK.

5.2.3. Evaluation metrics for policy ranking methods

To compare the performance of different ranking methods, we obtain an estimate of the expected return for each policy by running it in the target environment. This yields a ranking of all policies, which we refer to as the *ground truth ranking*. We refer to the expected return of each policy as its *actual value*. Then, each policy is ranked using the offline methods presented in the previous section. We compute the following evaluation metrics:

- **Spearman's rank correlation coefficient** is the Pearson correlation between the ground truth rank set and estimated rank set. The higher the correlation coefficient, the closer the estimated rank set is to the ground truth rank set.
- **Regret@n** is the difference between the actual value of the best policy in the ground truth set, and the actual value of the best policy in the top-n policies, i.e., the n policies with the highest estimated values according to the offline ranking methods. The lower the value of Regret@n, the better is our offline ranking method.

5.2.4. Policy selection

In Fig. 2, there are two places where policy evaluation is performed, namely Step 3 and Step 5. In Step 3, we rank the representative policies from each cluster to obtain the ranking of clusters, whereas in Step 5, we only rank the policies from the top cluster. The best-performing policy from the top cluster is then transferred over to the (novel) target environment. All steps shown in Fig. 2 are repeated for each zone in the target building to identify the policy that should be transferred and used for that particular zone.

5.3. Policy transfer and retraining

After assigning the best policy to each zone in the target building, we retrain all policies using the multi-agent reinforcement learning framework in an online fashion. Updating the policies through interaction with the target building allows the transferred policies to further adapt to the target building environment.

6. Source and target buildings

To study the efficacy of the proposed methodology, we evaluate it using the EnergyPlus model of three buildings, including a real campus building. Each building has a unique occupancy schedule which is encoded in the EnergyPlus model. We assume that if a control policy outcompetes other policies with respect to the HVAC energy use reported by EnergyPlus [23] without degrading thermal comfort, it also outcompetes them in the real building, should it be controlled using this policy.[2]

- **Building A** is a small office prototype building as defined by ASHRAE Standard 90.1 [25]. Fig. 3a shows the floor plan and 3D model of this building. It contains five thermal zones (4 perimeter zones and 1 core zone) and is located in Denver, Colorado. Each zone is conditioned using a dedicated AHU and contains a VAV system. The total floor area of this building is 511.16 m^2.
- **Building** $\mathrm{B}_{\mathrm{Denver}}$ is a medium office prototype building as defined by ASHRAE Standard 90.1 [25]. It contains 15 thermal zones across three floors and is located in Denver, Colorado. Fig. 3b depicts the floor plan of this building. There are 4 perimeter zones and 1 core zone on each floor. Each floor is conditioned using an AHU and all zones are equipped with a VAV system. Its total floor area is 4,982.19 m^2.
- **Building** $\mathrm{B}_{\mathrm{SanFrancisco}}$ is the same building as $\mathrm{B}_{\mathrm{Denver}}$ with two main differences: 1) it is located in San Francisco, California and 2) its orientation is rotated by 45 degrees (clockwise). We make these changes so as to investigate whether any of the learned policies works well after transfer to a building with a different orientation in a different climate.
- **Building C** is a medium campus building representing the model of the building that houses the Department of Energy Engineering at Sharif University of Technology in Tehran, Iran.[3] It contains 26 thermal zones spread across five floors, 11 of which are equipped with a VAV system. The HVAC, lighting, and blind systems are modelled such that they match the design of these systems in the physical building. We assume the building is located in San Francisco, California, because weather data is lacking for its actual location. The total floor area of this building is 5,051 m^2.

Notice that the 3D views are scaled in Fig. 3 to demonstrate the relative size of these buildings. Building A and Building B have similar floor plans, yet their HVAC systems are different and their core zones have different sizes. We also note that each building has a unique occupancy schedule that is specified in the respective EnergyPlus model.

7. Experimental results

In this section we describe our experiment setup, validate different parts of our methodology using microbenchmarks, and finally make a comparison with baseline control methods in terms of the total HVAC energy use. We start the evaluation on Building $\mathrm{B}_{\mathrm{Denver}}$, and then run experiments on Building $\mathrm{B}_{\mathrm{SanFrancisco}}$ and the model of a real building (Building C). Building A is our source building which is used to learn policies that constitute the policy library.

Our primary evaluation metric is the total HVAC energy consumption. This is because our agents only change the minimum damper position and all other control points are adjusted by the EnergyPlus controller to satisfy thermal comfort requirements. We empirically corroborate this by looking at Predicted Mean Vote (PMV) under different control policies. We find that the EnergyPlus controller manages to maintain the same PMV level regardless of the minimum damper position. We also check if the amount of outdoor air entering the zone satisfies the requirement defined in ASHRAE 62.1 [26] ($2.5L/s \cdot person$). We find the EnergyPlus controller violates this requirement less than 0.004% of the time, which is on par with the default controller.

