Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-20 (2021) 907–912

Reinforcement Learning for Control of Passive Heating and Cooling in Buildings ⋆

Bumsoo Park, * **Alexandra R. Rempel, Alan K. L. Lai** ** **Julianna Chiaramonte, Sandipan Mishra** ***

* *Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: parkb5@rpi.edu).*

** *University of Oregon, Eugene, OR 97405 USA (e-mail: arempel@uoregon.edu, alanl@uoregon.edu)*

*** *Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: chiarj2@rpi.edu, mishrs2@rpi.edu)*

Abstract: Mechanical space heating and cooling are responsible for over one-third of the greenhouse gases released by building operations globally. As a result, heating and cooling load reductions are high priorities in climate change mitigation efforts. Direct solar heating, natural ventilation, and shading are often able to condition indoor spaces "passively" using only climatic resources, but their performance is limited by the lack of effective and affordable controls for their operable elements: rule-based control strategies cannot anticipate changes in weather or adapt to seasonal changes, while model-based strategies require significant investment into the creation of customized thermal models. Here, we design and validate a model-free data-driven reinforcement learning approach by comparing tabular Q-learning and policy-gradient (REINFORCE) algorithms for passive heating and cooling. These algorithms are trained on a residential building simulated in EnergyPlus in Albany NY and evaluated on the basis of unmet heating and cooling loads in both the training climate and six others. We find that the learned operation of shading, night insulation, and window aperture opening, driven by indoor and outdoor air temperatures, window surface heat flux, and weather forecasts, reduces total loads by 47-76% compared to operation without passive systems. Additionally, the REINFORCE policy reduces loads by 13-64% over conventional rule-based control, with one exception. Together, these results show that reinforcement learning can improve passive heating and cooling performance substantially, ultimately reducing space heating and cooling energy requirements.

Keywords: Building automation, Reinforcement learning, Neural networks, Data-driven control

1. INTRODUCTION

Mechanical space heating and cooling of buildings account for an estimated 4 Gt of greenhouse gas emissions annually, exceeding one-third of the building-related total worldwide (IEA, 2020). Despite advances in building energy codes, growth in built space is outpacing efficiency efforts, causing these emissions to continue to rise; the need to avoid mechanical heating and cooling altogether where possible is therefore urgent (IEA, 2020; Lucon et al., 2015). Direct solar heating, natural ventilation, shading, radiative cooling, and other passive strategies are of interest in this effort because they use solar radiation, cool night air, and other climatic resources to heat and cool building spaces without mechanical systems (e. g. Oropeza-Perez and Østergaard, 2018). However, reliable performance requires the coordinated operation of shading, movable insulation, and aperture openings. Typically, these are controlled by indoor and/or outdoor air temperatures, illumination, or time of day (e.g. Liu et al., 2015; Grynning et al., 2014), and while these have shown promising performance, extensive further improvements appear possible (Rempel and Lim, 2019; Chen et al., 2018).

Traditionally, building thermal control schemes have relied on empirically tuned PID or rule-based controllers, often with poor performance and energy inefficiency. To address this, model-based approaches such as model predictive control strategies (MPC) have become more prevalent in the past two decades. However, these methods require significant modeling and commissioning effort, as suggested by (Drgoňa et al., 2020). As a result, interest is growing in data-driven approaches, including iterative learning control (Minakais et al., 2019) and machine learning-based control (e.g. Peng et al., 2018). The latter includes *model-free* approaches such as reinforcement learning (RL) (Sutton and Barto, 2018) that use *high-fidelity simulation* models to learn control policies that maximize a cumulative reward function through repeated trial and error.

RL control strategies for indoor environments have primarily focused on mechanical heating, ventilating, and air-conditioning (HVAC) systems, with the goal of reducing energy consumption (or cost) and improving occupant comfort. To minimize the number of control points, such studies often adjust the end state, e.g. the room air temperature setpoint, as the only action, shifting heating or cooling loads to hours when electricity is less expensive. Actions focused on direct control of actuators, such as

⋆ This work was funded by National Science Foundation CBET-1804218.

