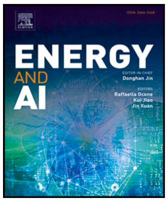
Contents lists available at ScienceDirect

Energy and AI

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



Reinforcement learning building control approach harnessing imitation learning

Sourav Dey [a,*], Thibault Marzullo [b], Xiangyu Zhang [b], Gregor Henze [a,b,c]

[a] Department of Civil, Environmental and Architectural Engineering, University of Colorado, Boulder, CO, USA
[b] National Renewable Energy Laboratory, Golden, CO, USA
[c] Renewable and Sustainable Energy Institute, Boulder, CO, USA

HIGHLIGHTS

- The approach addresses the practical challenges of advanced building control.
- Reinforcement learning harnessing imitation learning significantly reduces training time and early exploratory unstable behavior of an RL controller agent.
- The approach is effective in finding a control strategy better than a rule-based or heuristic strategy.

ARTICLE INFO

Keywords:
Reinforcement learning
Building controls
Imitation learning
Artificial intelligence

ABSTRACT

Reinforcement learning (RL) has shown significant success in sequential decision making in fields like autonomous vehicles, robotics, marketing and gaming industries. This success has attracted the attention to the RL control approach for building energy systems which are becoming complicated due to the need to optimize for multiple, potentially conflicting, goals like occupant comfort, energy use and grid interactivity. However, for real world applications, RL has several drawbacks like requiring large training data and time, and unstable control behavior during the early exploration process making it infeasible for an application directly to building control tasks. To address these issues, an imitation learning approach is utilized herein where the RL agents starts with a policy transferred from accepted rule based policies and heuristic policies. This approach is successful in reducing the training time, preventing the unstable early exploration behavior and improving upon an accepted rule-based policy — all of these make RL a more practical control approach for real world applications in the domain of building controls.

1. Introduction

Background. Buildings are responsible for about 40% of the total primary energy use in the United States and more than 70% of the electricity use [1,2]. Significant energy is spent in buildings for the provision of thermal comfort, specifically by the heating, ventilating and air-conditioning (HVAC) systems. They are responsible for more than 40% [3] of average building energy consumption. In the United States, this is more than 50% [4]. Humans spend more than 86% of their time indoors [5]. Building controllers operating these HVAC systems are responsible for maintaining comfortable, safe, and healthy indoor conditions, while also aiming at reducing the energy consumption of buildings. The aim of maintaining comfortable indoor conditions combined with the aim of reducing energy consumption is in conflict and requires a careful trade-off. Recently, due to developments in building technology and changes in our overall electric power system, building controls are becoming more complicated balancing multiple goals such as catering to grid flexibility, indoor occupancy, or managing on-site renewable energy production and storage [6]. The consideration of these various technologies and operational objectives requires advanced controllers that can make the trade-off between multiple conflicting goals and can also adapt to emerging technologies [7] over time with the resulting changes in the environmental feedback.

Conventional building controls, which are mostly rule-based and heuristic based on expert experience, are unable to achieve such multi-objective optimization. In rule-based controls, the controls rely on predetermined set points, and local proportional–integral–derivative (PID) control loops, is used to maintain these set-points. Best-in-class control strategies have been developed by building control experts and ASHRAE Guideline 36-2018 [8] offers a broad set of recommendations

[*] Corresponding author.
E-mail addresses: sourav.dey@colorado.edu (S. Dey), thibault.marzullo@nrel.gov (T. Marzullo), xiangyu.zhang@nrel.gov (X. Zhang), gregor.henze@colorado.edu (G. Henze).

https://doi.org/10.1016/j.egyai.2023.100255
Received 30 June 2022; Received in revised form 6 March 2023; Accepted 11 March 2023
Available online 14 March 2023

on these. While consensus driven, these rule-based and heuristic strategies may not necessarily be optimal as they are predetermined and not tailored to the specifics of the building and local conditions. These controllers also do not consider consideration forecasts such as weather and occupancy. For these reasons, such conventional rule-based controllers are sub-optimal in their performance and additionally, a good rule-based controller requires considerable engineering time for tuning and performance monitoring to achieve acceptable performance. New advanced control strategies or algorithms like RL or model predictive control (MPC) adapt their control policies to various objectives or cost function while heuristic controls cannot. Moreover, rule-based controls are not suitable for dynamic price based optimization problems that require arbitrage or demand response scenarios.

Although MPC has been successful in process control applications in chemical plants and refineries [9], and has also gained popularity in building control in research [10], it has not yet been adapted widely in the commercial building industry. One of the main bottlenecks is that MPC requires the development and identification of system models which capture the dynamic behavior of buildings and their HVAC systems. However, because every building is unique, unlike cars or airplanes, it is difficult to develop a model for each and every building as it is labor intensive and requires extensive expertise to develop and maintain an accurate model of the building [11]. In this context, RL seems attractive as it learns to take optimal actions by interaction and can improve its control policy over time. Furthermore, RL has the potential to learn better control policies over time and can adapt to changing dynamics even in challenging environments.

Over the past few years, RL has created several success stories. It was able to play a range of Atari 2600 video games at a human level [12], defeat a human world champion in the game of Go [13], and found success in fields like autonomous vehicles [14] and robotic applications [15,16]. Although there are benefits to using RL, the challenges are its long training time and the unstable behavior during the early training phase. Depending on the complexity of the problem in a building control task, the training time can take somewhere between 4 weeks to 40 years worth of data. In a real-world application, a control engineer cannot afford to wait for such a long time for a good control performance to emerge nor can they afford the thermal discomfort, energy costs, or equipment failure due to the unstable behavior of the RL agent in the early exploring phases. In this work, an imitation learning technique has been developed which effectively avoids these issues, whereby the RL agent starts with the knowledge of a rule-based policy and improves on this policy depending on the objectives set for the agent.

Previous work on transfer learning. Transfer learning (TL) is the process of transferring knowledge and information from a source domain and task to improve it in a target domain and task [17]. Due to the knowledge transfer, TL usually leads to a faster learning process in a control task than one that uses no transfer learning. The literature on transfer learning techniques in the context of RL building control applications is an emerging field and there are limited publications on this topic to date. The papers exploring TL in the context of buildings mostly use simplified building models or use simplified tasks. Some of the existing literature is discussed in the next section.

Lissa et al. [18] demonstrated a transfer learning application with a Q-learning framework. With transfer learning, the training time to adjust to rooms with similar sizes and construction was reduced by a factor of six for HVAC controls. This also showed that transfer learning could also achieve favorable results if applied to a new geographical location if the environmental variation is low. Lissa et al. in another publication [19], used a shared parallel transfer learning experience based on [20], from similar buildings equipped with similar equipment, to speed up the learning process. Zou et al. [21] used a long short-term memory (LSTM)-based deep reinforcement learning environment built using one year of building data to pre-train a deep deterministic policy gradient (DDPG) algorithm. The RL agent was able to keep comfort level at 10% predicted percentage of dissatisfaction (PPD) and a reduction of energy consumption by 27% compared to rule-based controls.

Zhang et al. [22] used a TL approach from an existing controller to accelerate the training time of a controller applied to a new, similar residential building by directly copying the control parameters from a trained building agent. Spangher et al. [23] used LSTM [24] planning models to populate the memory buffer of the soft actor critic (SAC) algorithm [25]. The memory buffer contains the vector of current observations, control action, and the next observations of the agent as a result of taking the control action in the environment. A short-term memory buffer that empties after collecting the most recent batches in lower numbers had the best performance when compared to long-term memory, a memory that never empties. Xu et al. [26] used a novel TL method that is scalable to multi-zone buildings with different layouts and building materials by taking advantage of two neural networks - a front-end network generalized to all buildings and a building-specific back-end network.

Costanzo et al. [27] used model assisted batch reinforcement learning (MABRL) using fitted Q iteration. This uses multiple instances of a single layer, single output extreme learning machine (ELM) models [28] to predict the change of temperature. The outputs are averaged to reduce the regression errors of the ELM models. The data from this model is used to fill the state space in areas where the experimental sample density is low. This uses a hybrid offline and online learning approach. There is also an action processor, which can override the control policy according to specific rules involving the outdoor air temperature and the comfort bounds leading to better control. The agent was able to find a decent policy in 10 days and a good policy within 90% of the mathematical optimum in 20 days.

Tsang et al. [29] use a TL method where a suggested action from a trained agent is added to the state observation of a DQN agent minimizing the training time required for convergence. Deng and Cheng [30] developed an occupant behavior model for thermostat and clothing level utilizing an RL approach. Here, a TL approach was utilized, which transferred only the high-level policy from states to action from one trained building to other buildings instead of transferring the actual weights of the neural network. Mocano et al. [31] propose a energy prediction method utilizing a RL approach with two RL algorithms, Q-learning and State–Action–Reward–State–Action (SARSA) [32]. The TL method was implemented utilizing a deep belief network (DBN) [33], which was able to map from discrete states to continuous states.

Tao et al. [34] showcased an effective transfer learning process in battery energy storage systems (BESS) and HVAC systems participating in a demand response (DR) program, utilizing trained fixed shallow convolution layers and proposing an Evolving Domain Adaptation Network (EDAN) approach based on [35], in the deeper layers for specific target domains. Fan et al. [36] used a similarity evaluation function based on Euclidean distance between supply and demand curve between source and target domains, to evaluate which source domain would be more suitable to transfer to the target in micro-grid scheduling.

We reviewed some transfer learning approaches in buildings as in this study we utilize imitation learning for the source task and use one of a popular RL algorithm for the target task. The details of the source and the target tasks used in this study are described in the following section.