7.1. Implementation details

We now describe our implementation.[4] Control agents are trained using PPO with $\epsilon = 0.2$ (see Eq. (1)). We use a neural network with two hidden layers, each consisting of 64 units, for actor and critic networks. Hyperbolic tangent is used as the activation function. To sample continuous actions from the actor network, we use a Gaussian distribution for the actor network to parameterize the action. We set the learning rate to 0.0003 and the batch size to 2,976 which is equivalent to one episode.

We simulate the building operation using EnergyPlus 9.3 [23] with the actual weather data for each geographical location, and use COBS [27] to interface with the simulation environment. The control policies are trained using PyTorch [28]. The EnergyPlus model uses a 15-min simulation time step, and each episode is one month. This is equivalent to 2,976 timesteps. The training and test periods are both in January to eliminate the seasonal effect in our simulation.[5] We use weather data from January 1991 for training and from January 2000 for testing. For each experiment, we consider 15 independent runs to calculate the average performance.

Building A is used to build the policy library considering both policy and environment diversity as outlined in the previous section. All policies are trained using PPO under the MARL framework for 1,000 episodes to ensure convergence. We consider three policy diversity weights $w \in \{0.1, 1, 10\}$ to identify near-optimal policies. These policies are forced to be different from the optimal policy π^* ($w = 0$) that is learned for the given zone, hence $\Pi_{learned} = \{\pi^*\}$ in Eq. (4). This results in 800 policies in the policy library — 10 random seeds for training × 4 training environments × 5 zones per environment × 4 diversity weights. We set the upper bound on the probability ratio to 100 ($\bar{\rho}$ in Eq. (4)). Since we select 5 representative policies from each cluster in Step 5 (as explained in Section 7.3), we use Regret@5 along with Spearman's rank correlation to evaluate different ranking methods. We build the policy library on a server with Intel Xeon E5-2650 v4 (2.2 GHz CPU) and NVIDIA Tesla P100 GPUs. It takes about 100 CPU days (mostly for EnergyPlus) and less than 1 GPU day to construct the library.

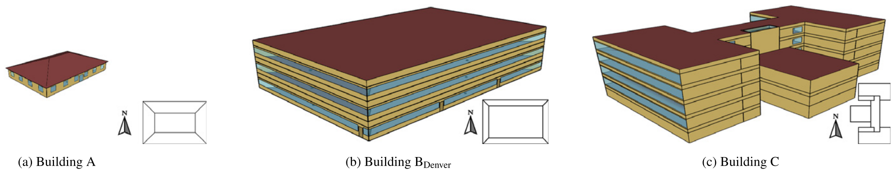
Baselines. We consider four baselines: 1) the default controller implemented in the building model, 2) zone-level control policies learned via interaction with the target building (without transfer learning) using the MARL framework, 3) a control policy that decides on the minimum damper position of all zones and is learned through interaction with the target building (without transfer learning) in the single-agent RL (SARL) framework, and 4) zone-level control policies learned on the source building and trans-

[2] We could not possibly deploy the many control policies we considered in this work on real buildings to run the microbenchmarks. As a result we evaluated them using EnergyPlus. In practice, the source building, where we train a diverse population of agents, might be an EnergyPlus model, but the target buildings are physical buildings.

[3] Model is downloaded from https://github.com/DOEE-BMS/EnergyPlus-Model

[4] Code is available at https://github.com/sustainable-computing/building-MARL.

[5] Studying seasonal effects is deferred to future work.



(a) Building A (b) Building $\mathrm{B}_{\mathrm{Denver}}$ (c) Building C

Fig. 3. The 3D view and floor plan of the buildings considered in this paper where north is marked on each floor plan.

ferred to the target building assuming an oracle produced the optimal assignment of policies to zones in the target building. The last baseline is unrealistic and gives a lower bound on the building energy consumption using the proposed method. We could not implement this baseline because identifying the best policy for each zone requires exhaustive search and expensive evaluation. The first baseline is a controller that can be readily used (or is actually being used in case of Building C) — if we beat this baseline, it means that our policies can reduce the HVAC energy use without sacrificing thermal comfort.

7.2. Comparing policy ranking methods

We compare different policy ranking methods on $\mathrm{B}_{\mathrm{Denver}}$. To obtain the ground truth ranking of the policies, we use brute force search: we deploy each policy onto the target building ($\mathrm{B}_{\mathrm{Denver}}$) and calculate the HVAC energy use. Since there are 15 zones in the target building, we obtain 15 sets of ground truth rankings. We then use different policy ranking methods to rank policies for each zone. The resulting ranking is compared with the ground truth ranking for the same zone to compute the two metrics described in Section 5.2.3.

Fig. 4 (left) shows Spearman's rank correlation for all the policy ranking methods. Among the OPE methods, IPW has a mean Spearman's rank correlation of 0.71, whereas SNIPW has a mean of 0.12, indicating that IPW performs considerably better than SNIPW. The GK method has a mean of 0.84 and its performance is better than that of IPW with statistical significance ($P < 0.0001$). The two ZCP methods have similar performance with GN* and SNIP* having a mean Spearman's rank correlation of 0.65, and 0.66 respectively. From this figure, we conclude that the GK method performs best in terms of Spearman's rank correlation.