10.1016/j.ifacol.2021.11.287

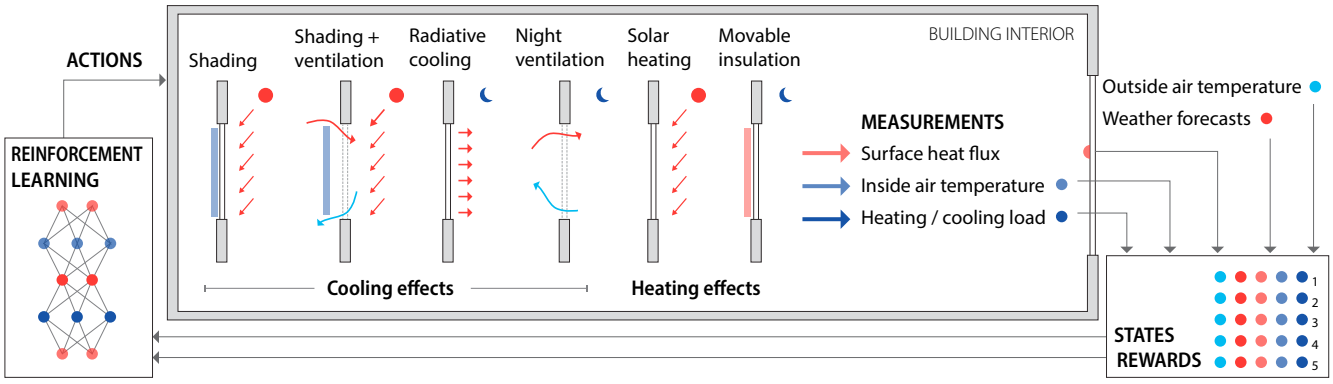

Fig. 1. Reinforcement learning (RL) for passive system operation. States reflect inside and outside air temperature, weather forecasts, window surface heat flux, and heating or cooling loads at each 10-min timestep (Table 1; Eqn. 3). Rewards prioritize indoor air temperatures between 20-25°C and reduced heating and cooling load. The control policy, learned either by tabular Q-learning or a policy-gradient algorithm like REINFORCE, generates actions that deploy or retract shading (for cooling); deploy or retract movable insulation (for heating); and open or close window apertures for ventilation with outside air.

economizer dampers, are seldom investigated (Wang and Hong, 2020).

The improvement of passive heating and cooling controls is also motivated by the desire to reduce energy use and preserve thermal comfort, resulting in analogous reward structures, but actions must address the actuator level, e.g. shading operation. Time-dependent building thermal behavior, driven by both weather and occupancy, also has greater importance in passive systems because pre-heating or cooling must be accomplished with variable climatic resources rather than predictable mechanical systems. Passive systems are therefore excellent candidates for RL because they must respond appropriately to multiple, potentially opposing environmental conditions, such as the co-occurrence of cold air and intense solar radiation, and because diurnal and seasonal thermal cycles cause the ideal responses to vary accordingly (Rempel and Lim, 2019).

However, few investigations of model-free RL for control of passive heating or cooling have been conducted to date and all have relied on tabular Q-learning to control a single operable element. For example, Cheng et al. (2016) developed policies to control office window blind slat angles to improve visual comfort and diminish lighting energy use, improving significantly upon traditional controls during field implementation. In related work, Chen et al. (2018) developed policies to control aperture opening for natural ventilation in the context of a simplified building heat transfer model, similarly finding noticeable reductions in cooling energy use over conventional heuristic control.

While tabular Q-learning methods are suitable for small state and action spaces, they do not scale well with increasing state and action complexity. To address this, we investigate a (policy-based) function approximation approach, REINFORCE (Williams, 1992), comparing it with Q-learning (Watkins and Dayan, 1992) for the development of control strategies for passive heating and cooling systems that minimize space-heating and cooling loads by leveraging climatic resources. We evaluate the performance of these algorithms in learning optimal control strategies, without prior knowledge of system dynamics, in terms of (1) training data required, (2) reduction of heating and cooling loads, and (3) robustness of the learned policies under contrasting conditions.

2. PROBLEM FORMULATION & PRELIMINARIES

2.1 Problem Statement

As a representative test case, we consider a dwelling with three controllable passive heating and cooling elements: window shading, movable insulation, and aperture opening. The goal is to find a feedback control strategy for these elements that minimizes mechanical heating and cooling loads using measurements of current indoor and outdoor air temperatures, heat flux across window surfaces, mechanical heating and cooling loads, and forecast outdoor air temperatures, *without directly knowing the dynamics of the system*, using an RL approach (Fig. 1).