2. An imitation learning approach with RL

RL algorithm preliminary. A reinforcement learning algorithm is mainly based on the Markov decision process (MDP), which assumes that the present state only depends on the last state but not on the trajectory that led to the last state. These are mainly formulated in the form of a tuple of (S, A, R, P), where S is the environmental state space in which the controller needs to make a control decision, A is the action space, R is the rewards or feedback from the control action implemented at each time-step and P is the probability distribution of the state transition. Usually in a MDP control task, a control action is taken after each time period, where at time t, the agent observes a state $s_t \in S$, chooses an action $a_t \in A$, the transitional probability P determines the next state observation s_{t+1} in which the agent will transition as well as the rewards r_{t+1} obtained (where $r_t \in R$), based on the action a_t.

The aim of the agent is to maximize the expected return $\mathbb{E}[G_t]$, where $G_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$. γ is a parameter ($0 < \gamma < 1$) called the discount rate, which determines the relative importance the agent will give to the immediate return compared to future rewards. A value closer to 0 makes the agent prioritize immediate returns rather than long-term returns. T is the terminal state, but for continuous tasks, $T = \infty$ and this sum returns a finite value as the infinite series converge due to $\gamma < 1$. Reinforcement learning algorithms can be of three types: (1) value-based, (2) policy based, and (c) hybrid actor–critic style which is essentially a mixed approach between value and policy based algorithm. Policy based algorithm work by developing a direct mapping of states to action from feedbacks from the building. Value-based algorithm on the other hand work by developing a value-function and indirectly deriving a policy from the value-function learnt from the environment. Value-function is essentially an expectation of the return or $\mathbb{E}[G_t]$. In a hybrid actor–critic method the policy is known as the actor, which is used to select the actions and critic estimates the value-function and evaluates the action taken by the actor. For high-dimensional and complex non-linear environments the value-function and the policy are usually represented by feed forward neural networks (NN).

A feed forward NN is a complicated function approximator consisting of densely connected artificial neurons. Typically an artificial neuron computes the weighted average of inputs and passes the sum through a non-linear activation function. Each layer in a NN comprises of a number of parameterized artificial neurons. NN has essentially three connected structures: (1) the first input layer, (2) the hidden layers in the middle and, (3) the last output layer. The input layer is where the NN receives the initial raw data for the NN, the hidden layers are intermediate layers between input and output layer where the data is processed applying complex non-functions to the inputs and the output layers produces the results for the given inputs. For further details on artificial neural network and its learning process readers can consult [37].

PPO algorithm. An RL controller learns to maximize the expected return $\mathbb{E}[G_t]$ from the environment by receiving feedback from the actions. RL algorithms can be classified into two types, value-based or policy based or a mixture of both. Proximal policy optimization (PPO) is a hybrid mixed policy between value and policy based state-of-the-art algorithm which has shown good performance in various tasks. In policy based methods, a control policy $\pi_\theta(a|s)$ is learned to maximize the expected return $J(\theta)$ generally formulated by $\mathbb{E}_{\pi_\theta}(G_t)$, where θ are the policy parameters. Here, the agent performs a stochastic estimation of the gradient $\hat{\nabla} J(\theta)$ from the trajectories of experience already collected and implements a gradient ascent (i.e., $\theta_{t+1} = \theta_t + \alpha \hat{\nabla}_\theta J(\theta)$) in a training update. The PPO algorithm originates from Trust Region Policy Optimization (TRPO) [38], a policy gradient method.

Instead of having a Kullback–Leibler (KL) divergence constraint [39] in TRPO, PPO simplifies this by using a clipped surrogate objective $L^{CLIP}(\theta)$ to limit the update of the new policy from the old policy. The ratio $r(\theta)$ is the probability ratio between new policy π_θ and old policy $\pi_{\theta_{old}}$, with θ being the parameters of the policy, and $\hat{A}_\theta$ representing an advantage function.

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} \tag{1}$$

$$L^{CLIP}(\theta) = \mathbb{E}[\min\left(r(\theta)\hat{A}_\theta(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_\theta(s,a)\right)] \tag{2}$$

The final objective consists of the clipped objective, an entropy term to encourage exploration, and an error term for a better value function estimate. PPO is implemented in an actor–critic style where the actor network tries to maximize the clipped objective as well as the entropy term, while the critic network tries to minimize the loss in the value function so that it has better value estimates, which is utilized to calculate the advantage function $\hat{A}_\theta$ estimates. Interested readers can consult [40] for more details on the PPO algorithm.

Imitation learning. Imitation learning has been a key learning approach in the autonomous behavioral systems commonly seen in robotics, computer games, industrial applications, and manufacturing as well as autonomous driving. Imitation learning aims at mimicking a human behavior or an agent which is considered to perform well in a particular task. This is essentially learning to map observations to actions. It helps in reducing the task of teaching an agent, by showing the agent the actions to take to complete a specific task. Imitation learning is useful in fields like autonomous vehicles, robotics, and other industries where large sensory data of expert human demonstrations are available.

The design of imitation learning approaches is usually of two types, inverse reinforcement learning, and behavioral cloning. In inverse reinforcement learning the reward function is unknown and this needs to be recovered from the existing expert demonstration. Behavioral cloning is a simple direct mapping of states s_t to the control inputs or actions a_t [41] as shown below,

$$a_t = \pi_{\theta_a}(s_t) \tag{3}$$

The policy can be learned by a supervised learning method from a dataset of demonstrated trajectories $D = \{s_t, a_t\}$. A parameter set θ_a of a neural network is used here to learn the mapping from the states to the actions. In this study, we utilize the behavioral cloning method to warm start the RL agent instead of inverse reinforcement learning due to its simplicity. We assume that we do not have access to any historical data or rule-based demonstrations. The data needed for training are artificially generated in the form of state–action (s_t, a_t) tuples where the actions a_t follow the conventional rule-based actions. The details of forming the artificial data for a particular application are mentioned in the paragraph on artificial data generation in Section 4.

Imitation learning approach in buildings. Multi-objective optimization in buildings is difficult as each building is unique and generally the optimization problems are complex and non-convex. Additionally, developing tailored and accurate building models is not a scalable approach. MPC, which requires a detailed accurate model to perform the optimization, thus suffers from a lack of scalability.

The tradition in building controls is to develop heuristic rules designed by control engineers from domain expertise and experience even though they may be sub-optimal in their long-term performance. These rule-based control policies are simple, easy to implement and interpretable, which makes them widely accepted in the building controls application. When trying to achieve high performance, these heuristic strategies require extensive engineering effort. The control engineers are generally hesitant to implement advanced controls in place of the rule-based controls as they may be unexplainable [42]. Imitation learning can address this unwillingness to implement advanced controls as with this proposed reinforcement learning approach the controller starts with a rule-based policy. With real feedback from the building over time, the RL algorithm evaluates whether the starting rule-based policy is best adjusted for the intended multi-objective optimization and modifies this policy in favor of reducing the penalty objectives.

In applications like autonomous vehicles and robotics, the human demonstrations are considered to be the expert and the controller learns how to effectively imitate the human. However, in the world of building control, this is different as the rule based or heuristic controllers are adequate but typically sub-optimal. They mostly consist of conditional rules usually based on temperature and time. Imitation learning reproduces these demonstrated rule-based behavior as the starting policy avoiding the pitfalls of the unstable early training period as well as the training time to reach a decent policy. This is done by pre-training the RL agent's actor policy to learn to map states to rule-based or heuristic actions without the knowledge of the consequences of the action, which in RL terms means the feedback in the form of rewards. The RL agent, after interacting with the real building by receiving real feedback, learns to evaluate this pre-trained imitated policy and finds an approach which improves upon the original imitated policy.

In this study, we generate an artificial set of data consisting of the tuples of (state s_t, action a_t) assuming that we do not have access to historical building data. The input states for the imitation learning are the same for reinforcement learning observation states s_t. The PPO algorithm is utilized here after the imitation learning supervisory training. The trained network parameters θ_a are passed only to the actor network of the PPO agent while the critic network is initialized randomly. The PPO agent then interacts with the buildings and modifies its policy to achieve better results than the rule-based policy. It is generally recommended to keep the hyper-parameters for learning rate and the entropy term low to limit the exploration process and avoid large training updates. If historical building data were available, they can also be used for this supervisory training by forming the states s_t and action a_t as per the formulation of the intended RL problem. However, sometimes historical data may not be sufficiently rich and the state space might not be sufficiently covered. Thus, some artificial datasets need to be created in the sparsely visited parts of the state space and added to the historical data for the imitation learning part.

Algorithm 1: Imitation Learning with RLC.

```
Formulate the RL problem with states $s_t$ out of available building feature space $x_t$ and extraneous variables $z_t$ as well as control action $a_t$
if Rule-based historical data available then
    Form state–action tuples of $(s_t, a_t)$ out of available historical building data and weather data
    Add artificial state–action tuples of $(s_t, a_t)$ where $s_t$ explores the state space, where the rule-based action does not, and $a_t$ is rule-based action based on $s_t$
else
    Form artificial state–action tuples of $(s_t, a_t)$ where $s_t$ explores the full state space, and $a_t$ is rule-based action based on $s_t$
Perform supervised learning with actor-network parameter $\theta_s$ to learn to map states $s_t$ to action $a_t$
End supervisory training until the training is validated with an acceptable accuracy
Pass the trained network $\theta_s$ to the actor network $\theta_{act}$ of the PPO agent ($\theta_{act} \leftarrow \theta_s$) and initialize the critic network $\theta_{cri}$ randomly
Train the PPO agent with network parameters, $\theta_{act}$ and $\theta_{cri}$ by interacting with the real building.
```

3. Building framework used

Advanced controllers are normally developed, tuned, and tested for their performance in virtual test bed environments before being deployed in the field. In this research, we use an open-source building performance simulation test bed, the Advanced Controls Test Bed (ACTB) [43], which is a software environment for developing and testing advanced building controllers in a high-fidelity realistic building environment to realistically simulate building control system behavior, a feature previously lacking in other open-source platforms.