Fig. 4 (right) compares the Regret@5 metric for all the ranking methods. The regret values are normalized according to the maximum difference between the actual value of the best and worst policies in the ground truth set. We see that the GK method with a mean regret of 0.19 is not the best performing method. Instead, SNIP* yields a mean regret of 0.05, which is the lowest among the ranking methods. When comparing SNIP* with GN*, the former has a lower Regret@5 with statistical significance ($P = 0.018$).

Although the GK method has the highest Spearman's correlation, SNIP* outperforms the OPE methods when it comes to Regret@5. To take advantage of the GK method's high Spearman's correlation as well as the ability of SNIP* to more accurately identify the top-performing policies, we employ GK for Step 3 and SNIP* for Step 5 of our proposed methodology as shown in Fig. 2.

7.3. Policy clustering analysis

Next we examine our policy clustering result to coordinate that policies in the same cluster perform similarly in the target building. Ideally the top cluster should contain the majority of well-performing policies. We consider $\mathrm{B}_{\mathrm{Denver}}$ for this microbenchmark. The elbow method suggests clustering the policy library into six clusters for all zones.

After ranking representative policies from each cluster in Step 3, we only consider the top performing cluster. Given that the elbow method yielded 6 clusters, Step 4 eliminates 83% of the policies in the policy library. We assess the risk of incorrectly removing well-performing policies by plotting the ground truth energy performance distribution for all clusters in Fig. 5. The x-axis represents the total monthly energy consumption if the policy is selected as the behavior policy for the given zone, and all other zones are controlled using the default controller. The left-most curve in Fig. 5 shows the empirical CDF of policies that belong to the top cluster. Interestingly, more than 50% of these policies keep the total monthly energy consumption below 9.5MWh. This is while the other clusters barely include a policy that achieves the same performance. This implies that the left-most curve represents the best performing cluster and neglecting policies in other clusters should not affect the HVAC energy consumption. Although there is a small overlap between the top three clusters, for every zone, there is at least one policy in the top cluster that is better than all the policies in these two clusters.

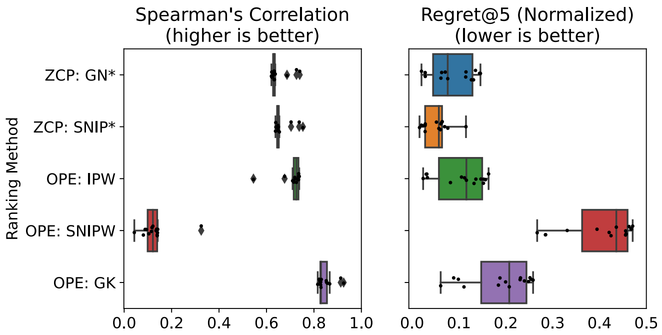


Fig. 4. The evaluation metrics for policy ranking methods where each dot represents the score for a zone. The ground truth ranking was obtained by manually testing each policy in the policy library on $\mathrm{B}_{\mathrm{Denver}}$.

Recall that we sample $n = 5$ policies from each cluster to estimate the performance of each cluster. From Fig. 5, we conclude that even if we sample only 1 policy from each cluster, the chance of incorrectly identifying the best performing cluster is slim. Sampling 5 policies would further reduce the probability of misidentifying the top cluster.

7.4. Policy transfer to $\mathrm{B}_{\mathrm{Denver}}$

In the previous section, we argued that combining GK for cluster ranking with SNIP* for policy ranking within the top cluster enables us to take advantage of high Spearman's correlation of GK as well as low Regret@5 of SNIP*. We refer to this combination of policy ranking methods as GK-SNIP*. We compare the zonal control policies selected from the policy library by GK-SNIP* with four baselines introduced in Section 7.1 to evaluate the efficacy of the proposed methodology.

Fig. 6 shows the performance of our proposed method with other baselines on the target building $\mathrm{B}_{\mathrm{Denver}}$, which is different from the source building, Building A, in terms of the floor area and HVAC design. However, both buildings are located in the same city and they have relatively similar floor plans. All policies are either selected from the policy library or initialized randomly

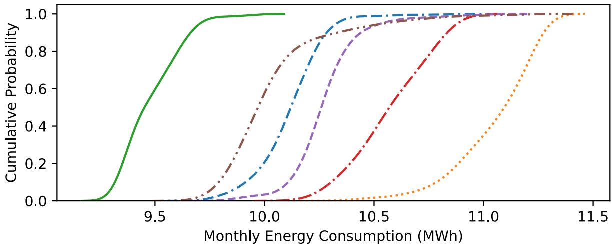


Fig. 5. The cumulative density plot for the distribution of policies' performance when we form six clusters on a select zone in building B_{Denver}. Each line represents the distribution of one cluster. A similar result can be obtained from other zones in the building as well.