2.2 Markov Decision Process

RL schemes are typically based on the formulation of a Markov Decision Process (MDP) consisting of states $S_t \in \mathcal{S}$, allowable actions $A_t \in \mathcal{A}$, rewards $R_t \in \Re$, and transition probabilities $P(S_{t+1}|S_t, A_t)$. The control law, commonly referred to as the policy, maps the states to actions $\pi : \mathcal{S} \to \mathcal{A}$ (or $\pi : \mathcal{S} \to \Pr(\mathcal{A})$). The goal of the RL algorithm is to find the optimal policy π^* (deterministic or stochastic) to maximize the expected cumulative reward, $G = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1}\right]$, given the reward (cost) function $r(\cdot)$. In a model-free learning scheme, where the transition probabilities of the MDP are unknown, the optimal policy is found through trial and error by interacting with the environment (Sutton and Barto, 2018).

2.3 Value-based RL

Given a policy π, $V^\pi(s)$ (the state value function) and $Q^\pi(s,a)$ (the action value function) represent the expected cumulative future rewards for a given state or state-action pair, explicitly written as:

$$V^\pi(s) \equiv \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right] \tag{1}$$

$$Q^\pi(s,a) \equiv \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]. \tag{2}$$

Above, $\mathbb{E}[\cdot]$ represents the expectation over the stochastic policy π, and γ is the discount factor that emphasizes immediate rewards.

RL algorithms that use value functions to determine the optimal policy are known as value-based methods. For these methods, the policy is simply the selection of the action that yields the highest value based on the value function (greedy policy), or alternatively the selection of a random action with a probability of ϵ (ϵ-greedy policy). During the training process, the value function is iteratively updated through the Bellman equation (Bellman, 1966) until it converges to the optimal value function.

Q-learning refers to a class of value-based temporal difference (TD) RL algorithms that use the action-value function $Q(s,a)$ to learn the optimal policy. While $Q(s,a)$ is (in general) a continuous function of s and a, it can be formulated into a tabular function if both the states and actions are finite and discrete. Function approximations can also be used to estimate $Q(s,a)$ directly through algorithms such as Deep Q-Networks (DQN) (Mnih et al., 2013). However, since smaller and finite state-spaces are easier to interpret, and because tabular methods have faster convergence, we first explore tabular Q-learning (Algorithm 1) for the application under study (Sec. 3.1).

Algorithm 1 Tabular Q-learning

Initialize $Q(s,a)$ arbitrarily
for episode = 1,...,M **do**
  Initialize s
  for each timestep of episode **do**
    Choose a given s using ϵ-greedy
    Take action a, observe r and s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma\ max_a Q(s',a) - Q(s,a)\right]$
    $s \leftarrow s'$
    until s is terminal
  end for
end for

2.4 Policy-based RL

Policy-based RL methods directly learn the optimal policy through a function approximator, parametrized by a set of basis functions and the vector $\boldsymbol{\theta}$, i.e., $\pi \equiv \pi_\theta$. One advantage of policy-based methods is their ability to learn stochastic policies with continuous state and action spaces. More importantly, initial policies can be pre-trained with samples from a pre-defined expert control, accelerating the training process and reducing random actions when deploying a model to a real-world system.

While policy-based schemes can use a variety of search methods such as finite difference or gradient ascent to optimize the parameters of the function approximator, gradient-based strategies are well-suited for larger parameter spaces. In policy-gradient (PG) methods, the objective is to update $\boldsymbol{\theta}$ based on the gradient of expected future rewards with respect to $\boldsymbol{\theta}$, maximizing the expected future rewards. Hence, the utility function $J(\theta)$, i.e., the performance measure of the policy with respect to $\boldsymbol{\theta}$ for an episodic case can be defined as the state-value function under policy π_θ as $J(\boldsymbol{\theta}) = V^{\pi_\theta}(s_0)$.

From the policy gradient theorem (Sutton and Barto, 2018), $\nabla_\theta J(\theta)$, can be written as

$$\nabla_\theta J(\boldsymbol{\theta}) = \mathbb{E}\left[G_t \nabla \ln \pi(A_t \mid S_t, \boldsymbol{\theta})\right]. \tag{3}$$

$\nabla \ln \pi(A_t \mid S_t, \boldsymbol{\theta})$ is known, and thus $\mathbb{E}[G_t]$ must be computed to use the analytical form of the gradient. We use REINFORCE, a Monte-Carlo PG algorithm that uses empirical rewards sampled from trajectories from a Monte-Carlo estimate instead of the actual expectation $\mathbb{E}[G_t]$. From this estimate of $\nabla_\theta J(\boldsymbol{\theta})$, we update $\boldsymbol{\theta}$ via gradient ascent. The pseudocode for REINFORCE is shown below.