The ACTB interfaces with Spawn of EnergyPlus building models, making it a high fidelity building model. Spawn of EnergyPlus is a model exchange development by the U.S. Department of Energy that uses EnergyPlus to simulate the heat balance method of the internal gains, envelope heat transfer, and thermal loads combined with Modelica-based HVAC systems, fluid loops, and underlying low-level controls [44]. This is an upgrade from EnergyPlus in terms of control applications as EnergyPlus is unable to implement closed-loop control capabilities, where the simulation engine applies a quasi-static load-based simulation of thermal loads and HVAC response. This is addressed by the equation-based Modelica HVAC and controls model where the simulation is dynamic and implements control actions as it would in real physical buildings. Spawn allows the packaging and compilation of the EnergyPlus model as functional mock-up units (FMUs), which interact with HVAC components written in the Modelica language.

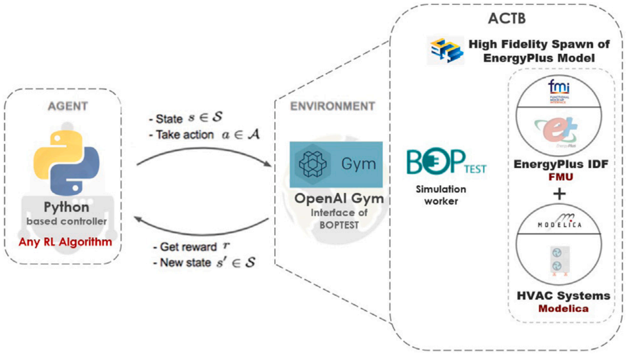


Fig. 1. ACTB framework.

The ACTB utilizes the Building Operation Performance Test (BOPTEST) [45] and Alfalfa [46] frameworks to manage the simulations, a representational state transfer (REST) application programming interface (API), and key performance indicators (KPIs) for evaluating the effectiveness of control strategies. The REST API provides the control developers a user-friendly experience to develop an advanced controller agnostic of the programming language, further enabled by well-established control libraries to interact with a realistic building environment. Fig. 1 illustrates the architecture of the advanced control testbed.

The RL research work presented in this paper has been realized by using the OpenAI Gym [47] interface of the ACTB. OpenAI Gym was created to standardize research in the field of artificial intelligence (AI) and RL, as well as for benchmarking and comparing algorithms easily. The framework provides the benefits of providing customizable realistic building environments to the AI research community without having to develop the physical system of the environment alone. Here, the Gym interface for the ACTB has flexibility in selecting the control actions, the states used for observation, as well as to formulate the rewards which allocate importance to a different objective like energy consumption and thermal comfort. Both supervisory set-point and low-level controls can be implemented through the ACTB. These controls of the HVAC systems in the ACTB are designed to follow ASHRAE Guideline 36 when not overridden with an external controller script. Here, supervisory set-point controls are implemented.

Points of departure from previous literature & contributions. Few studies on RL control applications on buildings rely on a realistic and high-fidelity building environment. Instead, the RL agents are usually tested on simplified state space models which are not able to simulate the nonlinear dynamics of a real building, and the control task might be overly simple to solve by the RL agent. This research addresses this drawback by using the Advanced Control Test Bed (ACTB) to test RL controls on a realistic Spawn of EnergyPlus small office building. Additionally, most of the literature utilized a RL pre-training with a similar building environment before being deployed in a target building environment. In this approach, this early pre-training developmental work can be avoided as the RL agent starts by mimicking a rule-based

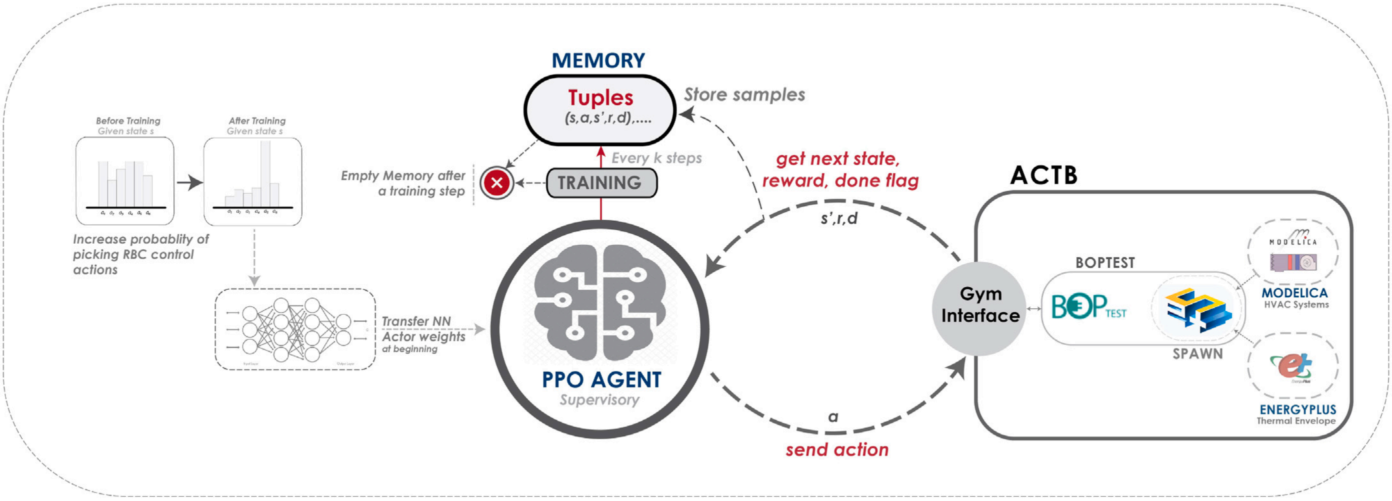


Fig. 2. Imitation Learning from RBC with PPO.

policy that is well accepted by the building community and control engineers although it is sub-optimal. Moreover, this does not require model development or previous actual building data. It can be deployed instantly to a real building where the RL agent starts by following the imitated rule-based policy. Subsequently, it assesses if this imitated policy is optimal for the multi-objective goals defined for the control problem and improves depending on the real feedback received from the actual building environment.

4. Case study

Description of the building environment. The building model used here is a Spawn model of the U.S. Department of Energy's (DOE) Reference Small Office Building, which features four perimeter zones and one core zone in a single story with a floor to ceiling height of 3 m. The floor area of the building is 511 m^2. The aspect ratio of the building is 1.5 with the longer side aligned on the east–west axis, and the glazing fraction is 0.21. The envelope consists of a mass wall (continuous insulation wall) which has a U-value of 0.857 $\mathrm{W/m^2C}$, and the windows have U-values of 3.23 $\mathrm{W/m^2C}$ with a solar heat gain value of 0.39. The light density of the building is 10.76 $\mathrm{W/m^2}$.

The HVAC systems of this building consists of several constant air volume air handling units (AHUs), composed of a gas-fired heating coil, a single-stage direct expansion cooling coil, an outside air damper without economizer, and a constant volume fan. The five thermal zones are each supplied by a packaged roof-top AHU, having a total of five AHUs in the building. The building is located in Chicago, IL, USA. A summer cooling control application is implemented here to demonstrate the benefits of this approach but this approach is not limited to summer cooling and can be applied to winter weather conditions with a heating scenario. A schematic of the building is shown in Fig. 3.

The RL agent controls the single speed cooling coil to control the indoor air temperature. The objectives of the RL agent is reduce energy consumption, reduce thermal discomfort and participate in a demand response (DR) scenario with a target demand limit program. During the peak summer hours the whole building demand was found to be 20 kW. Although, the target demand limit is usually set by the utility provider here we assume that during the DR event the building is expected to shed 25% of its peak load. Thus we take the target demand limit for the building to be 15,000 W and the controller incurs a penalty if the whole building energy exceeds the target demand limit. The DR event usually takes place in the afternoon between 2:30 p.m. to 3:30 p.m., and can last between 2 to 2.5 h.

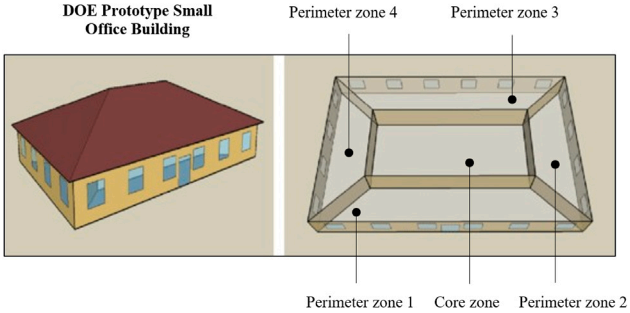


Fig. 3. Layout of the DOE Reference Small Office Building.
Source: [48].

Approach. This application demonstrates a novel approach of using imitation learning before the agent is transferred to the Spawn of EnergyPlus model, the latter representing to the extent possible a real building. The PPO agent essentially has two feed forward neural networks structures, the actor and the critic sharing the same network parameters except for the output layer. The actor outputs the action to take and the critic outputs the estimated value function of the state. Here the actor-network is trained in a supervisory fashion to learn to imitate the actions of the supervisory rule-base actions. The structure of the actor-network θ_{act} used was $[4, 600, 700, 1200, 1000, 800, 750, 3]$, while the critic network θ_{cri} had $[4, 600, 700, 1200, 1000, 800, 750, 150, 1]$. The training was initially started with a smaller and a shallow neural network. Through trial and error, a large dimension of the neural network was selected as this helped to reduce the loss in the imitation learning training to map states to rule-based actions.

This is essentially a supervisory learning process, in which the actor NN is trained to map input states to actions which are the rule-base controls set-point. The occupancy schedule of the building for a summer weekday is shown in Fig. 4.