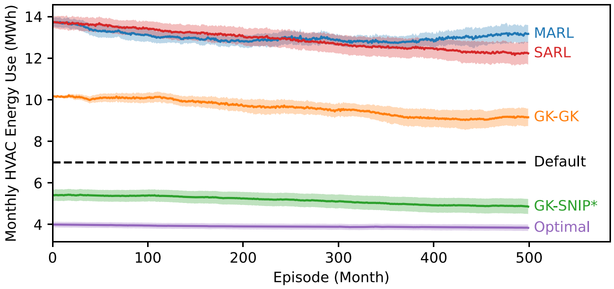


Fig. 6. Learning curve of different controllers on Building B_{Denver}. Each solid line shows the average performance of 15 runs and the shaded area shows one standard error from the mean. The y-axis is exaggerated.

(SARL and MARL). Regardless, they are (re) trained for 500 episodes (months). Policies that need extensive training are not suitable for deployment on real buildings. For instance, the SARL controller trained on the target building (without transfer learning) reaches the same level of performance as the optimal policies assigned from the policy library only after 15,000 episodes, i.e., 1,250 years after the deployment!.

It can be readily seen that the proposed policy selection and transfer method provides a reasonable assignment for all zones in B_{Denver}. The performance of the proposed GK-SNIP* policy ranking method at episode 0 (5.41 MWh) is 22.5% better than the default controller that is presumably designed by HVAC engineers (6.98 MWh). It is also significantly better than SARL (13.74 MWh) and MARL (13.77 MWh). This suggests that GK-SNIP* can be employed to select policies that have reasonable performance on the target building. The optimal assignment has an initial total energy cost of 3.99 MWh. The difference between the proposed policy selection method and the optimal selection is partly due to how we sample policies from the top cluster. Note that the policies assigned to the target building under the optimal assignment do not benefit significantly from retraining. Specifically, the total HVAC energy consumption reduces by 3.8% (from 3.99 MWh to 3.84 MWh) after 500 episodes. We believe this is because there is not much room for improvement as we are already close to the minimum HVAC energy consumption that could be realized by a controller in this building given its occupancy schedule and comfort requirements.

The performance of zonal control policies selected by GK-SNIP* improves by 10.2%, reaching the total monthly energy consumption of 4.86 MWh after 500 episodes of training on B_{Denver}. This is 30.4% less than the energy consumption of the default controller. Policies trained only on B_{Denver} (not transferred from Building A) fail to reach a level of performance that is comparable with the default controller at the end of the 500 episodes. SARL reaches 12.23 MWh and MARL reaches 13.17 MWh of monthly energy consumption. We also witness an increase in the energy consumption under MARL after around 200 episodes. This might be because agents are not collaborating with each other. As a result, they start to cancel out each other's action (creating a "fighting zones" situation), thereby increasing the total HVAC energy use.

7.5. Policy transfer to other buildings

To further validate our proposed methodology, we consider two target buildings ($B_{SanFrancisco}$ and C) that have some major differences with the source building (Building A). Building $B_{SanFrancisco}$ is located in a warmer climate than the source building. Moreover, it differs from the source building in terms of the floor area and HVAC design. Building C is a real building. It has several differences with the source building, including its size, occupancy, floor plan, HVAC design, and weather conditions.

Fig. 7 depicts the performance comparison in Building $B_{SanFrancisco}$. The total energy consumption in all cases is lower than Fig. 6 because we are looking at a winter month with a higher average outside temperature in San Francisco, reducing the heating demand of the building. Most of the observations made in Section 7.4 are true in this case. Before retraining, the zonal control policies selected from the policy library by GK-SNIP* achieve 16.4% lower monthly energy consumption (3.31 MWh) than the default controller (3.96 MWh). The optimal assignment yields the lowest monthly energy consumption at episode 0 (2.01 MWh), which is 49.2% lower than the default controller. After 500 episodes of training, the policies selected by GK-SNIP* reduce the total HVAC energy consumption by 10.3%, reaching 2.97 MWh. This is 25.0% lower than the energy consumption of the default controller, yet 50.8% higher than the optimal assignment.

Fig. 8 compares the performance of the proposed method with the four baselines in Building C. The same observation can be made here too. GK-SNIP* performs better than the default controller, SARL, and MARL, and is slightly worse than the optimal assignment. The default controller consumes 4.83 MWh of energy in one month, whereas the proposed GK-SNIP* method reduces it to 4.71 MWh before retraining and to 4.64 MWh after 500 episodes of training in the target building. These numbers are 4.413 MWh and 4.411 MWh for the optimal assignment.

Our experiments support the claim that diversity-induced RL offers clear benefits for transferring policies to a novel target building, and that the proposed GK-SNIP* policy selection and transfer method can efficiently identify policies, among the policies in the policy library, that perform relatively well in the novel target building using only 2 weeks of historical data. The transferred policies consistently outperform the default controller in terms of the HVAC energy use without sacrificing thermal comfort. This is the case even before these policies are retrained to adapt to the new environment.

7.6. Ablation studies

This section considers two ablation experiments to further explore understand the performance of the different approaches.