Algorithm 2 REINFORCE: Monte-Carlo PG

Initialize parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$
for episode = 1,...,M **do**
  Generate trajectory $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following π_θ
  for each timestep of episode $t = 0, \ldots, T-1$ **do**
    $G_t \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G_t \nabla_\theta \ln \pi(A_t \mid S_t, \boldsymbol{\theta})$
    until $\gamma^t G_t \nabla_\theta \ln \pi(A_t \mid S_t, \boldsymbol{\theta})$ is small enough
  end for
end for

3. METHODOLOGY AND IMPLEMENTATION

3.1 Q-Learning MDP Formulation

As discussed in Section 2, tabular Q-learning requires the states and actions to be defined in finite and discrete spaces. The states should be defined to capture sufficient information for the controller to make appropriate decisions. Indoor and outdoor temperatures are central to decisions regarding natural ventilation, while knowledge of heat flux through windows is necessary for making decisions regarding movable insulation and shading status. Additionally, passive heating and cooling processes operate over time periods of many hours due to the thermal inertia of building materials, requiring forecast information so that the controller can pre-cool or pre-heat the building in advance of hot days or cold nights. Finally, an indication of the energy consumption status should be included in the states. These raw measurements are then discretized based on seven parameters $k_1, \ldots, k_7$ (Table 1).

Table 1. Q-Learning State Discretization

Parameter	Type	Value range
k_1	Indoor temperature trend	0,1,2
k_2	Indoor temperature (T^{in})	0,...,10
k_3	Outdoor temperature trend	0,1,2
k_4	Temperature difference ($T^{in} - T^{out}$)	0,...,7
k_5	Heat flux (q)	0,1
k_6	Forecast	0,1
k_7	Energy consumption status	0,1

Parameters k_1 and k_3 are determined from the trends of the past 10 measurements (collected every 10 minutes) of indoor and outdoor temperature, where 0 corresponds to an increasing trend, 1 corresponds to a decreasing trend, and 2 is assigned to any other pattern. Parameter k_2 is assigned a value based on 11 discrete bins for indoor air temperatures between 20 and 25℃, and the value of

k_4 is established by the difference between indoor and outdoor temperature within 8 discrete bins. Parameter k_5 is determined from the direction of the net heat flux through window glass, where 0 indicates outward heat flux and 1 indicates inward heat flux. Parameter k_6 is assigned a value of 1 if the maximum forecast outdoor temperature for the next 20 hours exceeds 26 ℃, and 0 otherwise. Finally, k_7 indicates the status of the mechanical heating and cooling energy requirement, where 0 indicates the absence and 1 indicates the presence of a heating or cooling load. The state of the system for Q-learning, $\boldsymbol{S_t^{QL}}$, is then expressed as:

$$\boldsymbol{S_t^{QL}} = [k_1, k_2, k_3, k_4, k_5, k_6, k_7]. \quad (4)$$

The set of possible actions (Table 2) is defined as a finite combination of natural ventilation and window shading (or insulation) positions. Partial opening for natural ventilation is allowed, while shading may only be on or off.

Table 2. Q-Learning / REINFORCE Actions

Action	Ventilation	Insulation / Shading
a_1	0 (closed)	1 (on)
a_2	1 (fully open)	1
a_3	0	0 (off)
a_4	0.5	0
a_5	0.5	1
a_6	0.1	0
a_7	0.1	1
a_8	1	0

Finally, the reward function is designed to reduce mechanical heating and cooling loads and to avoid frequent changes in actions. The reward function can be expressed as a weighted sum of two components, r_1 and r_2, where each component corresponds to the reward component of the energy consumption at time t, E_t (normalized by constant C), and change in action, respectively, while w_1 and w_2 are the corresponding weights.

