Highlights

Ten Questions Concerning Reinforcement Learning for Building Energy Management

Zoltan Nagy, Gregor Henze, Sourav Dey, Javier Arroyo, Lieve Helsen, Xiangyu Zhang, Bingqing Chen, Kadir Amasyali, Kuldeep Kurte, Ahmed Zamzam, Helia Zandi, Ján Drgoňa, Matias Quintana, Steven McCullogh, June Young Park, Han Li, Tianzhen Hong, Silvio Brandi, Giuseppe Pinto, Alfonso Capozzoli, Draguna Vrabie, Mario Berges, Kingsley Nweye, Thibault Marzullo, Andrey Bernstein

- Buildings transition from passive consumers to active grid assets for resiliency.
- Reinforcement learning (RL) addresses flexible energy management challenges.
- Ten critical questions on RL in buildings, considering data, tools, and deployment.
- Overview of research, accomplishments, challenges, and opportunities in RL.
- Future research directions to expedite RL adoption for improved energy management.

Ten Questions Concerning Reinforcement Learning for Building Energy Management

Zoltan Nagy[a,*], Gregor Henze[b,d], Sourav Dey[b], Javier Arroyo[c], Lieve Helsen[c], Xiangyu Zhang[d], Bingqing Chen[e], Kadir Amasyali[f], Kuldeep Kurte[f], Ahmed Zamzam[d], Helia Zandi[f], Ján Drgoňa[g], Matias Quintana[h,n], Steven McCullogh[i], June Young Park[i], Han Li[j], Tianzhen Hong[j], Silvio Brandi[k], Giuseppe Pinto[k,l], Alfonso Capozzoli[k], Draguna Vrabie[g], Mario Berges[m], Kingsley Nweye[a], Thibault Marzullo[d], Andrey Bernstein[d]

[a] *Department of Civil, Architectural and Environmental Engineering, University of Texas at Austin, USA*
[b] *Department of Civil, Environmental and Architectural Engineering, University of Colorado Boulder, USA*
[c] *Department of Mechanical Engineering, University of Leuven (KU Leuven), part of EnergyVille, Belgium*
[d] *National Renewable Energy Laboratory, Golden, CO, USA*
[e] *Bosch Center for Artificial Intelligence, Bosch Research North America, Pittsburgh, PA, USA*
[f] *Oak Ridge National Laboratory, Knoxville, TN, USA*
[g] *Pacific Northwest National Laboratory, Richland, WA, USA*
[h] *Department of the Built Environment, National University of Singapore*
[i] *Department of Civil Engineering, University of Texas at Arlington, USA*
[j] *Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory, USA*
[k] *Politecnico di Torino, Department of Energy, TEBE research group, BAEDA Lab, Italy*
[l] *PassiveLogic, Salt Lake City, UT, USA*
[m] *Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, USA*
[n] *Future Cities Laboratory Global, Singapore-ETH Centre, Singapore*

Abstract

As buildings account for approximately 40% of global energy consumption and associated greenhouse gas emissions, their role in decarbonizing the power grid is crucial. The increased integration of variable energy sources, such as renewables, introduces uncertainties and unprecedented flexibilities, necessitating buildings to adapt their energy demand to enhance grid resiliency. Consequently, buildings must transition from passive energy consumers to active grid assets, providing demand flexibility and energy elasticity while maintaining occupant comfort and health. This fundamental shift demands advanced optimal control methods to manage escalating energy demand and avert power outages. Reinforcement learning (RL) emerges as a promising method to address these challenges. In this paper, we explore ten questions related to the application of RL in buildings, specifically targeting flexible energy management. We consider the growing availability of data, advancements in machine learning algorithms, open-source tools, and the practical deployment aspects associated with software and hardware requirements. Our objective is to deliver a comprehensive introduction to RL, present an overview of existing research and accomplishments, underscore the challenges and opportunities, and propose potential future research directions to expedite the adoption of RL for building energy management.

Keywords:

Introduction

Significant energy is expended in buildings for the provision of thermal comfort, particularly by the heating, ventilating and air-conditioning (HVAC) systems, which account for more than 40% [104] of average building energy use. Humans spend more than 86% of their time indoors [59]. Building automation and control systems operating these HVAC systems are responsible for maintaining comfortable, safe, and healthy indoor conditions

*Corresponding author
Email address: nagy@mail.utexas.edu (Zoltan Nagy)

while also aiming to reduce the energy use of buildings. Balancing the goals of maintaining comfortable indoor conditions and reducing energy use requires careful trade-off. Increasingly, building control systems are expected to consider additional goals such as energy flexibility, carbon emissions, and managing on-site renewable energy production and storage [120], while involving human occupants more actively in building operations. Addressing the multitude of operational objectives requires advanced controllers capable of trading off between multiple, potentially conflicting, goals and adapting to emerging technologies [15].

This paper aims to detail and summarize the developments of one particular advanced control paradigm, reinforcement learning (RL), for building energy management and contextualize it within the general research field. The paper is organized around ten questions: The first four questions, Q1—Q4, serve as general introduction and background to RL. Subsequently, Q5—Q8 address practical questions for RL in buildings. Finally, Q9 and Q10 delineate the challenges and future opportunities.

The paper targets early career researchers entering this exciting research area, practitioners interested in the benefits RL can provide to the industry, and policymakers seeking to understand the role of RL in decarbonizing the built environment.