There are five PPO agents acting, one on each of the thermal zones. Each PPO actor network is provided with the trained imitated policy. After interacting with the Spawn model and getting real feedbacks in the form of rewards, the PPO agent modifies its actor network for receiving higher scores from the building environment.

States considered. The states considered for the feedback of the RL agent in Zone i are:

- T_{zi} - Current temperature of zone i.

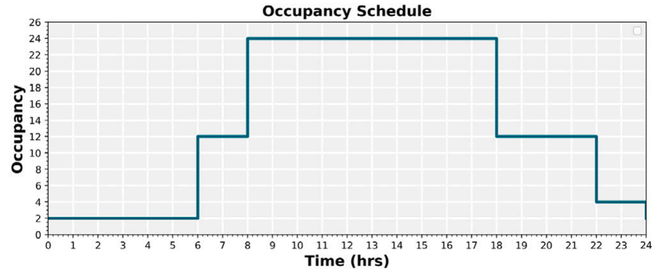


Fig. 4. Occupancy schedule.

- t_h - Current time in hours.
- δ_{sp} - The deviation of the previous temperature set-point from the rule-based temperature setpoint.
- DR_0 - Time remaining in hours from the DR event.
- DR_1 - A binary signal indicating if the current time falls in the DR event for the day.

DR_0 is a countdown time signal to the DR event. During and after the DR event, this state returns 0. The binary DR_1 signal is 1 during the DR event and 0 during other times. The states were normalized between 0 and 1, by a min–max normalization except δ_{sp} which was normalized between −1 and 1. The final state considered was $[T_{zi}, t_h, \delta_{sp}, DR_0, DR_1]$.

Actions. The RL controller can take three supervisory setpoint actions (a_t) at each step. The agent can increase or decrease the setpoint by 0.5 °C or keep the same setpoint ($a_t \in \{\pm 0.5\ °\mathrm{C}, 0\ °\mathrm{C}\}$). Here, the simulation step time is taken to be 5 min. An action constraint was used where the supervisory temperature setpoint can take any setpoint within the bounds of [18 °C, 27 °C] during the occupied hours and [15 °C, 31 °C] during the unoccupied hours. Here, the constraint implemented is a weak enforcement, which means that if the controller outputs any supervisory setpoint outside the mentioned range, the actions are overridden to project the value to the extreme bounds. Each of the five RL agents has the same setup for the controller action.

Reward formulation. The reward formulation for individual zone controller is the same and is shown in Eq. (4). Each zonal controller receives the feedback/rewards resulting from the thermal discomfort, energy consumption of the zone controlled, and the power penalty of the whole building. The power penalty term for the whole is included in the rewards terms for individual controller only if it is utilizing a nonzero cooling power during the particular time step. These objectives in the reward function are adapted to a monetary reward objective by assigning a dollar amount to them. The price for thermal discomfort is based on the assumption of a desk job salary in Chicago and its relation to the reduced productivity of the employees. This means the penalty for thermal comfort in the building is proportional to the occupancy of the zones. The thermal comfort bound is between 21 °C and 24 °C during the occupied hours of 6:00 a.m. to 10:00 p.m. and between 15 °C and 30 °C during the remaining unoccupied hours of the day.

Since the building is an office building with white-collar jobs, the average salary of the employees was considered to be slightly higher than average than the average Chicago salary. Here, the salary was assumed to be $80,000/year which is approximately $40/h. From [49], it was assumed for a 1 °C temperature rise, the productivity decreases by 2%. This resulted in a thermal discomfort cost of $0.8/Kh. The average cost for commercial electricity is taken to be $0.0405/kWh [50] and the demand charge was taken to be $7.89 per kW of power demand limit violation.

When the KPIs have converted to monetary penalties, the thermal discomfort price is much higher when compared to the energy price. Thus, the linear w_i weights are included in the penalty objective to scale the monetary penalties such that all KPIs have similar importance in the objective function. The linear weights w_1, w_3 and w_5 are negative as these are penalty costs that the agent aims to minimize.

$$Reward(r) = w_1 p_{disc} T_{disc}^{w_2} occ + w_3 p_e E^{w_4} + w_5 p_{dr} (P - P_t)^{w_6} \tag{4}$$

where:

T_{disc} = Thermal Discomfort Penalty per step [Kh]
E = Energy Consumption per step [kWh]
P = Whole Building Power per step [kW]
P_t = Whole Building Power Threshold during DR Event [kW]
w_1 = Linear hyper-parameter for T_{disc}
w_2 = Exponential hyper-parameter for T_{disc}
w_3 = Linear hyper-parameter for E
w_4 = Exponential hyper-parameter for E
w_5 = Linear hyper-parameter for Power Penalty
w_6 = Exponential hyper-parameter for Power Penalty
p_{disc} = Price for Thermal Discomfort [$/Kh]
p_e = Linear hyper-parameter for Power Penalty [$/kWh]
p_{dr} = Price for Demand Charge [$/kW]
occ = Occupancy [-]

The hyper-parameter weights w_i for the reward formulation chosen in this problem are $w_1 = -100$, $w_3 = -1$, $w_5 = -2$ and $w_2 = w_4 = w_6 = 1$, during the DR event and $w_1 = -200$, $w_3 = -(100 + 3000s)$, $w_5 = 0$ and $w_2 = w_4 = w_6 = 1$, during other times of the day. s is a scaling factor used based on the outside air temperature T_{oa}. If $T_{oa} > 25$ °C, the scaling factor s is $1/(T_{oa} - 25)$ and for $T_{oa} \leq 25$, the scaling factor s is equal to 1. This scaling factor discourages the RLC controller from over-cooling the indoor air temperature by penalizing heavily the use of excessive cooling energy when the outside air temperature is low. The setup of these variable weights avoids considering the outside air temperature as part of the observation states s_t reducing the complexity of the problem. Zone i controller receives the reward formed from the thermal discomfort and energy consumption of Zone i, but shares the power penalty term of the whole building. The whole building demand limit threshold P_t during the DR event is taken to be 15,000 W.

Artificial dataset generation. An artificial dataset is created following rule-based actions. The actor-network of the PPO agent is trained on this artificial dataset to mimic the rule-based actions, which consist of the following:

- The supervisory setpoint action can be incremented, decremented, and kept unchanged every 5 min by 0.5 °C.
- During normal operations (no DR event) of a day, the cooling coil follows a set-back supervisory temperature control strategy. During the occupied hours, the setpoint is 24 °C while during unoccupied hours the set-point is 30°, which are the higher limits of the thermal comfort bound.
- The supervisory set-point action has bounds between 18 °C and 27 °C, during the occupied hours and between 15 °C and 31 °C during the unoccupied hours.
- Pre-cooling is done by lowering the setpoint until 18 °C, two hours before a scheduled DR event to avoid the higher thermal discomfort and power costs during the DR event.
- During the DR event, the supervisory setpoint is increased up to 27 °C to reduce the demand charge costs.
- To avoid the early thermal discomfort from transitioning from the unoccupied to the occupied setpoint, a cooldown time is included for 1.5 h.

The occupied hours is between 6 a.m. and 10 p.m. The building has an on–off controller for the cooling coils in all five zones. The dataset for imitation learning is created artificially in the form of states and actions without any formulation of any rewards or consequences of the action taken. For example, if the hour of the day in the state space is 9:00 a.m. (occupied hours) and the previous set-point is 30 °C, the action would be to decrease the set-point by 0.5 °C to move towards the rule-based supervisory set-point action of 27 °C. A large artificial dataset is created by randomly starting with a initial room temperature (T_{zi}), the time of the day (t_h), previous set-point to form δ_{sp} and the DR signals (DR_0, DR_1) based on the time of the day and the DR event. The DR event occurs between 2:30 p.m. to 3:30 p.m. lasting for a duration between 2–2.5 h similar to the DR assumptions mentioned previously. The actions are also generated based on the rule-based conditions mentioned above. A large dataset was generated amounting to roughly four years' worth of artificial tuples of (states, action) so that most of the rule-based actions mentioned above are covered in this data-set. The actor-network is trained on this four years' worth of data to replicate the actions of the rule-based controls.

The advantage of creating this dataset is that it covers and explores a large part of the state space. Thus, when the trained actor-network is passed to the RL agent, if the RL agent wanders far off the rule-based action it has a higher probability of moving back towards the rule-based action in the initial stages. If the consequences of moving further from the rule-based action end up being positive, then the RL agent will move the policy away from this rule-based action after a training update. Thus, the imitation learning approach helps to reduce the RL controller agent to wander too far off the accepted rule-based control, preventing unstable and erratic behavior of an RL agent in the early training stages.

The rule-based policy is shown in Fig. 5 showing the temperature response of one of the zones and an average response of all the zones. The upper two plots represent the west zone temperature and power plots. The other four zones have very similar response to the same rule-based controls as that shown in Fig. 5. The bottom plot shows the total power consumption of the HVAC system for all five zones.

The performance of the rule-based policy is shown in Fig. 6. The upper two plots shows the temperature and HVAC power consumption of the west zone. Here only one out of the five zones is shown as the other zones follow the same rule based policy and have a very similar response. The lower two plots show the responses of the whole building. The third plot presents the mean temperature response of the all the zones with the blue line, while the lighter blue patch represents the minimum and maximum temperature of all the zones. The lowest plot displays the total HVAC and the whole building power of the entire building. The pink band indicates the time of the DR event.