$$r(E_t, a_t, a_{t-1}) = w_1 r_1 + w_2 r_2, \quad \text{where} \quad (5)$$

$$r_1 = \begin{cases} +1 & E_t = 0 \\ -1 - E_t/C & E_t > 0 \end{cases}, \quad \text{and}$$

$$r_2 = \begin{cases} +1 & a_t = a_{t-1} \\ -1 & a_t \neq a_{t-1} \end{cases}.$$

3.2 REINFORCE MDP Formulation

For the REINFORCE implementation, the reward function and actions are the same as in the tabular Q-learning case. The key difference is that the raw measurements are used to define continuous states. Instead of determining the trends of the time window of the 10 most recent indoor and outdoor temperature measurements, the vector containing the 10 most measurements itself is used in the state. Since the forecast previews a much longer time period, i.e., 20 hours into the future (120 timesteps), 10 maximum values from equally-spaced bins are selected from the raw forecast information. The maximum value of each bin is used to define the forecast vector $\boldsymbol{T_t^F} = \{T_1^F, \ldots, T_{10}^F\}$, where T_i^F is the maximum value of the ith bin. Prior to adding the heat flux information, the raw heat flux q is normalized and stretched into a vector $\boldsymbol{Q_t} = \{q_{norm}, \ldots, q_{norm}\} \in \mathbb{R}^5$, where q_{norm} is the normalized value of the heat flux at time t. This is done to compensate for the fact this scalar value is being combined with vectors and to ensure that the information does not lose relative significance. Similarly, the energy consumption status (1 for mechanical heating/cooling on and 0 for off) is also stretched into a vector form, $\boldsymbol{E_s} \in \mathbb{R}^5$, prior to combining the information. The state for REINFORCE $\boldsymbol{S_t^{PG}}$ can then be expressed as a concatenated vector:

$$\boldsymbol{S_t^{PG}} = [T_{t-9}^{in}, \ldots, T_t^{in}, T_{t-9}^{out}, \ldots, T_t^{out}, \boldsymbol{T_t^F}, \boldsymbol{Q_t}, \boldsymbol{E_s}]. \quad (6)$$

3.3 Demonstration Setup in Energy Plus

A single 45 m^2 multi-family dwelling unit with southern and rooftop exposure was modeled in EnergyPlus 9.2 and used in all simulations. Opaque envelope and glazing materials, infiltration, internal gain rates, and thermostat setpoints (heating: 20℃; cooling: 25℃) were specified according to the 2018 International Energy Conservation Code, and fresh air ventilation was set according to ASHRAE Standard 62.2-2016. Interior operable panels (k=0.014 W/mK) with edge seals served as both movable insulation and shading. TMYx 2004-2018 weather files (Lawrie and Crawley, 2019) were used as noted.

The Q-learning algorithm was implemented in MATLAB, and the EnergyPlus Co-Simulation Toolbox (Dostal and Baumelt, 2019) was used to communicate between EnergyPlus and the algorithm. The REINFORCE algorithm was implemented in Python, and MATLAB was used to communicate information between Python and EnergyPlus.

For Q-learning, a tabular setup represented and updated the value function $Q(s, a)$. For REINFORCE, an artificial neural network with three layers, each with 200 nodes, was used as the function approximator. With a discrete set of actions, a softmax policy was used, in which the output of the neural network is a multinomial distribution of the normalized probabilities of each action, where the final action is sampled from the given multinomial distribution. Both value- and policy-based approaches were trained in Albany NY in May, chosen for its representation of outdoor air temperatures both above and below thermostat setpoints.

4. RESULTS AND DISCUSSION

4.1 Training Results

The Q-learning algorithm was trained over 30 iterations each lasting 31 days (1 month), converging after approximately 15 iterations, while the REINFORCE algorithm was trained over 1700 iterations, requiring about 1500 to converge (Fig. 2). This difference stems primarily from the fact that Q-learning has a much smaller finite state-space, whereas REINFORCE is a Monte Carlo method in which trajectories are sampled over each entire episode. Additionally, function approximation-based methods tend to require more data, resulting in longer training periods.