1. What is Reinforcement Learning and what is its promise for buildings?

Gregor Henze, Sourav Dey

Conventional building controls cannot achieve the multi-objective optimization required for efficient energy management, as they rely on predetermined schedules and limits based on expert experience, and are unable to adapt to changing objectives. Despite being developed by building control experts and expressed in ASHRAE Guideline 36-2018 [7], these rule-based and heuristic strategies may not necessarily be optimal as they are not tailored to the specific building and site conditions. Additionally, they are reactive and do not consider available forecasts such as weather, utility price signals, and building occupancy. Thus, although requiring considerable engineering expertise and time for development, tuning, and performance monitoring to achieve acceptable performance, they prove sub-optimal in their performance. Here, advanced control strategies such as reinforcement learning (RL) and model predictive control (MPC) offer the benefit of adapting their control policies in response to changing objectives which purely reactive heuristic controls are unable to.

To temper the enthusiasm, a simulation study recently revealed that GLD36 control strategies for variable air volume HVAC systems in a medium office building performed on par with advanced optimization based and reinforcement learning controllers when energy consumption is selected as the objective [79]. This revealed that learning control will not always provide substantial benefits to best-in-class rule-based controls, but instead, RL is most suitable when optimal sequential decision-making over a long-time horizon is expected to yield significant benefits. Asking a learning agent to uncover reactive control strategies found through the year-long consensus-driven research by domain experts is a fool's errand and RL control applications need to be chosen wisely. Prime among them are those that benefit from anticipating dynamic changes in important driving forces such as weather, a variety of utility signals (including energy cost, demand cost, and marginal carbon emission factors), occupancy, building utilization and in response planning strategies over longer time scales that optimize scalar or multi-objective cost functions whose solutions are non-trivial and commonly elusive. This suggests that RL is better suited for high-level energy management than low-level local feedback control and control sequences.

Although MPC has been successful in process control applications in chemical plants and refineries [118], and has gained considerable popularity in building control research [35], it has not yet been adopted widely in commercial buildings. One of the main bottlenecks is that MPC requires the development of control models which capture the dynamic behavior of the building and its HVAC systems. Since every commercial building is unique, extensive expertise is required to develop and maintain an accurate control model for each building [146]. RL, in turn, appears attractive as it can learn optimal control actions by interaction without any model development and improve its control policy over time, adapting to changing dynamics even in challenging environments. In Q2, we compare MPC with RL in greater detail.

RL is a form of goal learning by interacting with an environment. It is a branch of machine learning alongside supervised and unsupervised learning where the aim of RL is to learn and behave in a

desired manner to maximize a goal [132]. Unlike supervised learning, it does not learn from labeled data nor uncovers the structure and patterns in unlabeled data with unsupervised learning. Markov Decision Processes (MDPs) provide the mathematical framework for RL algorithms to learn how to make decisions in uncertain and dynamic environments. The MDP framework assumes that the environment is Markovian, meaning that the current state fully captures all relevant information about the past. This allows the agent to make decisions based solely on the current state without needing to consider the entire history of past states and actions. An MDP models a decision-making problem as an agent interacting with an environment in the form of the tuple of (S, A, P, R, γ). The agent takes actions from a set of possible actions (A), being in a state s from a set of possible states (S) in the environment, and the environment responds by providing a reward signal r from the reward domain (R) and a new state. P is the transition function, which specifies the probability of transitioning from one state to another given an action while γ is a value between 0 and 1, called the discount factor that determines the relative importance of immediate versus future rewards. The agent's goal is to maximize its expected cumulative reward over time, which requires the agent to learn a policy in a trial-error fashion, mapping states to actions.

In domains unrelated to buildings, RL has been responsible for several success stories and media attention: It was able to play a range of Atari 2600 video games at a human expert level [90], defeat a human world champion in the game of Go [130], outrace the world's best Gran Turismo video game drivers [152], and found success in fields like autonomous vehicles [123, 43, 54], robotic applications [64, 65], automatic trading [30], targeted recommendation systems, natural language processing (NLP) like ChatGPT [105], and others.

For building applications, an initial work in this area is the Adaptive Control of Home Environment (ACHE) or Neural Network House by Mozer [92]. One of the early uses of model-free RL in the buildings energy systems domain was conducted by Henze et al. [52] to control a thermal energy storage system, followed by Liu et al.[74] which performed several systematic RL studies on finding control strategies for using passive thermal systems. In [75, 76], a field implementation was conducted with RL, where the RL was pre-trained with an emulator model, identified from experimental data. The studies mentioned were able to find effective strategies but were not able to outperform some conventional control strategies like MPC. These were conducted before the deep learning revolution in 2012. The decade between 2005-2015 saw comparatively few advances in RL for building energy management; rather most research focused on MPC.