7.6.1. Other combinations of ranking methods

We now study the effect of using different policy ranking methods in the OPE evaluation steps. In particular, we look at the result of using GK in both steps (labelled GK-GK) and using SNIP* in both steps (labelled SNIP*-SNIP*). From Figs. 6–8, it can be readily seen that GK-SNIP* always achieves lower monthly energy consumption than GK-GK. This is true before and after retraining on the target environment. We attribute this to the fact that once the best performing cluster is identified, the Regret@5 metric becomes more relevant as we aim to identify the top performing policies.

It is important to point out that the difference between GK-SNIP* and SNIP*-SNIP* is not significant. For this reason, we remove SNIP*-SNIP* from all figures so that GK-SNIP* result can

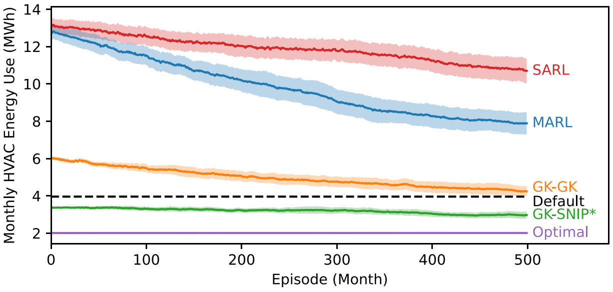


Fig. 7. Learning curve of different controllers on Building $B_{\text{SanFrancisco}}$. Each solid line shows the average performance of 15 runs and the shaded area shows one standard error from the mean.

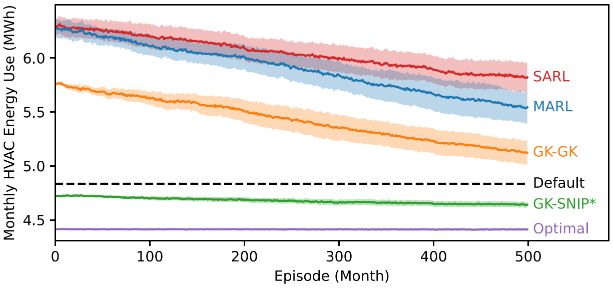


Fig. 8. Learning curve of different controllers on Building C. Each solid line shows the average performance of 15 runs and the shaded area shows one standard error from the mean. The y-axis is exaggerated.

be better seen. The difference between GK-SNIP* and SNIP*-SNIP* becomes more pronounced as the number of episodes for training increases. But even after 500 episodes of training, the policies selected by SNIP*-SNIP* consume only 1.0% (4.91 MWh), 3.4% (3.07 MWh), and 0.2% (4.65 MWh) more energy in Building B_{Denver}, Building $B_{\text{SanFrancisco}}$, and Building C respectively, when they are compared with the policies selected by GK-SNIP*. We attribute this small gap to the fact that the top cluster is well-separated from the other clusters; therefore, the first round of evaluation is more robust to potential estimation errors.

7.6.2. Effects of incorporating diversity

Is it necessary to incorporate both types of diversity for transfer learning? To answer this question, we examine the policy assigned to each zone in the optimal assignment. Several important observations can be made. First, all the policies assigned to the target building are trained with a nonzero policy diversity weight, so they are near-optimal policies in the source building. Second, the best policy found for six zones in Building B_{Denver}, one zone in Building $B_{\text{SanFrancisco}}$, and one zone in Building C is learned on the source building using both environment and policy diversity. Third, the core zones on the top and middle floors of B_{Denver} and $B_{\text{SanFrancisco}}$ are assigned the same policy from the policy library. This is the policy that was learned with diversity weight 0.1 for a perimeter zone in the replica of Building A with window blinds that were closed (created using environment diversity). Fourth, the proposed GK-SNIP* method tends to select policies that are trained with diversity for all target buildings. This confirms our hypothesis that both kinds of diversity would benefit transfer learning.

8. Discussion

Our experimental results confirm that by selecting high-quality policies from a library of diverse policies that are learned in a source building and transferring them to the target building, we can significantly reduce the cost of training RL-based controllers and achieve a better performance than the default controller, even before training them on the target building. To identify these policies, we need only two weeks of log data from the target building. This data is typically generated by a reference controller, such as a rule-based controller that was previously used in that building. The reliance on the log data raises an interesting question: how would the performance of our baselines, in particular SARL and MARL agents, change if the same amount of log data was used to train them in an offline fashion?.

In a recent paper, Nweye et al. [29] found that this kind of offline training leads to improved performance in the long run if the log data is generated by an optimized rule-based controller. To investigate the efficacy of offline training in our setting, we use the two weeks of log data generated by the default (rule-based) controller in the target building (our first baseline) to train MARL and SARL agents in an offline fashion via *behavioral cloning* – an imitation learning approach that learns a direct mapping from states to actions. Specifically, in the MARL setting, we use the mean absolute error between the action of the default controller in each zone and the action taken by the corresponding zone-level RL policy as the loss. In the SARL setting, we define the loss as the average Euclidean distance between the default controller actions in an N-dimensional space (where N is the number of zones in the target building) and actions returned by the RL policy for all zones.