Imitation training of the actor network. In the case for a direct application of an untrained RL agent to the building environment, the NN training starts with no prior knowledge. The weights of the NN, in this case, are usually randomly initialized within the range of [−1,1]. During the supervised training for imitation learning, a min–max constraint was applied to avoid large weights. Here, the weights were limited within the range of −5 and 5. Having large weights in the actor NN does not help with the PPO training, as the actor NN was either found to forget what it learned in the supervised stage or caused unstable behavior after some training updates. Constraining the weights helped to avoid these issues. When limiting the weights to smaller bounds such as −1 and 1, the NN was not able to reduce the loss function in supervised learning. Thus, selecting a wider range of [−5, 5] helped to reduce the loss function in the supervisory training. These values of [−5, 5] were chosen by trial and error by slowly increasing the bounds starting from [−1, 1] in increments of ±0.5 and stopping if this reached an acceptable accuracy within a certain number of training epochs. Here, the supervisory learning was performed on the dataset for a total of 300 epochs, with 800 batch samples per epoch of training. The NN weights were accepted if the validation accuracy was above 90%. If the NN failed to reach this accuracy within 100 epochs, the min–max bounds were increased. The training was done for another 200 epochs where it reached an accuracy of 93%.

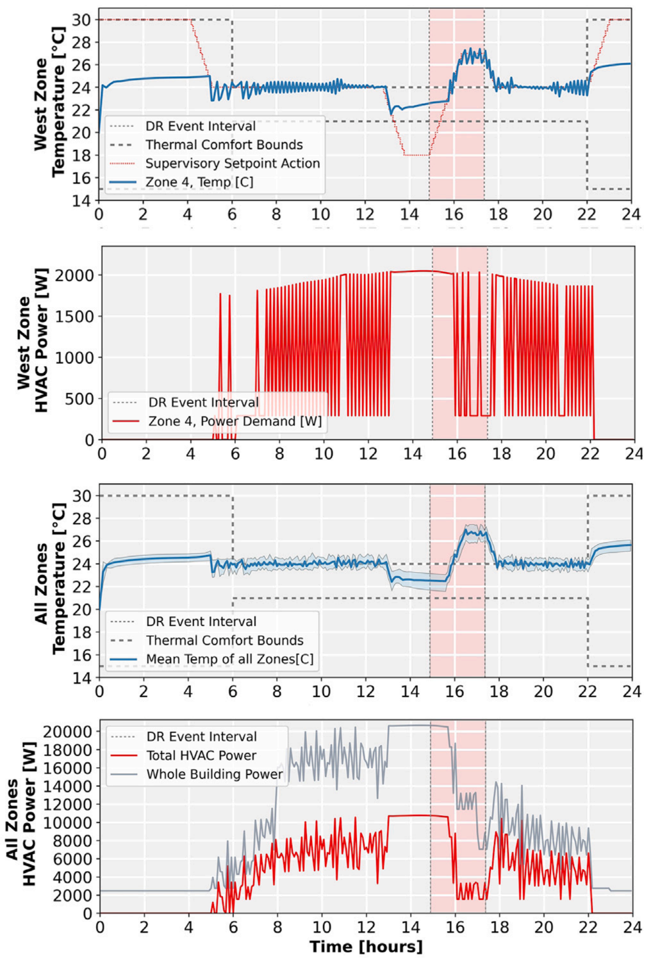


Fig. 5. Figure showing the temperature and power response of the west zone and all zones with the rule-based policy.

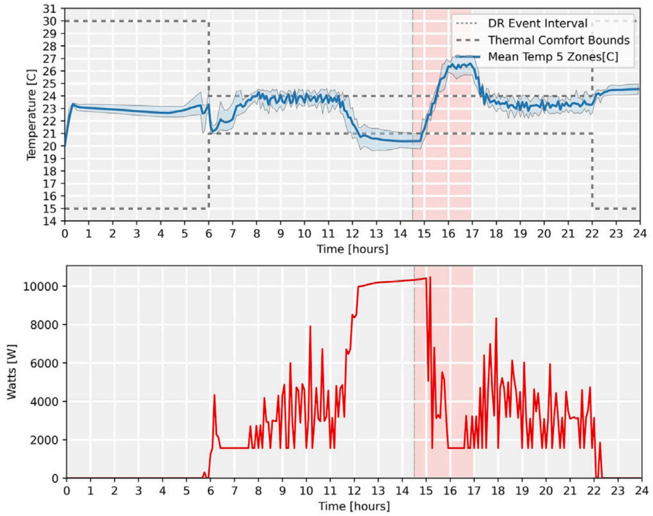


Fig. 6. Example of an Imitated policy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

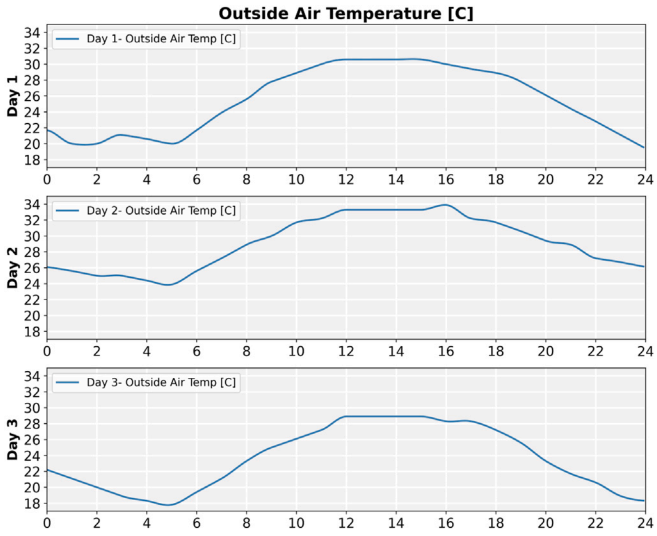


Fig. 7. Figure providing an example of the different.

Accuracy is measured by comparing the action output of the trained NN and the action coming from the artificial data. For example, if in a training batch, out of 100 samples, the NN outputs an action to move or keep the set-point at or towards the rule-based set-point action for more than 90 samples it would have an accuracy of more than 90%. In this case, the supervisory training achieved a steady 93% accuracy at the last 100 epochs of batch training. The learning rate for the training was reduced by a factor of ten after every 100 epochs and was started with a learning rate of 0.0001. The training was conducted using the Tensorflow [51] platform with Adamax optimizer from the Keras [52] library.

Points of departure from previous literature & contributions. Few studies on RL control applications on buildings rely on a realistic and high-fidelity building environment. Instead, the RL agents are usually tested on simplified state space models which are not able to simulate the nonlinear dynamics of a real building, and the control task might be overly simple to solve by the RL agent. This research addresses this drawback by using the Advanced Control Test Bed (ACTB) to test RL controls on a realistic Spawn of EnergyPlus small office building. Additionally, most of the literature utilized a RL pre-training with a similar building environment before being deployed in a target building environment. In this approach, this early pre-training developmental work can be avoided as the RL agent starts by mimicking a rule-based policy that is well accepted by the building community and control engineers although it is sub-optimal. Moreover, this does not require model development or previous actual building data. It can be deployed instantly to a real building where the RL agent starts by following the imitated rule-based policy. Subsequently, it assesses if this imitated policy is optimal for the multi-objective goals defined for the control problem and improves depending on the real feedback received from the actual building environment.

The training and testing process considers 31 random summer weekdays, where each episode is a day. The training and the testing split is done roughly in the ratio of 5:1, where the training split had 25 days and the testing split had 6 days. The training was conducted over 100 episodes of a day where each episode of a day is picked randomly from the 25 summer weekdays set aside for training. Since some days are more challenging than others due to differences in the outside air temperatures, there is a high variance in the scores achieved by the agent. An example of the difference in the outside air temperature during the different dates of training is provided in Fig. 7.

The progress of the training process is shown in Fig. 8, with the upper plot showing the 7-day mean cost achieved per episode and the lower showing the 7-day mean scores. A 7-day mean cost is shown as this reduces the variance of the scores achieved and makes it easy to portray the weekly progress in the graph. The horizontal dashed lines exhibit the 3-week average cost and scores at the beginning and end of the training process. The upper plot conveys a decrease of 50$ average cost per day of training while the lower plot conveys an improvement of 2,000 average rewards at the end of 100 days of training. The plot shows a trend of decreasing costs with interaction from the building with the RL agent modifying the imitated policy slowly to achieve higher goals as set by the control engineer. The average cost per day in the training period has reduced approximately from $800 to $750 at the end of a training period of 100 days. Fig. 12 showcases the performance of the agent at the end of 100 training days starting with the imitation policy.

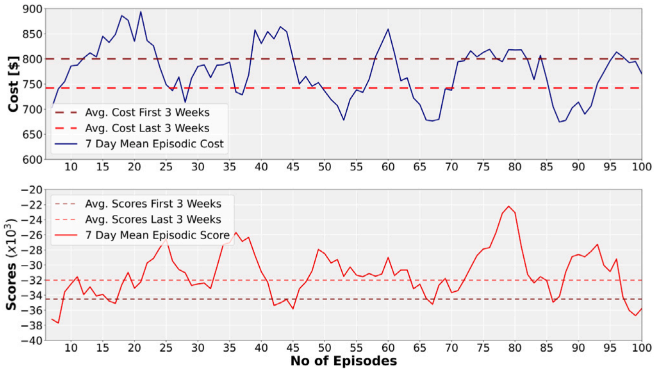


Fig. 8. Progress during training.

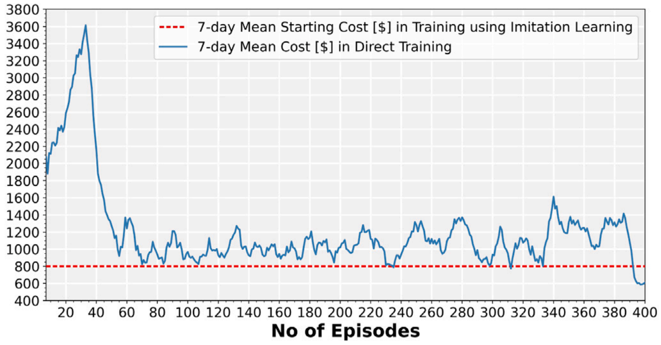


Fig. 9. Number of training days required by the agents to reach the average performance of the RL agent starting with imitated policy.