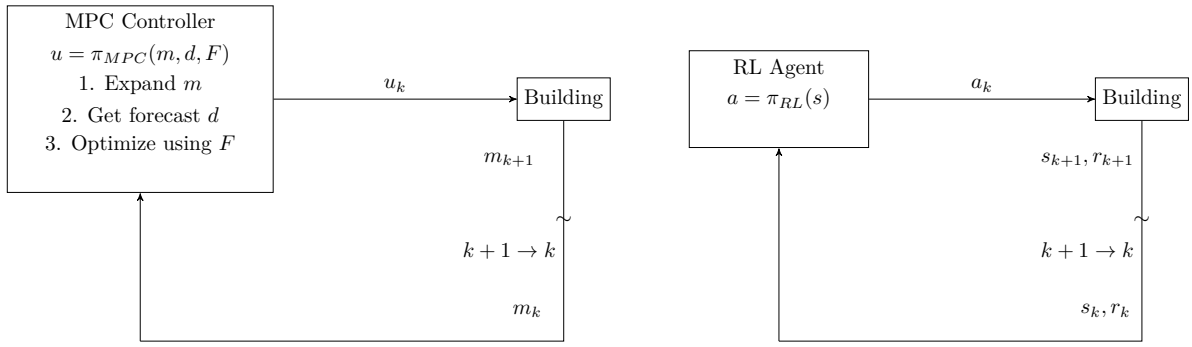
The advancements in deep learning, the availability of powerful computational resources, and the development of RL algorithms again enabled research interest in deep RL algorithms applied to building energy systems. Yang et al. [156] first implemented a multi-agent deep RL approach for low-exergy buildings able to handle continuous states. Several RL applications in building controls were conducted in [66, 24, 112, 149]. The interactions between human and building system interfaces (e.g., thermostat, light switch) were also investigated by Park et al. [108]. Furthermore, multi-agent RL has been used to discover the complex relationships between buildings (or grid) by Vazquez-Canteli et al. [142]. The most recent breakthroughs involve marrying the power of model-based approaches with adaptive learning algorithms: Chen et al. [18] and Drgoňa et al. [37] demonstrated an impressive reduction of learning times by employing a low-order model whose parameters are learned from the controlled building, and Arroyo et al. [5] explored combining the advantages of RL for tuning the MPC policies.

2. How is RL related to other control theory methods?

Javier Arroyo, Lieve Helsen

Reinforcement learning (RL) is an advanced control technique that is particularly well-suited for sequential decision-making problems. It is part of a broader family of machine learning methods that includes supervised and unsupervised learning, but RL is unique in its ability to deliver an action that maximizes the performance of a process. RL has its roots in Dynamic Programming (DP), an optimization method that breaks down the main problem into simpler subproblems. RL is also related to Approximate Dynamic Programming (ADP), another DP variant. While RL is arguably the most powerful machine learning technique due to its ability to deal with control, it is also the most complicated to train.

As introduced in Q1, the two other main control categories that are considered for HVAC appli-



(a) MPC (b) RL

Figure 1: Simplified schemas of MPC (left) and RL (right). Inspired from [5]. k indicates the discrete time step.

Aspect	**RBC**	**MPC**	**RL**
Community	Engineering	Control theory	Machine learning
Formulation	Arbitrary	Receding horizon	Markov Decision Process
Goal	Accomplish a set of predefined rules	Minimize an objective integrand function	Maximize the total expected cummulative return
Control policy	$u = \pi_{RBC}(m)$	$u = \pi_{MPC}(m, d, F)$	$a = \pi_{RL}(s)$
Is adaptive	No	Not in its basic form	Yes
Is predictive	No	Yes	Yes
Prediction horizon	-	Hours to days for its application in buildings	Infinite (using discount factors)
Requires configuring a system model	No	Yes	No (except for simulation- or model-based RL)
Data requirements	None	Order of days/weeks for its application in buildings	Order of years for its application in buildings
Computational cost	Very low	High online cost	High offline cost
Optimality	None	Can provide necessary conditions for optimality. Sufficient conditions only if strict convexity properties are met	Can only guarantee optimality when using linear policy function approximations and Monte-Carlo learning.
Constraint handling	Constraints are explicitly imposed but can lead to violations since it is a reactive controller	Constraints are explicitly imposed. Violations can arise because of model mismatch and system uncertainty.	Constraints are not explicitly. Hence, constraint violations are expected.
State observer	-	Yes	No, but may extend observation vector with regressive variables
Forecast data	-	Introduced as time-series trajectories	Introduced in the agent's observation vector
Penetration in HVAC industry	High	Low	Very low
Barrier for adoption	None, already widely spread	Configuration effort and optimization solvers	Data efficiency

Table 1: Comparison of different aspects of the main types of controllers considered for HVAC control: RBC, MPC, and RL.

cations in buildings are rule-based control (RBC) and model predictive control (MPC). Historically, RBC has been the predominant choice for HVAC applications because RBC is fundamentally simple. RBC is generally reactive and consists of a predefined set of rules that trigger actions based on a gathering of measurements. These rules may easily escalate with the HVAC complexity. Contrarily to RL, RBC does not naturally integrate mechanisms to shape the control policy, which remains static throughout the lifetime of the controller. The most intuitive and common RBC is a conventional thermostat. Other examples are: thermostatic radiator valves, hysteresis controllers, proportional-integral-derivative controllers, heating curves, or complete programs encoded in a building management system. All RBCs can be defined by a vector of measurements "m", a control policy that is *manually predefined* π_{RBC}, and a vector of controls u, as: $u = \pi_{RBC}(m)$.

More advanced controllers are needed to provide building-to-grid services and to enhance the energy efficiency and/or guarantee the cost-effectiveness of HVAC systems as they become more sophisticated.

A first step is to make control algorithms predictive to take advantage of weather forecasts and to anticipate occupancy. MPC has arisen as a genuine candidate for predictive control with already an extensive research background for their application in buildings and some (rather limited) penetration in the HVAC industry [35]. MPC uses a model to determine the influence of future inputs on the building dynamics. RL does not use a model of the building, at least in its basic form, though in the case of simulation- or model-based RL a model is used as well (see Q3 for a categorization of RL algorithms). Instead, vanilla RL continuously learns from empirical observations using a reward signal to construct a control policy that is refined online.

The machinery of MPC starts from the state observer, which expands the available measurements to the expected values of the full state vector of the model. RL does not include a state observer but can extend the current observation with past observations to better capture the system state. The state observer is core to any MPC implementation and constitutes one of its major strengths because it can provide an accurate and comprehensive representation of the system state at every control step. Forecast variables can be included in both the MPC model and the RL observation vector. In MPC, the forecasting variables are introduced as time trajectories and set as exogenous variables in the associated optimal control problem. On the other hand, RL includes every forecasting point as an independent new observation, which can easily grow the observation space dimension with the prediction horizon.