Our result indicates that warm-starting MARL and SARL from the default controller improves their performance across all episodes. Nevertheless, the performance gain varies greatly among the three target buildings that we considered, and our proposed approach (GK-SNIP*) consistently outperforms the warm-started MARL and SARL agents, assuming the same amount of log data is used for policy selection and offline training. In particular, in Building $B_{\text{SanFrancisco}}$, the warm-started MARL agent is better than the warm-started SARL agent, and can save 5.1% and 22.0% on monthly HVAC energy consumption compared to the default controller at episode 0 and episode 500, respectively. Moreover, its performance is on par with GK-SNIP* between episode 220 and episode 500. But the performance of the warm-started SARL and MARL agents is markedly worse than GK-SNIP* across all episodes in Building B_{Denver} and Building C. In fact, the warm-started agents cannot even beat out the default controller in Building C. We attribute this variable performance to two main factors. First, more log data is required for behavioral cloning in larger buildings with more zones. In this experiment, we used 2 weeks of log data for a fair comparison with our approach. Second, the default controller is not optimized in some buildings, hence performing offline training using the log data generated by this controller may not be effective. We plan to explore other types of imitation learning and compare the performance of resulting agents with our diversity-based transfer learning approach in future work.

9. Related work on HVAC control

Mounting interest in optimal operation of complex physical systems has led to the design of various rule-based, model-based, and learning-based control strategies. In the context of HVAC control, these strategies minimize the building energy use while maintaining a comfortable indoor environment for occupants. In rule-based HVAC control, rules and setpoint schedules are typically defined by the facilities manager based on their intuition of the building's function and occupancy. Rule-based controllers are relatively easy to implement and can considerably reduce the building energy consumption [3,30]. But their performance heavily relies on the quality of the control rules and setpoints.

In model-based HVAC control, models for heat transfer, occupancy, and other dynamics are used to predict the heat load and energy demand of the building. These models are built using physics-based or data-driven approaches. In MPC, these models are used to minimize energy use and occupant discomfort over a finite time horizon. Such controllers can significantly reduce the energy consumption [31–33], but developing models for large, multi-zone buildings is a challenging task, requiring manual effort or a substantial amount of training data [34]. Even if accurate models are developed and incorporated in the control loop of one building, they cannot be directly transferred and reused in another building.

Learning-based control algorithms, such as model-based and model-free reinforcement learning, are proven to be useful for HVAC control. Specifically, they can find an optimal policy that minimizes energy consumption while maintaining thermal comfort [5,7,35,36]. An RL agent learns the mapping between the state of the building and an action via trial and error. Unfortunately, training these RL agents requires a substantial amount of data to sufficiently explore a large or continuous state-action space. As more features are added to the state, the complexity and the number of parameters used to represent the agent grows exponentially. Moreover, a single agent that controls multiple actuators cannot be easily transferred to another building that has a different state and/or action space. Chen et al. [35] reduce the training cost of an RL agent that controls the HVAC system through a differentiable MPC policy that encodes system dynamics and offline imitation learning, using the operational data collected under a default controller. However, learning an accurate model can be challenging in a given building and more historical data would be needed to fully capture the system dynamics. Similarly, offline RL techniques generally require a substantial amount of historical data before they can learn a high-quality policy.

In a recent survey paper, Pinto et al. [37] have reviewed the applications of transfer learning to buildings, including papers that use transfer learning to address the data inadequacy challenge in developing learning-based controllers for building systems. This literature review reveals that there is no paper that takes advantage of diversity for transfer learning in this domain. Xu et al. [38] address the problem of transferring previously learned HVAC control policies to an unseen building. Their methodology involves decomposing the policy neural network into a transferable front-end network and a trainable back-end network. The front-end network captures building-agnostic behavior, whereas the back-end network needs to be trained on the target building. Although this approach reduces the training cost of RL to some extent, control performance can still be poor while the back-end network is being trained in the target building. In another line of work, Fazel et al. [39] propose augmenting the training data collected from the target building. The authors use generative adversarial networks to learn the building performance profile from the actual data, and generate synthetic data that reflect climate and operation variations, while keeping the building profile the same. However, 1 year data is required to train the generative model, which may not be readily available in all buildings.

Multi-Agent Reinforcement Learning (MARL)-based controllers are proven to be useful in energy-efficient control of building systems [40,41], and are amenable to transfer learning [42]. MARL enables controlling different knobs in one or multiple building systems, e.g., Zhao et al. [43] use separate agents to manage electricity flow, cooling components, heating components in a building. Unlike the previous work that control multiple building systems using MARL, in this paper we decompose the optimal control of the HVAC system into the problem of controlling the environment of individual thermal zones in the building, which can be solved using MARL. In this setting, each agent is responsible for controlling the HVAC components (e.g., control points in the variable air volume system) in one zone of the building. This reduces the size of an agent's state-action space and enables the transfer of policies to other buildings, regardless of the number of zones they have or their floor plan. This is because, at the zone level, most buildings have the same set of sensors and actuators, so the corresponding agents will have an identical state-action space.