The same problem is also trained without any imitation learning to compare the benefits of using the imitation learning with a **direct** RL training approach without any imitation learning. An example of the performance of the untrained agent with a direct RL training approach is shown in Fig. 10. This exploratory behavior in the initial training phases in a real building can be avoided with the imitation learning approach. The training is continued for 400 days of summer. Fig. 11 shows the performance of the five agents near the end of the 400-day training period. The agents still have room for improvement and have the potential to learn a better policy with further training. Almost all of the five agents have learned to raise the supervisory setpoint temperature during the DR event to avoid the power penalty and some agents have started showing signs of pre-cooling to incur less of the thermal penalty cost associated with raising the setpoint during the DR event.

Fig. 9 shows the number of training days the agent took to reach the performance of the RL agent using an imitation learning agent. As

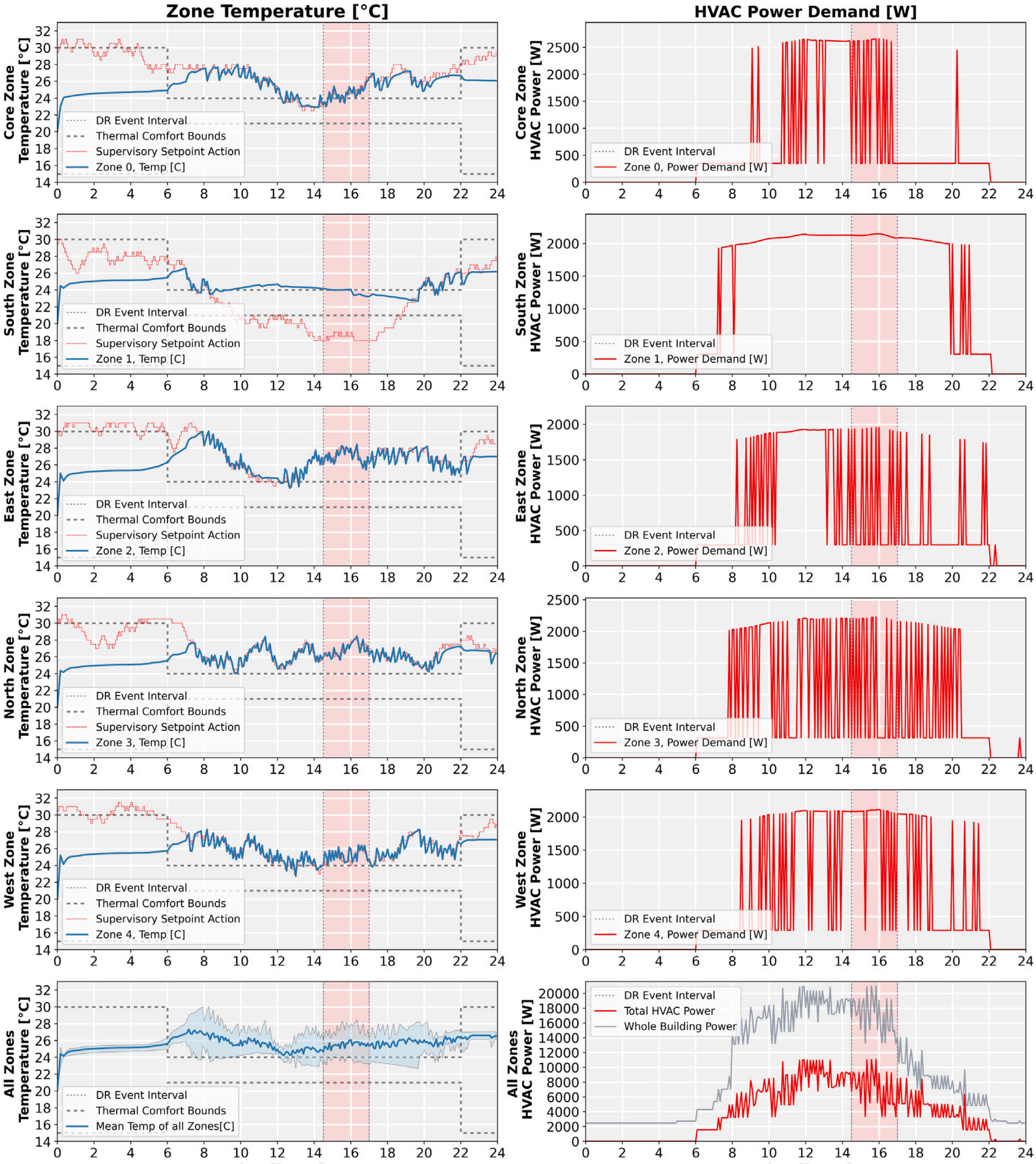


Fig. 10. Example of an untrained agent during initial episodes in direct RL training.

is evident from the figure, the RL agent using normal training took more than a year's worth of summer training days to reach the level of performance of the RLC agent using imitation learning. In real-world applications, this is equivalent to four years assuming each year has roughly 90 days of summer. This shows how this imitation learning approach avoids the large training time it takes the agent to learn an effective control strategy. Please refer to Section 6 for an in-depth discussion of the benefits of utilizing this approach.

***Testing the trained policy*.** During the testing of the trained agent, the DR event was randomized to a time between 2 p.m. and 3 p.m. and lasted for 2 to 2.5 h. Both the rule-based controller (RBC) and the reinforcement learning controller (RLC) were tested on the same summer days with the same DR conditions (start time and end time).

5. Results

The test results for the six summer days are provided in Table 1. This shows a 6.3% reduction in average cost and an average improvement of 7.2% in scores over the six test days with the imitated learning approach compared to the baseline rule-based heuristic policy. The trained RL policy reduces the thermal discomfort cost by increasing the probability of taking a slightly lower temperature setpoint to avoid thermal comfort costs but accepting the energy consumption costs for having a slightly lower setpoint temperature. The modified policy has also slightly increased the pre-cooling time from the rule-based policy. This is evident from Fig. 2.

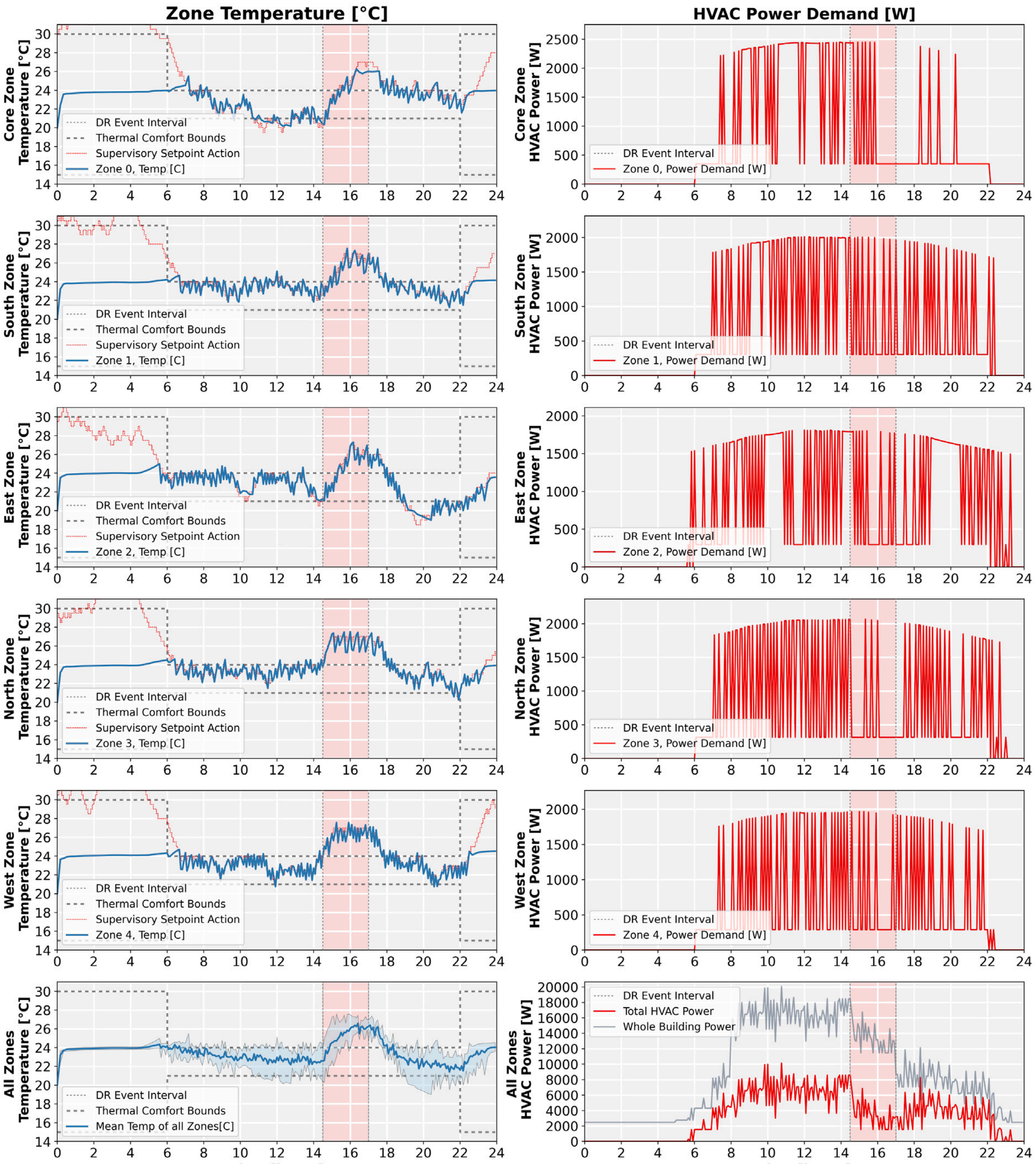


Fig. 11. Example of a Trained agent after 400 days of training with the direct approach without any imitation learning.