Once the initial state and forecasts are known, MPC uses the building model to solve a trajectory optimization problem over a finite prediction horizon. Only the first control input from the trajectory is applied to iteratively repeat the process, as such using a receding horizon to address uncertainties. Hence, the MPC control policy can be summarized as $u = \pi_{MPC}(m, d, F)$, where d is the vector of forecasted disturbances and F represents the controller model dynamics. RL directly uses the agent's learned policy to pick a control action from an observation aiming to maximize the total expected cumulative return. Therefore, RL typically implements an infinite prediction horizon, but it uses a discount factor that determines the extent to which future rewards are effectively considered. A summary of the control policy of RL is $a = \pi_{RL}(s)$, where a is the action taken by the agent, which is analogous to the vector of controls u in the machine learning community, and s is the RL agent's observation of the state. Note that an observation can include not only measurements, but also regressive and predictive variables, or any notion of time like the day of the week.

Solving a trajectory optimization problem online in MPC is computationally more expensive than retrieving the action from an RL agent given a current observation, which only requires a function evaluation. However, constructing an RL policy typically requires a substantial data set for training and RL suffers from the so-called curse of dimensionality. MPC may also require historical data for training or calibration of its model parameters. However, the data requirements to calibrate the building model of an MPC are in the order of days to weeks, while those of RL to learn a policy are often in the order of years [16]. For MPC more data are needed when we move from the white-box over the grey box to the black-box approach, and the physics included in white and grey-box approaches allow extrapolation beyond data-collection operation conditions.

None of the control approaches can guarantee global optimality unless strict conditions are met. In the case of RBC, performance is suboptimal because the control logic entirely relies on engineering experience and heuristics. MPC requires convexity of the objective function and constraints to obtain optimality guarantees, qualities that are hard to maintain when aiming for an accurate representation of the building envelope and HVAC system. On the other hand, RL requires linear policy function models or Monte Carlo learning. The former limits the degrees of freedom of the control policy, and the latter leads to high learning variance [131].

In general, MPC and RL have complementary strengths and weaknesses, which suggests a high potential in combining both. RL has higher data requirements, but MPC is more computationally intensive online and demands higher engineering effort to configure the controller model. MPC can better deal with constraints because they are directly defined in the optimization problem while RL only indirectly takes them into account through penalization in the reward function. Hence, MPC explicitly imposes constraints whereas RL has to derive them. On the other hand, RL is more naturally adaptive as it has been developed to genuinely learn from the environment. Table 1 compares the main characteristics of each controller. The penetration of one or the other in the HVAC control industry

will be determined by which barriers for adoption are lowered first. RL requires more data-efficient methods while MPC would benefit from tools automating the configuration of building models and more powerful solvers. Maybe the future is not the one or the other but rather a hybrid method where the strengths of both are merged.

3. What are different types of RL algorithms and what are their advantages and disadvantages?

Bingqing Chen, Xiangyu Zhang, Kadir Amasyali, Sourav Dey

While the large number of RL algorithms may appear daunting, they share the same objective: considering a policy π_θ, parameterized by learnable θ in a parameter space Θ, the goal is to search for the *optimal* parameter $\theta^* \in \Theta$ that maximizes the expected total reward, as shown by (1).

$$\theta^* = \arg\max_\theta \underbrace{\mathbb{E}_{\pi_\theta}\big[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\big]}_{J(\theta)} \tag{1}$$

$J(\theta)$ is a shorthand for the expected total reward, which will be referred to later.

To differentiate and analyze these RL algorithms, there are multiple ways they can be categorized (see Fig. 2): 1) Model-based vs. model-free; 2) Value-based vs. policy-based vs. actor-critic; 3) On-policy vs. Off-policy; 4) Online vs. Offline; and 5) Single agent vs. Multi-agent. In this section, we provide an overview of these categories and present their key features.

3.1. Model-based vs Model-free

Depending on whether a model of the environment is given to the agent or learned by the agent, RL methods can be categorized as model-based and model-free. In the model-based algorithms, an agent focuses on understanding its environment and develops a model of the environment through interacting with it. A common idea for model-based RL is to simultaneously learn a model of the environment and plan ahead based on the learned model. The classical Dyna-Q [131] is an example of such approach. Developing upon the idea, people have modeled the environment with Gaussian Process [57], locally linear models [148], and neural networks [23]. Previously, it was believed that model-based RL could not perform as well as model-free RL asymptotically [23]. But, recent work has shown that model-based RL can match the asymptotic performance of model-free RL algorithms, while being significantly more sample efficient [23].

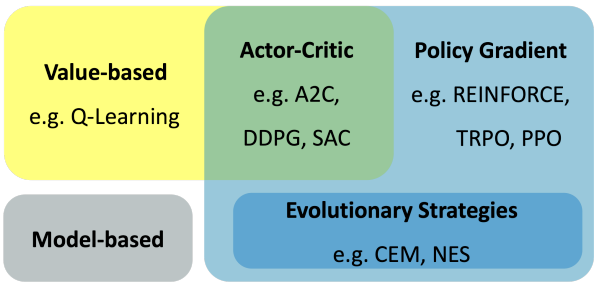


Figure 2: Classification of RL Methods

The model-free RL algorithms learn the outcomes of their actions through experience. As such, these algorithms carry out an action a number of times and adjust the policy for optimal rewards, based on the outcomes, without learning any model of the environment. For model-free RL algorithms, they can be further classified into three types, i.e. value-based methods, policy gradient methods, and actor-critic methods, see Fig. 2 and Table 2.