4.2 Performance

Following training on one month of typical May weather in Albany NY, the policy derived from Q-learning reduced heating and cooling loads in the experimental dwelling from a baseline level of 115 MJ (i.e. without passive systems operating) to 53 MJ. The policy trained by REINFORCE, however, reduced total loads to 22 MJ demonstrating substantially greater effectiveness. Indoor air temperatures achieved by the two policies are compared for

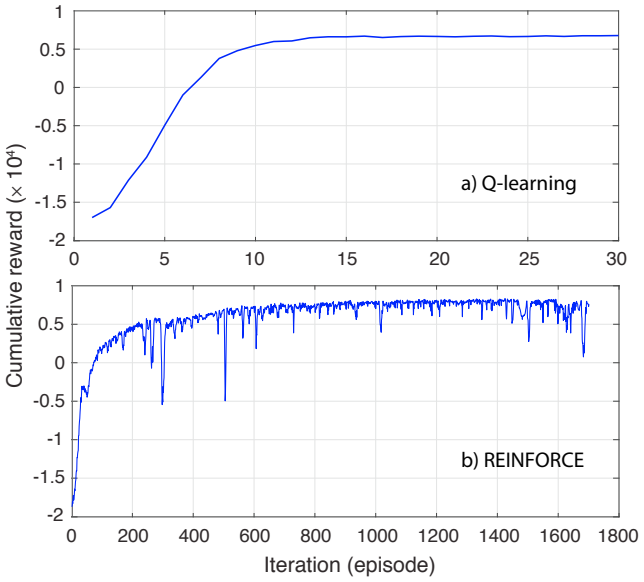

Fig. 2. Training of a) Q-learning and b) REINFORCE algorithms.

one representative week in Fig. 3, in which REINFORCE most improved upon Q-learning primarily by maintaining cooler temperatures in advance of hot days and by limiting ventilation during the warmest hours. Although both algorithms used weather forecasts, function approximation-based methods are better suited to continuous state-spaces, increasing the ability to distinguish between similar states. In Q-learning, ambiguity within state definitions often hinders the learning process because the agent may need to take different actions for the same state. Further efforts therefore focused on the policy-gradient algorithm.

Fig. 4 illustrates an example of the shading and ventilation actions learned by the REINFORCE-trained policy. When outdoor air is cool (early May 8), the policy maintains indoor air near the midpoint between thermostat setpoints by allowing solar gain (shading off) and excluding outdoor air (ventilation off); when warmer outdoor air is forecast, the policy pre-cools the space with both ventilation and shading, at the expense of inducing a small heating load. May 9 shows a similar pattern: When cool outdoor temperatures resume, the policy maintains warmer indoor conditions; then, as outdoor air warms, the policy returns to shading and ventilation to keep indoor conditions cooler. The same pattern occurs on May 10, but in this case, the pre-cooling is not sufficient to avoid all cooling loads. Additionally, intermittent natural ventilation during the afternoon, when warm outdoor air should have been excluded, increases cooling loads unnecessarily. This represents an important opportunity for future improvement.

To evaluate robustness, the policy trained in Albany, NY was deployed in six contrasting climates: warm and cool-summer Mediterranean (Los Angeles; Portland OR), semi-arid (Salt Lake City), humid subtropical (Kansas City), and hot and warm-summer continental (Pittsburgh; Detroit). Comparisons included baseline models both without passive systems and with conventionally controlled shading (on if incident solar radiation $\geq 250\,\mathrm{W/m^2}$) and natural ventilation (on if $20°\mathrm{C} \leq \mathrm{T_{out}} \leq 25°\mathrm{C}$) (Fig. 5). Strikingly, the trained policy reduced total loads by 47-76%, showing 13-64% improvement over conventional controls, in five of the six cities. These five again illustrate the strategy of pre-cooling shown in Fig. 4, shown by the trading of higher cooling loads for lower heating loads, while the one exception, Detroit, emphasizes the need for improved warm-hour ventilation strategies.

Next, we investigated heating and cooling performance of the REINFORCE policy trained in Albany in May among contrasting months in the same climate, again in comparison with models without passive systems or having only conventional controls (Fig. 6). Conventional controls sometimes increased loads above those of models without passive systems, illustrating a well-known problem. As expected, the trained policy out-performed models without passive systems in all months, although not to the extent accomplished in the training month. More importantly,