Table 1
Comparison of results.

Mean cost ($)	E	T_{disc}	Cost	Scores ($\times 10^3$)
Baseline (RBC)	8.32	483.85	781.17	−27.16
RLC w/Imitation	8.92	431.54	724.73	−25.52

6. Discussion and conclusion

This approach presents the opportunity of using RL building controls without any model development and avoiding the early unstable training period which could last for a long time before the RL agent learns something useful. Moreover, this approach starts with a rule-based policy commonly accepted among building control managers. This makes it easier for real-world implementation and RL acceptance, where a building manager may be willing to try out an RL approach as it can start with a policy set by the building manager, making improvements to the policy over time. As a consequence, this approach may address the hesitancy associated with the adoption of advanced building controls by building managers. In this study, the results show that the approach without using imitation learning took about 400-days of summer to reach to a performance comparable to a rule-based control. In real-world applications, the building manager and occupants cannot afford to wait for four years worth of summer data to learn a good policy whose performance is equivalent to a rule-based controller.

This study showcases an application where the rule-based control development was easy and intuitive, leading to a near-optimal rule-based solution. In this study, the rule based controls set supervisory

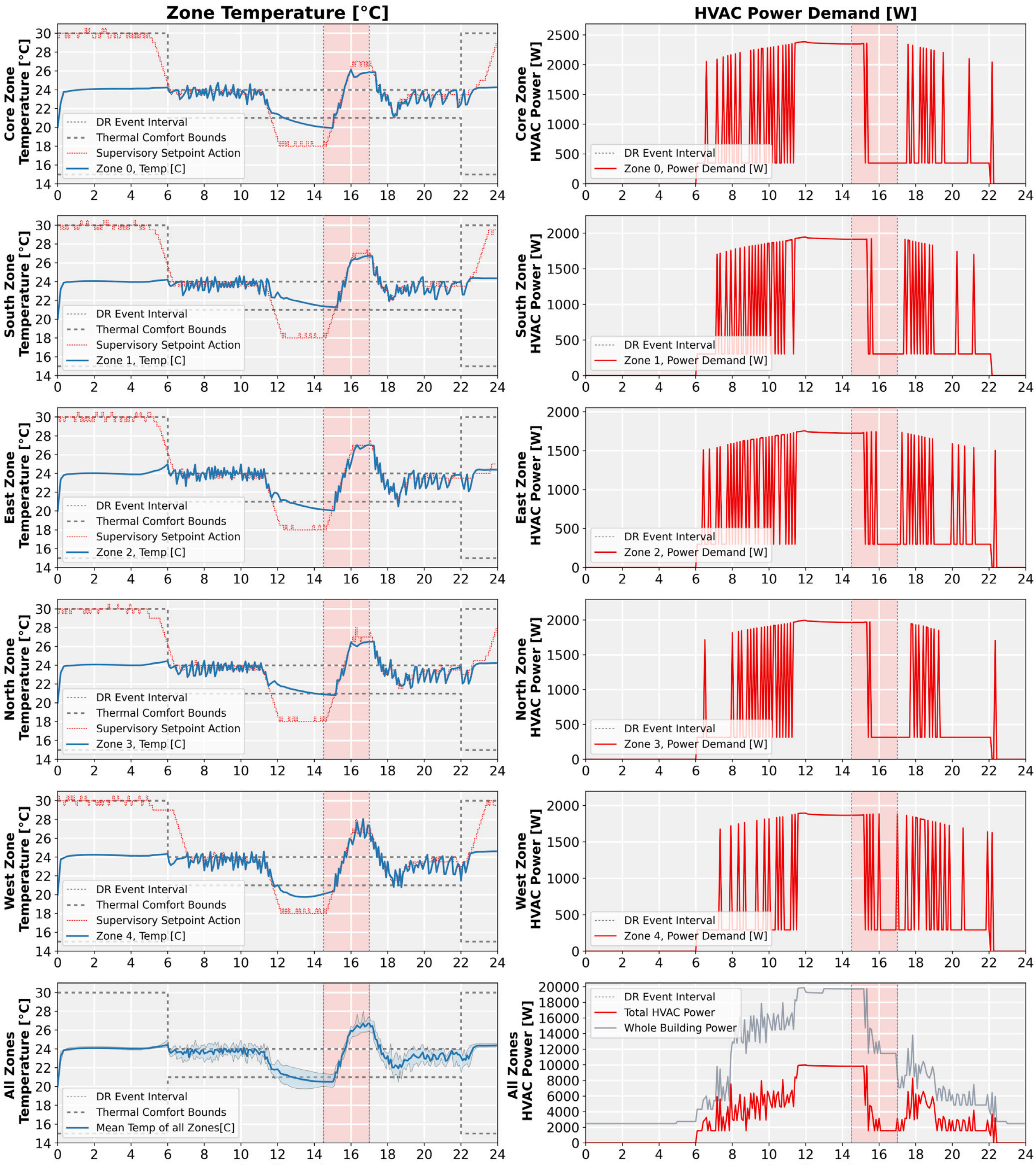


Fig. 12. Performance near the end of training 100 episodes starting with the imitated policy.

set-points near the upper-comfort bound for a summer day to optimize for thermal discomfort, energy consumption, and target demand limit. The rule-based strategies involved setting up supervisory set-points near the upper human comfort bounds, pre-cooling the building, and increasing the set-point during the demand response event. These policies are intuitive to an extent and a strategy can be developed which can fairly do well. However, for further savings and benefits, the RL has the potential to find better solutions and find more specific solutions to questions like how many hours before the DR event should pre-cooling commence that is more adapted to this particular type of environment, the specific supervisory set-point that gives the best trade-off between thermal comfort and energy consumption, the cool-down time needed for the changeover from a larger comfort band to a narrow comfort band. We can start with an educated guess of the control sequences or follow the recommended and widely accepted generalized guidelines, and then let the RL agent figure out the best solution improving on this depending on the feedback from the real building and environment. The RL approach is able to find control rules that are more

adapted to a particular environment and the multi-objective optimization
goals.

However, when it comes to complex building environments with a lot of conflicting objectives, the development of a good rule-based control becomes difficult as they tend to become less intuitive and their performance might be poor. Although they may be sub-optimal, the RL can utilize this heuristic rule-based policy and continue to adapt and improve upon this policy. In this study, the objectives were limited to energy consumption, thermal discomfort, and power target demand response. With growing needs in the future, the multiple objectives can be extended to include other objectives like solar PV utilization, battery control, indoor air quality, and other services like vehicle-to-grid (V2G). MPC is difficult to utilize as model development and continued training for a complex environments is challenging and requires significant time to develop. Moreover, they are not scalable to similar building environments. RL in this aspect do not require any model development and is scalable to similar building environments.

Future work would investigate this approach with a more complicated building environment to determine the potential benefits in doing so. Here, a PPO agent was used which is a sample inefficient online algorithm (all the samples are forgotten after a training update). Moreover, sometimes PPO suffer from training instability. For future research, an offline algorithm like soft-actor critic (SAC) is expected to offer better results in terms of training speed and performance as it can re-utilize all the past experiences for the training update.

7. Limitations and future research

The limitation of this research is that this approach may not be applicable for pure value-based RL algorithms like DQN, which do not have a policy network. Second, after imitation learning with artificial data, there is no established strategy to tune the learning rate parameter to improve upon this learned policy by interaction. A high learning rate can lead to unstable behavior, as the policy may deviate too quickly from the imitated policy before correct value estimates are learned. Conversely, an extremely low learning rate may result in no policy development at all. Therefore, an effective tuning strategy for the learning rate is necessary. In this research, we adopted an ad-hoc approach and used a learning rate for the policy network that was one-hundredth of the learning rate of the baseline RL learning strategy for the first ten episodes and one-tenth for subsequent training episodes. To address this limitation, further research is needed to develop a more effective and generalizable strategy for tuning the learning rate in imitation learning with artificial data. Such a strategy could potentially improve the stability and learning progress enabling the application of this technique in a wider range of RLC in building controls.

Exploring the topics of explainability and transparency in reinforcement learning (RL) has become increasingly important as it relates to understanding why an RL agent made specific decisions or took certain actions. This is especially critical in high-stakes scenarios such as autonomous driving and healthcare, where humans need to trust and evaluate the decisions made by RL agents. While imitation learning as shown in this research, is inherently interpretable as it learns from an existing rule-based policy, which is both intuitive and interpretable, the RL agents modify this policy over time to achieve desired goals. If the rule-based policy is far from optimal, the resulting policy may deviate significantly from the imitated policy, which could require further explanation and interpretability from control engineers. Therefore, research into explainable AI (XAI) is crucial to achieve wider adoption of RL to real-world problems beyond building controls.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was authored in part by the National Renewable Energy Laboratory, United States, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Building Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

[1] Richter B, Crabtree G, Glicksman L, Goldstein D, Goldston D, Greene D, et al. Energy future: Think efficiency. 2008, p. 1–109, URL http://www.aps.org/energyefficiencyreport, unpublished.

[2] Tyra B, Cassar C, Liu J, Wong P, Yildiz O. Electric power monthly with data for November 2020. (January). 2019.

[3] Department of Energy (DOE). Chapter 5: Increasing Efficiency of Building Systems and Technologies. (September):2015, p. 143–81, URL https://www.energy.gov/sites/prod/files/2017,

[4] Pérez-Lombard L, Ortiz J, Pout C. A review on buildings energy consumption information. Energy Build 2008;40(3):394–8. http://dx.doi.org/10.1016/j.enbuild.2007.03.007.