3.2. Value-based vs Policy-based

Value-based methods, e.g. Q-learning and its variants, learn the policy indirectly by learning value functions with exploration, e.g. $Q_\pi(s,a)$ or $V_\pi(a)$, see (2a) and (2b), and take the action that maximizes the value function. Alternatively, one may use the advantage function [127], $A_\pi(s,a)$ as given in (2c), which could be interpreted as how much a given action improve upon the policy's average behaviour.

The value function may be updated via methods such as Bellman backup [131], $Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')$. Depending on the dimensionality of the state-action space, the value function can be an exact representation, e.g., tabular-Q, or a function approximation, e.g., a neural network. A major shortcoming of Q-learning is that it is only applicable to problems with discrete action spaces. Thus, for problems with continuous action spaces, which are common among building control problems, each continuous action needs to be discretized into a number of discrete actions, which results in

a large action space.

$$Q_{\pi_\theta}(s,a) = \mathbb{E}_{\pi_\theta}\big[\sum_{k=0}^{\infty}\gamma^k r_{t+k}|s_t=s,a_t=a\big] \quad (2a)$$

$$V_{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}\big[\sum_{k=0}^{\infty}\gamma^k r_{t+k}|s_t=s\big] \quad (2b)$$

$$A_{\pi_\theta}(s,a) = Q_{\pi_\theta}(s,a) - V_{\pi_\theta}(s) \quad (2c)$$

Policy gradients methods directly search for an optimal policy π_θ^*, using stochastic estimates of policy gradients, and thus are applicable to problems with continuous action spaces. To do that, these methods compute an estimate of the policy gradient defined in (3a) and optimize the objective with stochastic gradient ascent (3b). Aside from the obvious benefit of being able to handle continuous action spaces, policy gradients methods have several advantages over value-based ones, as discussed in [131]. Firstly, policy gradient methods can learn both deterministic and stochastic policies, while there is no natural way to learn stochastic policy with value-based methods. Secondly, the policy may be a simpler function to approximate, and thus policy-based methods typically learn faster and yield a superior asymptotic policy. Finally, the choice of policy parameterization is a natural way to inject domain knowledge into RL.

$$g \coloneqq \nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\pi_\theta}\big[\sum_{k=0}^{\infty}\gamma^k r_{t+k}\big] \quad (3a)$$

$$\theta \leftarrow \theta + \alpha\hat{g} \quad (3b)$$

A variety of policy gradient algorithms have been proposed in the literature, and most of them approximate g based on the *policy gradient theorem* in Eq. 4. Refer to [131, Chapter 13] for its deviation and more details..

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) Q_{\pi_\theta}(s,a)\right] \quad (4)$$

Perhaps, REINFORCE [131] is the most well-known one, which follows directly from the *policy gradient theorem*. However, it suffers from large performance variance and unstable policy updates [128]. PPO [128] is the most recent work in a line of research that improved upon vanilla policy gradient methods, including Natural Policy Gradients [56] and Trust Region Policy Optimization (TRPO) [126]. The intuition behind these methods is that in each update the policy π_θ should not change too much. PPO is known to be stable and robust to hyperparameters and network architectures [128] and outperform methods, such as A3C [88] and TRPO [126]. Also, note that even though we classify methods such as TRPO and PPO as policy gradient methods, one can and should incorporate a critic for variance reduction and more robust performance [127].

Evolutionary strategies (ES) are black-box optimization algorithms inspired by natural evolution, and can be considered as a special case of policy gradient algorithms, which make 0^{th}-order, instead of 1^{st}-order estimates of policy gradients. Unlike policy gradient methods, it is not necessary to take derivatives through the policy, exemplified by the update rule of Natural Evolutionary Strategies (NES) [122] (Eq. 5), where ϵ is a perturbation on the policy parameter θ. Instead, ES methods use 0^{th}-order information to estimate the policy gradients.

$$\begin{aligned}\nabla_\theta \mathbb{E}_{\theta\sim N(\mu,\sigma^2 I)} J(\theta) &= \nabla_\theta \mathbb{E}_{\epsilon\sim N(0,I)} J(\theta+\sigma\epsilon) \\ &= \frac{1}{\sigma}\mathbb{E}_{\epsilon\sim N(0,I)} J(\theta+\sigma\epsilon)\epsilon\end{aligned} \quad (5)$$

Recent work has demonstrated ES to be a scalable [122] and competitive [83] alternative to other more sophisticated RL methods, rekindling research interest in ES. Some well-known ES approaches include Cross-entropy Method (CEM) [133], Natural Evolutionary Strategies (NES) [122], and Finite Difference method [83].

Actor-critic methods, e.g. Advantage Actor-Critic (A2C), Asynchronous Advantage Actor-Critic (A3C) [88], Deep Deterministic Policy Gradient (DDPG) [71], and Soft Actor-Critic (SAC) [49] are hybrids of the value-based and policy gradient approaches. Actor-critic methods use a policy network to select actions (the actor), and a value network to evaluate the action (the critic). Actor-critic methods have been used for building control in [69, 164, 47]. A2C use a Q-network as critic and thus are only applicable to problems with discrete action spaces, and thus suffer from the same drawback as Q-learning. DDPG, on the other hand, is applicable to problems with continuous action spaces. SAC is a successful off-policy algorithm that can also take continuous state and action spaces and is also robust to various hyperparameter settings. It tends to explore the environment better and not get stuck in local optima. It has been found to be successful in many building applications recently.