In previous work [9], we showed that introducing diversity when learning MARL policies is advantageous for transfer learning. However, we did not explore how to efficiently select high-quality policies for transfer. In this work, we propose a two-step policy selection method, which involves policy clustering and evaluation. We also address the potential issue with the unbounded action probability ratio as discussed in Section 5.1.

10. Conclusion and future work

In this paper we investigated efficient evaluation of a large number of HVAC control policies through policy representation and clustering, and took advantage of it to transfer select policies that are trained on a prototype building to multiple new buildings. The novelty of our work is in (a) designing a policy diversity loss that helps create a library of diverse policies, (b) combining policy clustering and policy evaluation techniques to quickly identify high-quality policies among the policies in the library for the given building, and (c) modifying standard ZCP-based methods to make them applicable to the policy selection problem in reinforcement learning.

We compared the efficacy of various OPE and modified ZCP methods using microbenchmarks, and found that GK and SNIP* are the best performing policy ranking methods. We then ran experiments on three target buildings. These experiments revealed two key findings. First, the proposed offline policy selection algorithm effectively identifies high-quality zone-level control policies using only two weeks of log data generated by the default controller in the target building. Second, diversity can help with generalization to novel environments to a great extent.

Although the transferred zonal control policies perform 4.0–30.4% better than the default controller, which is remarkable, the gap between these policies and the optimal policy is still large. In future work, we plan to investigate other policy ranking methods and improvements to close this gap. Moreover, we will further explore online policy selection methods, and study seasonal effects and policy transfer across different seasons.

Data availability

Our code and building models can be downloaded from this repository: https://github.com/sustainable-computing/building-MARL.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Omid Ardakanian reports financial support was provided by Natural Sciences and Engineering Research Council of Canada. Matthew E. Taylor reports financial support was provided by Natural Sciences and Engineering Research Council of Canada. Matthew E. Taylor and Omid Ardakanian report financial support was provided by Alberta Innovates.

Acknowledgements

This work is supported in part by research grants from NSERC; Alberta Innovates; the Alberta Machine Intelligence Institute (Amii); and a Canada CIFAR AI Chair, Amii.

References

[1] International Energy Agency, Buildings: A source of enormous untapped efficiency potential,https://www.iea.org/topics/buildings, 2022.

[2] N.E. Klepeis et al., The national human activity pattern survey (NHAPS): a resource for assessing exposure to environmental pollutants, J. Exposure Sci. Environ. Epidemiol. 11 (3) (2001) 231–252.

[3] O. Ardakanian, A. Bhattacharya, D. Culler, Non-intrusive occupancy monitoring for energy conservation in commercial buildings, Energy Build. 179 (2018) 311–323.

[4] J. Drgoňa et al., All you need to know about model predictive control for buildings, Annu. Rev. Control 50 (2020) 190–232.

[5] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Appl. Energy 269 (2020).

[6] D.P. Zhou, Q. Hu, C.J. Tomlin, Quantitative comparison of data-driven and physics-based models for commercial building HVAC systems, American Control Conference, ACC 2017, IEEE 2017 (2017) 2900–2906.

[7] T. Zhang, G. Baasch, O. Ardakanian, R. Evins, On the joint control of multiple building systems with reinforcement learning, in: Proceedings of the 12th ACM International Conference on Future Energy Systems, e-Energy '21, New York, NY, USA, 2021, pp. 60–72.

[8] X. Ding, W. Du, A. Cerpa, Octopus: Deep reinforcement learning for holistic smart building control, in: Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '19, ACM, 2019, pp. 326–335.

[9] T. Zhang, et al., Diversity for transfer in learning-based control of buildings, in: Proceedings of the 13th ACM International Conference on Future Energy Systems, e-Energy '22, ACM, New York, NY, USA, 2022, pp. 556–564.

[10] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, MIT press, Cambridge, MA, USA, 2018.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms preprint (2017). arXiv:1707.06347.

[12] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, Y. Wu, The surprising effectiveness of ppo in cooperative, multi-agent games, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, Vol. 2 of NeurIPS 2022, Curran Associates Inc, 2022.

[13] M.A. Masood, F. Doshi-Velez, Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence AI for Improving Human Well-being, IJCAI 2019, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 5923–5929.

[14] J. Parker-Holder, A. Pacchiano, K.M. Choromanski, S.J. Roberts, Effective diversity in population based reinforcement learning, in: Advances in Neural Information Processing Systems, Vol. 33 of NeurIPS 2020, Curran Associates Inc., 2020, pp. 18050–18062.

[15] K.R. McKee, J.Z. Leibo, C. Beattie, R. Everett, Quantifying the effects of environment and population diversity in multi-agent reinforcement learning, Auton. Agent. Multi-Agent Syst. 36 (1) (2022).

[16] D. Precup, R.S. Sutton, S.P. Singh, Eligibility traces for off-policy policy evaluation, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 759–766.

[17] A. Swaminathan, T. Joachims, The self-normalized estimator for counterfactual learning, in: Advances in Neural Information Processing Systems, NeurIPS 2015, Curran Associates Inc, Red Hook, NY, USA, 2015, pp. 3231–3239.