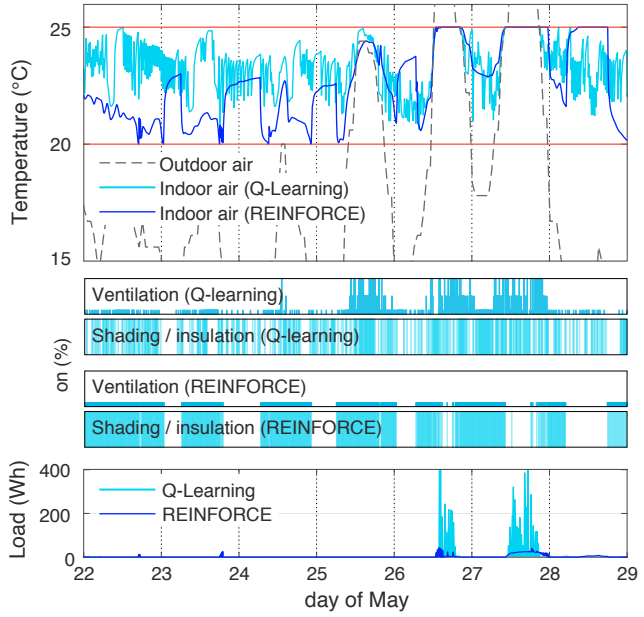

Fig. 3. Comparison of temperatures and heating/cooling loads achieved by Q-learning and REINFORCE-trained policies.

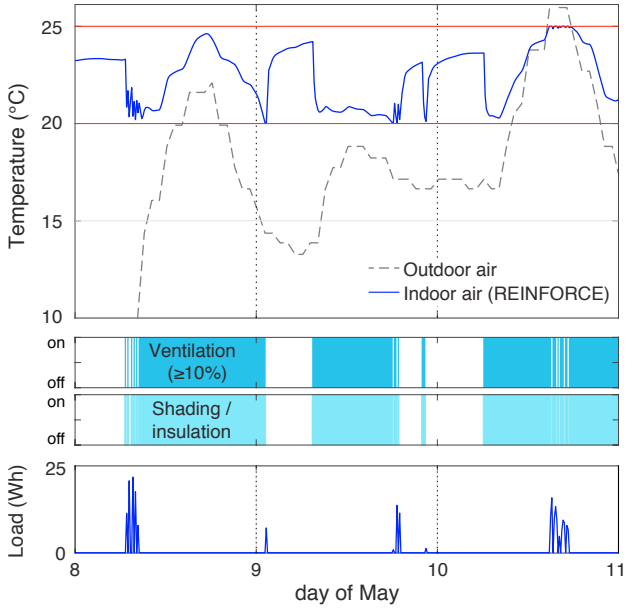

Fig. 4. Actions, indoor air temperatures, and unmet heating/cooling loads of the REINFORCE-trained policy in Albany NY.

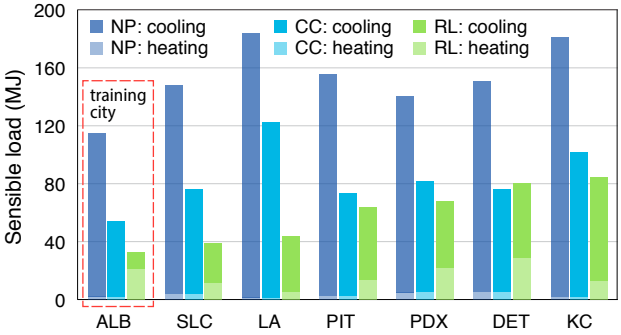

Fig. 5. Heating and cooling loads in models without passive systems (NP); with conventionally controlled systems (CC); and with REINFORCE-learned controls (RL). ALB: Albany NY; SLC: Salt Lake City UT; LA: Los Angeles CA; PIT: Pittsburgh PA; PDX: Portland OR; DET: Detroit MI; KC: Kansas City MO, each represented by May in TMYx 2004–2018 weather files.

however, the trained policy improved upon conventional controls by substantial margins (31-39%) in all months but one, suggesting that learning-based approaches could provide a pathway for greater adoption of passive systems.

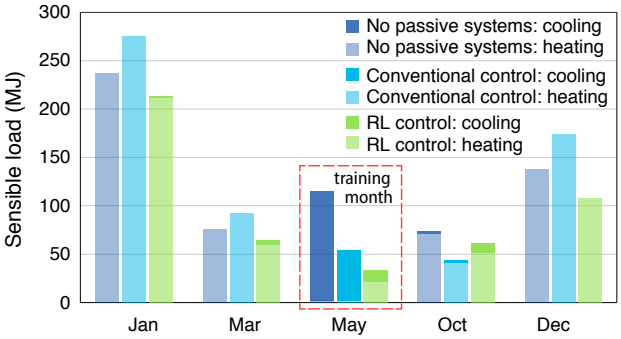

Fig. 6. Heating and cooling loads in models without passive systems (NP); with conventionally controlled systems (CC); and with the REINFORCE-learned policy in contrasting months of the Albany NY TMYx 2004-2018 weather file.