3.3. On-policy vs Off-policy

Depending on what samples are used for learning, RL algorithms can be divided into *on-policy* and *off-policy*: On-policy algorithms conduct each update (on policy or value functions) using data collected by the current policy, whereas off-policy methods make use of data collected by other policies as well. Though it is easier to compute unbiased on-policy gradients and on-policy algorithms are relatively more stable, the sample efficiency could suffer, i.e., each data point is only used once before it is discarded. On the other hand, off-policy methods, e.g., DQN and DDPG, store samples in an experience replay and reuse them for off-policy updates. This can improve the sample efficiency but inevitably increases the complexity of implementation and also potentially causes instability in training. Overall, on-policy algorithms may be more useful for the problems where the agent can explore its environment. Off-policy algorithms, on the other hand, are more appropriate when the agent can not explore much. In Table 2, the algorithm column is shaded with two different colors, indicating two types of RL algorithms.

3.4. Online vs Offline

The next two algorithm classifications are more related to the problem formulation instead of a choice of algorithm to make. Specifically, depending on the accessibility of an environment to interact with, RL algorithms are categorized as online RL and offline RL. Online RL assumes the existence of an environment for the RL agent to explore. In contrast, offline RL algorithms learn a policy from a static batch of data (so offline RL is also known as batch RL) without any exploration. The absent of exploration can lead to high extrapolation error in value approximation when using standard off-policy methods, so some special algorithms are designed for offline RL to address such issue, see batch-constrained Q-learning (BCQ) [45].

3.5. Single Agent vs Multi-Agent

Finally, there are also RL problems with multi-agent settings, in which multiple RL agents interact with each other in an environment either competitively or cooperatively. The goal for multi-agent RL (MARL) training is to learn control policies for every agents. A key challenge in MARL is the non-stationarity, making single agent RL algorithms less effective. Paradigms like centralized training and decentralized execution (CTDE) leverages the actor-critic approach and helps to train MARL policies in a stationary manner, see [78] and [41]. We explore applications of MARL in the next Section.

4. How can RL be applied at scale for clusters of buildings?

Kuldeep Kurte, Ahmed Zamzam, Helia Zandi, Thibault Marzullo

With the increase in flexibility of buildings electrical demand and the proliferation of EV charger installations, buildings become more attractive for electrical grid load management problems. This entails the need for coordination between buildings controllers to achieve meaningful load shaping services to the electrical grid. Thus, RL solutions are often faced with the problem of controlling connected communities of grid-interactive buildings.

When applying RL to multiple buildings or clusters of buildings, a practitioner may opt for using one RL agent, or design multiple agents where each building is controlled using a separate RL agent. In the multi-agent RL (MARL) solutions, the goal is often to design local control policies that minimize or eliminate the need for continuous communications between controllers during operation. However, MARL environments suffer from i) non-stationarity which implies that the statistical properties of the environment are changing over time mainly due to the evolving policies of other agents during training ; and ii) non-uniform reward structures which often challenges the learning process. Thus, utilizing MARL algorithms for control of clusters of buildings requires specialized algorithms to tackle non-stationarity and careful design of reward functions for each control agent. On the other hand, single-agent RL solutions can be easier to learn control policies that provide optimal solutions from the grid perspective. This naturally comes at the price of requiring continuous communication of observations between buildings and the central controller. This hinders the ability to apply such solutions due to the lack of communication infrastructure that can support this requirement in addition to the privacy and security concerns.

Additionally, in scenarios where a cluster of buildings coordinate to deliver a service to the grid, the operational cost/reward of each building should be designed to reflect the contribution of

Table 2: An Overview of Popular Model-free RL Algorithms

Categories		Algorithms*	Key Features	Used by
Single Agent RL	Value Based	SARSA [121]	On-policy temporal difference update.	[77], [26], [157], [25], [150]
		Q-learning/DQN [89]	The use of experience replay and a target network.	
		Double DQN [136]	Overcomes the value function overestimation in Q-learning.	
		Duel DQN [147]	Uses dueling architecture to decouple value and advantage.	
		Rainbow DQN [53]	Combines all above modifications.	
		...	...	
	Actor-Critic	DDPG [71]	Extends DQN to continuous action space.	[163], [67], [8], [161], [12]
		TD3 [44]	Based on DDPG, addresses value function overestimation.	
		A3C [88]	Parallel training, each thread conducts minibatch SGD.	
		ACER [144]	A3C's off-policy counterpart, higher sample efficiency.	
		SAC [49]	Ensures stability and exploration via entropy maximization.	
		...	...	
	Policy Gradient	REINFORCE	Esitmates PG using full trajectories (a Monte-Carlo method)	[162], [86], [17], [20], [19]
		TRPO [126]	Considers KL Divergence → monotonic improvement.	
		PPO [128]	Similar to TRPO, but with simpler implementation.	
		ARS [83]	Back-propagation (BP) free, learns good linear policies.	
		ES [122]	Highly scalable, BP free, suitable for longer episode envs.	
		Off-policy policy gradient [29]	Estimates PG using importance sampling.	
		...	...	
Multi-agent RL		MADDPG [78]	CTDE, for both cooperative and competitive environments.	[158], [97]
		COMA [41]	Designed for cooperative MARL environments.	
		...	...	

*Color of shades: *Yellow*: On-policy RL algorithms, *Purple*: off-policy algorithms.

each building. Thus, the design of the reward functions and the consideration of coupling operational constraints may yield the control problem a competitive game.