[5] Klepeis NE, Nelson WC, Ott WR, Robinson JP, Tsang AM, Switzer P, et al. The National Human Activity Pattern Survey (NHAPS): A resource for assessing exposure to environmental pollutants. J. Expo. Anal. Environ. Epidemiol. 2001;11(3):231–52. http://dx.doi.org/10.1038/sj.jea.7500165.

[6] Roth A, Reyna J. Grid-Interactive Efficient Buildings Technical Report Series: Whole-Building Controls, Sensors, Modeling, and Analytics. (December). 2019, URL https://www.energy.gov/eere/buildings,

[7] Chen B, Cai Z, Bergés M. Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable MPC policy. In: BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. 2019, p. 316–25. http://dx.doi.org/10.1145/3360322.3360849.

[8] ASHRAE. ASHRAE guideline 36-2018 high-performance sequences of operation for HVAC systems. 2018.

[9] Richalet J, Rault A, Testud JL, Papon J. Model predictive heuristic control. Applications to industrial processes. Automatica 1978;14:413–28. http://dx.doi.org/10.1016/0005-1098(78)90001-8.

[10] Drgoňa J, Arroyo J, Cupeiro Figueroa I, Blum D, Arendt K, Kim D, et al. All you need to know about model predictive control for buildings. Annu Rev Control 2020;50(May):190–232. http://dx.doi.org/10.1016/j.arcontrol.2020.09.001.

[11] Wang Z, Hong T. Reinforcement learning for building controls: The opportunities and challenges. Appl Energy 2020;269(February):115036. http://dx.doi.org/10.1016/j.apenergy.2020.115036.

[12] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33. http://dx.doi.org/10.1038/nature14236, URL http://dx.doi.org/10.1038/nature14236.

[13] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature 2016;529(7587):484–9. http://dx.doi.org/10.1038/nature16961, URL http://dx.doi.org/10.1038/nature16961.

[14] Pomerleau DA. Alvinn: An autonomous land vehicle in a neural network. Adv Neural Inf Process Syst 1988;1.

[15] Levine S, Finn C, Darrell T, Abbeel P. End-to-end training of deep visuomotor policies. J Mach Learn Res 2016;17(1):1334–73.

[16] Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int J Robot Res 2018;37(4–5):421–36.

[17] Weiss K, Khoshgoftaar TM, Wang DD. A survey of transfer learning. vol. 3, Springer International Publishing; 2016, http://dx.doi.org/10.1186/s40537-016-0043-6.

[18] Lissa P, Schukat M, Barrett E. Transfer learning applied to reinforcement learning-based HVAC control. SN Comput. Sci. 2020;1(3):1–12. http://dx.doi.org/10.1007/s42979-020-00146-7.

[19] Lissa P, Schukat M, Keane M, Barrett E. Transfer learning applied to DRL-Based heat pump control to leverage microgrid energy efficiency. Smart Energy 2021;3:100044. http://dx.doi.org/10.1016/j.segy.2021.100044.

[20] Taylor A, Dusparic I, Gueriau M, Clarke S. Parallel Transfer Learning in Multi-Agent Systems: What, when and how to transfer? Proc. Int. Joint Conf. Neural Netw. 2019;2019-July:1–8. http://dx.doi.org/10.1109/IJCNN.2019.8851784.

[21] Zou Z, Yu X, Ergan S. Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. Build Environ 2020;168(November 2019):106535. http://dx.doi.org/10.1016/j.buildenv.2019.106535.

[22] Zhang X, Jin X, Tripp C, Biagioni DJ, Graf P, Jiang H. Transferable reinforcement learning for smart homes. In: RLEM 2020 - Proceedings of the 1st international workshop on reinforcement learning for energy management in buildings and cities. 2020, p. 43–7. http://dx.doi.org/10.1145/3427773.3427865.

[23] Spangher L, Gokul A, Khattar M, Palakapilly J, Agwan U, Tawade A, et al. Augmenting reinforcement learning with a planning model for optimizing energy demand response. In: RLEM 2020 - Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings and Cities. 2020, p. 39–42. http://dx.doi.org/10.1145/3427773.3427863.

[24] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[25] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: 35th International Conference on Machine Learning, ICML 2018. vol. 5, 2018, p. 2976–89, arXiv:1801.01290.

[26] Xu S, Wang Y, Wang Y, O'Neill Z, Zhu Q. One for many: Transfer learning for building HVAC control. In: BuildSys 2020 - proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. 2020, p. 230–9. http://dx.doi.org/10.1145/3408308.3427617, arXiv:2008.03625.

[27] Costanzo GT, Iacovella S, Ruelens F, Leurs T, Claessens BJ. Experimental analysis of data-driven control for a building heating system. Sustain. Energy Grids Netw. 2016;6:81–90.

[28] Huang GB, Zhu QY, Siew CK. Extreme learning machine: Theory and applications. Neurocomputing 2006;70(1–3):489–501. http://dx.doi.org/10.1016/j.neucom.2005.12.126.

[29] Tsang N, Cao C, Wu S, Yan Z, Yousefi A, Fred-Ojala A, et al. Autonomous Household Energy Management Using Deep Reinforcement Learning. In: Proceedings - 2019 IEEE international conference on engineering, technology and innovation, ICE/ITMC 2019. 2019, http://dx.doi.org/10.1109/ICE.2019.8792636.

[30] Deng Z, Chen Q. Reinforcement learning of occupant behavior model for cross-building transfer learning to various HVAC control systems. Energy Build 2021;238:110860. http://dx.doi.org/10.1016/j.enbuild.2021.110860.

[31] Mocanu E, Nguyen PH, Kling WL, Gibescu M. Unsupervised energy prediction in a Smart Grid context using reinforcement cross-building transfer learning. Energy Build 2016;116:646–55. http://dx.doi.org/10.1016/j.enbuild.2016.01.030, URL http://dx.doi.org/10.1016/j.enbuild.2016.01.030.

[32] Sutton, Richard S. Barto AG. Reinforcement learning : An introduction. The MIT Press; 2014, http://dx.doi.org/10.4018/978-1-60960-165-2.ch004.

[33] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18:1527–54.

[34] Tao Y, Qiu J, Lai S. A hybrid cloud and edge control strategy for demand responses using deep reinforcement learning and transfer learning. IEEE Trans Cloud Comput 2021;7161(c). http://dx.doi.org/10.1109/TCC.2021.3117580.

[35] Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T. Deep Domain Confusion: Maximizing for Domain Invariance. 2014, arXiv preprint arXiv:1412.3474, arXiv:1412.3474, URL http://arxiv.org/abs/1412.3474.

[36] Fan L, Zhang J, He Y, Liu Y, Hu T, Zhang H. Optimal scheduling of microgrid based on deep deterministic policy gradient and transfer learning. Energies 2021;14(3):1–16. http://dx.doi.org/10.3390/en14030584.

[37] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning - data mining, inference, and prediction. 2nd ed. Springer; 2017, http://dx.doi.org/10.1007/b94608, URL http://www.springer.com/series/692.

[38] Schulman J, Levine S, Moritz P, Jordan M, Abbeel P. Trust region policy optimization. In: 32nd international conference on machine learning, ICML 2015. vol. 3, 2015, p. 1889–97, arXiv:1502.05477.

[39] Cover TM, Thomas JA. Elements of information theory. Wiley-Interscience; 2005, p. 1–748. http://dx.doi.org/10.1002/047174882X.

[40] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, p. 1–12, arXiv arXiv:1707.06347.

[41] Bain M, Sammut C. A framework for behavioural cloning. Mach. Intell. 1999;15:103–29.

[42] Nagy Z, Nweye K. Real-world challenges for reinforcement learning in building control. 1, 2021, arXiv preprint arXiv:2112.06127.

[43] Marzullo T, Dey S, Long N, Vilaplana JL. A high-fidelity building performance simulation test bed for the development and evaluation of advanced controls. J. Build. Perform. Simul. 2022. http://dx.doi.org/10.1080/19401493.2022.2058091.

[44] Wetter M, Benne K, Gautier A, Nouidui T, Ramle A, Roth A, et al. Lifting the garage door on spawn, an open-source BEM-controls engine. In: 2020 building performance modeling conference and SimBuild. 2020, p. 518–25.

[45] Blum D, Jorissen F, Huang S, Chen Y, Arroyo J, Benne K, et al. Prototyping the BOPTEST framework for simulation-based testing of advanced control strategies in buildings. In: IBPSA international conference and exhibition. vol. 17, 2019, p. 2737–44.

[46] Benne K, Ball B, Bernal Heredia W, Cutler D, Frank S, Brackney L. Alfalfa. 2018, http://dx.doi.org/10.11578/dc.20181205.2, https://www.osti.gov/biblio/1484597.

[47] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. OpenAI Gym. 2016, p. 1–4, arXiv preprint arXiv:1606.01540, arXiv:1606.01540, URL http://arxiv.org/abs/1606.01540.

[48] Deru M, Field K, Studer D, Benne K, Griffith B, Torcellini P, et al. US department of energy commercial reference building models of the national building stock. 2011.

[49] Seppänen O, Fisk WJ, Faulkner D. Cost benefit analysis of the night-time ventilative cooling in office building. In: Proceedings of healthy buildings 2006. 2006, p. 243–7.

[50] Local E. Chicago, IL electricity rates | electricity local. 2012, URL https://www.electricitylocal.com/states/illinois/chicago/.

[51] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX symposium on operating systems design and implementation, OSDI 2016. 2016, p. 265–83, arXiv:1605.08695.

[52] Chollet F, et al. Keras. 2015, URL https://github.com/fchollet/keras.