[18] N. Kallus, M. Uehara, Intrinsically efficient, stable, and bounded off-policy evaluation for reinforcement learning, in: Advances in Neural Information Processing Systems, vol. 32 of NeurIPS 2019, Curran Associates Inc, 2019, pp. 3320–3329.

[19] N. Kallus, A. Zhou, Policy evaluation and optimization with continuous treatments, in: Proceedings of the 21st International Conference on Artificial Intelligence and Statistics, Vol. 84 of AISTATS 2018 PMLR, 2018, pp. 1243–1251.

[20] F. Hutter, L. Kotthoff, J. Vanschoren, Automated machine learning: methods, systems, challenges, Springer Nature, 2019.

[21] M.S. Abdelfattah, A. Mehrotra, L. Dudziak, N.D. Lane, Zero-cost proxies for lightweight nas, in, in: Proceedings of the 9th International Conference on Learning Representations, 2021.

[22] N. Lee, T. Ajanthan, P.H.S. Torr, SNIP: Single-shot network pruning based on connection sensitivity, in: Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, 2019.

[23] D.B. Crawley et al., Energyplus: creating a new-generation building energy simulation program, Energy Build. 33 (4) (2001) 319–331.

[24] D. Freedman, P. Diaconis, On the histogram as a density estimator: L2 theory, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 57 (4) (1981) 453–476.

[25] R. American Society of Heating, A.-C. Engineers, Standard 90.1-2019, Energy Standard for Buildings Except Low-Rise Residential Buildings, ASHRAE Inc, Peachtree Corners, GA, USA, 2019.

[26] R. American Society of Heating, A.-C. Engineers, Standard 62.1-2022, Ventilation and Acceptable Indoor Air Quality, ASHRAE, Inc., Peachtree Corners, GA, USA, 2022

[27] T. Zhang, O. Ardakanian, COBS: Comprehensive building simulator, in: Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '20, ACM, New York, NY, USA, 2020, pp. 314–315.

[28] A. Paszke, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, NeurIPS 2019, Curran Associates Inc, Red Hook, NY, USA, 2019, pp. 8024–8035.

[29] K. Nweye, B. Liu, P. Stone, Z. Nagy, Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings, Energy AI 10 (2022).

[30] E. Shen, J. Hu, M. Patel, Energy and visual comfort analysis of lighting and daylight control strategies, Build. Environ. 78 (2014) 155–170.

[31] S. Privara, J. Širokỳ, L. Ferkl, J. Cigler, Model predictive control of a building heating system: The first experience, Energy Build. 43 (2) (2011) 564–572.

[32] D.A. Winkler, A. Yadav, C. Chitu, A.E. Cerpa, Office: Optimization framework for improved comfort & efficiency, in: Proceedings of the 19th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2020, New York, NY, USA, 2020, pp. 265–276.

[33] C. Turley, M. Jacoby, G. Pavlak, G. Henze, Development and evaluation of occupancy-aware HVAC control for residential building energy efficiency and occupant comfort, Energies 13 (20) (2020).

[34] A. Aswani, H. Gonzalez, S.S. Sastry, C. Tomlin, Provably safe and robust learning-based model predictive control, Automatica 49 (5) (2013) 1216–1226.

[35] B. Chen, Z. Cai, M. Bergés, 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, BuildSys '19, ACM, 2019, pp. 316–325.

[36] Z. Jiang, M. Risbeck, V. Ramamurti, S. Murugesan, J. Amores, C. Zhang, Y. Lee, K. Drees, Building HVAC control with reinforcement learning for reduction of energy cost and demand charge, Energy Build. 239 (2021).

[37] G. Pinto, Z. Wang, A. Roy, T. Hong, A. Capozzoli, Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives, Adv. Appl. Energy 5 (2022).

[38] S. Xu, Y. Wang, Y. Wang, Z. O'Neill, Q. Zhu, 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, BuildSys '20, ACM, New York, NY, USA, 2020, p. 230–239.

[39] F. Khayatian, Z. Nagy, A. Bollinger, Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties, Energy Build. 251 (2021).

[40] J.R. Vazquez-Canteli, G. Henze, Z. Nagy, MARLISA: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings, in: Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '20, ACM, 2020, pp. 170–179.

[41] Q. Fu, X. Chen, S. Ma, N. Fang, B. Xing, J. Chen, Optimal control method of HVAC based on multi-agent deep reinforcement learning, Energy Build. 270 (2022).

[42] S. Nagarathinam, V. Menon, A. Vasan, A. Sivasubramaniam, MARCO multi-agent reinforcement learning based control of building HVAC systems, in: Proceedings of the 11th ACM International Conference on Future Energy Systems, e-Energy '20, New York, NY, USA, 2020, pp. 57–67.

[43] P. Zhao, S. Suryanarayanan, M.G. Simoes, An energy management system for building structures using a multi-agent decision-making control methodology, IEEE Trans. Ind. Appl. 49 (1) (2013) 322–330.