In practice, one may need to control a cluster of buildings that are independently operated, i.e., not coupled through the electrical grid or any other coupling. The buildings in such case may have similarities such as construction type or occupancy patterns. To enhance the sample efficiency and reduce the data need of RL controller, transfer learning (TL) approaches allow for the utilization of learned control strategies in other buildings. TL is an approach that utilizes the learning from one task (called source task) to benefit another task (called target task) by utilizing the knowledge learned from the source task. There are few recent works with respect to TL for building control. In [154], the Q-network is decomposed into two parts: a forward Q-network which captures building-agnostic features, and a backward Q-network which captures action values in a supervised fashion. In the transfer process, the forward network is copied to the new building and backward network is trained with data in the new environment. In [72], TL algorithm transfers HVAC control policies and adjust according to geographical variations. In this way the policy trained in the source is used in the target domain and new policy does not need to train from scratch. TL approaches may utilize features of the source and target domains to parameterize the transfer mapping such as building size, number of thermal zones, construction materials, and ambient weather conditions. One challenge in TL is that there are times that finding the transferable domain source is a challenging task and also the current approaches may not always be able to calculate the mapping function between two domains [167]. Some recent research attempts to alleviate this source model misspecification, at the cost of greater compute requirementes [111, 32].

In this context, the emerging field of Semantic Interoperability proposes solutions that can facilitate system identification, agent training and transfer learning. With increasing volumes of available data, coming from different systems in different buildings designed by different engineers, extending or transferring a control algorithm to another application often requires considerable efforts. For instance, two engineers might decide to label

data coming from an air handling unit's discharge temperature sensor as *AHU-1-T* or *Unit1DisTemp*. Should these engineers exchange datasets, they would start an error-prone and time-consuming process of guessing which data point corresponds to which system. Semantic metadata standardizes the description of data, or its meaning, to ensure that different actors using different software platforms can still collaborate. Based on semantic metadata, schemas have been proposed that can provide a rich description of datasets and detailed hierarchical models of building systems. A review of metadata ontologies specific to buildings and building systems is available in [115]. Using this unified schemas, an RL agent will be able to recognize the mapping of training samples when it is transfered to a different building. Similarly, should an agent be trained to control a specific actuator, such as a damper actuator, if we leverage semantic metadata then controlling a different damper will not involve modifying the agent, the control sequence, or the controller's code.

5. What are the typical hardware, software, and data requirements to deploy RL in real-world buildings?

Ján Drgoňa, Draguna Vrabie

Reinforcement learning algorithms learn the control policies by interaction with the environment that represents the controlled system, i.e., a single building or a cluster of buildings. In general, there are two categories of RL methods for buildings with an environment represented by either: i) a simulation model of the building or ii) a real-world building. In the latter case, an RL agent needs to learn control policies by interacting with real-world buildings. This is an extremely challenging scenario because the controlled building needs supporting hardware (HW) and software (SW) infrastructure allowing for real-time monitoring and actuation of the heating, ventilation, and air conditioning (HVAC) equipment providing sufficiently large datasets to learn from (see Q8 for examples).

The HW infrastructure is typically deployed by adopting supervisory control and data acquisition (SCADA) architecture, which represents an industry standard in various fields such as process control and power systems [35]. In the built environment, the SCADA solution is often referred to as a building automation system (BAS) or building management system (BMS). Modern BMS is composed of four layers: a) field layer, b) automation layer, c) data management layer, and d) supervisory layer:

- The *field layer* represents all necessary sensors and actuators. The need for different sensor types varies from building to building based on the sophistication level of their overall control solution. Most common sensors provide measurements for quantities such as temperature, mass flow, CO2 concentration, humidity, illuminance, or motion. Modern actuators include controllable valves or pumps.
- The *automation layer* includes local controllers and data processing modules such as relays, programmable logical controllers (PLC), remote terminal units (RTU), or routers, and necessary cables interconnecting all devices.
- The *data management layer* is composed of industrial computers, which aggregate the communication streams from the automation layer, and human-machine interfaces (HMI) that provide access to the BMS for the operator. Advanced control methods such as RL could be deployed on this level.
- The *supervisory layer* represents the highest level of the BMS, which is responsible for running system optimization functionality. Most of the advanced control methods are commonly implemented at this level often in a server-client architecture utilizing external computational resources provided via the cloud as opposed to local computers. In either case, one needs to have sufficient computational resources for running the RL algorithms.

Since RL is an extremely versatile method, it can be in principle applied in a multitude of BMS levels. Primary examples include i) local setpoint tracking tasks typically deployed at the data management layer, or ii) energy and comfort performance optimization tasks typically deployed at the supervisory level.

From a SW perspective, one needs to have complete read/write access to the building's BMS in case the RL deployment is executed locally. Even though most of the BMS in common buildings are nowadays based on open communications protocols such as BACnet or Modbus, the vendors do not provide open access to these low-level communication streams for end-user of the BMS system. Hence, in case the building is already equipped with sensor

and actuator infrastructure, the current challenge is not technological. Instead, the challenge is in the prevalent business model of the contemporary BMS vendors, which do not provide access and tags to these data streams by default. In case the RL agent is to be deployed in a server-client setting with learning algorithms computed on the cloud, one needs to have sufficient cloud computing infrastructure purchased from the cloud providers. The advantages of this software as a service (SaaS) setting include the following: provider support, maintenance, scalable SW updates, and potential savings associated with outsourcing required computational resources. However, the downsides of SaaS include potential privacy issues, communication and network failues, as well as additional costs associated with purchasing necessary Internet of things (IoT) HW devices, and service fees.