5. CONCLUSIONS

This study investigated two RL algorithms for reducing space heating and cooling loads in buildings through integrated control of shading, movable insulation, and natural ventilation. The REINFORCE-trained policy reduced heating and cooling loads to less than half of those achieved by the tabular Q-learning policy in the training environment (Albany NY in May); it also reduced May loads by 47-76%, compared to models without passive systems in six other climates ranging from humid subtropical to cold semi-arid. Together, these results show the advantage of policy-gradient over tabular methods for the control of multi-faceted passive conditioning systems in buildings, which has not previously been demonstrated, as well as the potential robustness of resulting policies for deployment in contrasting climates. Future work will address three primary limitations: first, the greater challenge of learning effective responses to warm days vs. cool ones; second, the challenge of operating shading and natural ventilation independently; and finally, the time required for training. These issues may be addressed by using well-defined expert systems to pre-train the initial policy, reducing undesirable random actions, accelerating the training process, and guiding the algorithm toward a more seasonally robust policy through expert demonstrations. Finally, actor-critic or policy gradients with baselines may be used to reduce the variation during training and accelerate learning.

REFERENCES

Bellman, R., 1966. Dynamic programming. Science 153, 34–7.

Chen, Y., Norford, L. K., Samuelson, H. W., Malkawi, A., 2018. Optimal control of HVAC and window systems for natural ventilation through reinforcement learning. Energy Build. 169, 195–205.

Cheng, Z., Zhao, Q., Wang, F., Jiang, Y., Xia, L., Ding, J., 2016. Satisfaction based Q-learning for integrated lighting and blind control. Energy Build. 127, 43–55.

Dostal, J., Baumelt, T., 2019. EnergyPlus Co-Simulation Toolbox. https://github.com/dostaji4/.

Drgoňa, J., Arroyo, J., Figueroa, I., Blum, D., Arendt, K., et al., 2020. All you need to know about model predictive control for buildings. Annu. Rev. Control 50, 190–232.

Grynning, S., Time, B., Matusiak, B., 2014. Solar shading control strategies in cold climates–heating, cooling demand and daylight availability in office spaces. Sol. Energy 107, 182–194.

IEA, 2020. Tracking Buildings 2020. www.iea.org/reports/tracking-buildings-2020.

Lawrie, L., Crawley, D., 2019. Development of typical meteorological years (TMYx). climate.onebuilding.org.

Liu, M., Wittchen, K. B., Heiselberg, P. K., 2015. Control strategies for intelligent glazed façade and their influence on energy and comfort performance of office buildings in Denmark. Appl. Energy 145, 43–51.

Lucon, O., et al., 2015. Ch. 9: Buildings. In: Climate Change 2014: Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press.

Minakais, M., Mishra, S., Wen, J. T., 2019. Database-driven iterative learning for building temperature control. IEEE T. Autom. Sci. Eng. 16, 1896–1906.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., et al., 2013. Playing Atari with deep reinforcement learning. arXiv preprint:1312.5602.

Oropeza-Perez, I., Østergaard, P. A., 2018. Active and passive cooling methods for dwellings: A review. Renew. Sustain. Energy Rev. 82, 531–44.

Peng, Y., Rysanek, A., Nagy, Z., Schlüter, A., 2018. Using machine learning techniques for occupancy-prediction-based cooling control in office buildings. Appl. Energy 211, 1343–58.

Rempel, A. R., Lim, S., 2019. Numerical optimization of integrated passive heating and cooling systems yields simple protocols for building energy decarbonization. Science Technol. Built Environ. 25, 1226–36.

Sutton, R. S., Barto, A. G., 2018. Reinforcement learning: An introduction. MIT Press.

Wang, Z., Hong, T., 2020. Reinforcement learning for building controls: The opportunities and challenges. Appl. Energy 269, 115036.

Watkins, C. J., Dayan, P., 1992. Q-learning. Mach. Learn. 8, 279–92.

Williams, R. J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. 8, 229–56.