One major hurdle in real-world deployment is the large data requirements of classical RL algorithms that can often lead to prohibitively long training times in terms of months or years [146, 97]. Due to these challenges, most of the deployment of RL agents in buildings has been done based on offline simulations using a digital environment model. This setup requires either data-driven system identification of a simplified building model [6, 38] or the availability of high-fidelity building emulators such as Energy+ or Modelica models [4, 11] which are used to generate the training data for the RL algorithms (see also Q7). Under the assumption of sufficient environment model accuracy, the RL training loop can be detached from the real building and performed offline. Then for real-world applications, one can just deploy trained RL policy in the existing BMS system. Pre-computing or pre-training the RL control policy offline [99] before deployment in a real building is computationally advantageous compared to real-time online optimization as used in MPC. Thus potentially reducing the need for deploying costly computational resources locally in the building by employing cloud computing services. However, offline training from the digital environment requires an accurate simulation model of the controlled building, thus imposing the same modeling expertise requirements and associated challenges as in the case of MPC. To alleviate this drawback, researchers have pre-trained RL agents using imitation learning from existing control strategies such as MPC [36] or RBC [100]. Others [18, 37] have integrated RL algorithms with domain-aware priors or system identification models of buildings to reduce the data requirements in online learning settings.

Finally, in conjunction with the HW, SW, and data requirements, successful deployment of RL in real-world buildings requires substantial domain knowledge on the side of the control system engineers and field installers. This includes a basic understanding of computer science, building physics, HVAC systems, control systems architectures, and engineering documentation, including piping and instrumentation diagrams (P&ID), which are typically not taught as a package within single university curricula. This requires updating current educational programs, focusing on alignment with industry needs.

6. How can RL be used for Human-Building Interaction?

Matias Quintana, Steven McCullough, June Young Park The developments of sensing, control, and computing enable to automate and optimize the operation of building systems. However, the main objective of such innovation has aimed at energy efficiency rather than human experience and comfort in buildings [107]. Given that we spend 80 – 90% of our time indoor, it is essential to shift our perspective toward occupants primarily and further balance between building energy performance and occupant comfort, i.e., occupant-centric building operation [110]. From the results from both IEA-EBC Annex 66 [155] and 79 [103], it is important to emphasize that understanding how occupants behave and interact with building systems is one of the key information to acquire for the optimal building operation. Therefore, an important learning objective in building automation and operation is to resolve the complexity of human-building interaction (HBI) [3, 9].

With the emerging paradigm of personal comfort models [58], various machine learning methods have been implemented to solve this [84]. RL is also one of the promising algorithms that researchers have utilized to learn occupant behavior in buildings. Due to its interactive nature and model-free learning approach, RL is particularly viable to discover the knowledge of HBI [94].

Here we review notable works where researchers used RL as a main learning algorithm to discover the knowledge of HBI in real building environments. As mentioned in Q1, the seminal contribution of

deploying RL in a residential setting was the Neural Network House that prototyped an adaptive control of home environments [92]. This deployment showcased the potential capabilities of adaptive control of living environments. The adoption of RL in building controls was particularly successful with the recent trend of occupant-centric building controls [110]. For example, both *LightLearn* and *HVACLearn* implemented an RL algorithm using customized interfaces, monitoring indoor environments and user interactions to calculate the optimal policy for occupant-centric building controls [106, 109]. Cheng et al. used a Q-learning based system to control lighting and blind based on the satisfaction vote from a web interface [21]. Lei et al. also implemented a Q-learning algorithm, specially branching dueling Q-Networks, and tabular-based personal in a real office building [63]. Similarly, Esrafilian-Najafabadi et al. used double Q-learning while considering dynamic occupancy patterns [40]. Another recent contribution is *ComfortLearn* environment which provides a more accurate representation of occupant's thermal comfort without the need for intense simulation [116]. These developments showed that using simple HBI combined with RL, comfort prediction and assessment can be greatly improved in both accuracy and efficiency.

Despite the significant research on RL in the built environment such as demand-response, building-to-building, and building-to-grid interaction [139], there is little work on RL based HBI especially for occupant-centric building controls. The main challenges are 1) Unavailability of data, in diverse buildings and locations, to validate RL [34]. 2) Occupants' unwillingness and inability to interact with building systems that implement RL based controllers. This would be related to privacy and security concerns. Primarily, because some interfaces are hard (or impossible) to use [28]. 3) Building operators are unfavored of new installation on their building control, especially trail-and-error based RL approach. Current automation systems and sensor technologies do not exceed simple queries from fault messages [50]. 4) Evaluation and metrics of HBI are not well defined. There is a lack of consensus on how to evaluate building performance considering the occupant [102]. As mentioned in Q1, in the RL learning structure, the reward signal should be properly defined as it plays a critical role to address the control problem. 5) Building systems are not well developed to feed a novel RL method (i.e., granular measurements of HVAC and environmental parameters) and have a real-time interaction with human signal. In most traditional building automation systems, data is only collected to ensure essential operating activities, and occupant data is primarily obtained from Post-Occupancy Evaluation (POE) processes [50].

While some of the problems mentioned above require better communication and collaboration with building operators, most of them can be seen as data centric. To address this, solutions can be split into two not mutually exclusive approaches: 1) acquiring new data and 2) exploiting existing data. More human-based experiments, specially across different regions with various factors (e.g., climate, building type, etc) could alleviate the transferability of test data to multiple buildings [28]. Additional steps towards acquiring new data could incorporate various new streams of data such as BMS, IoT, and occupant-centric [87]. Paradigms like Digital Twins are already converging different data modalities and looking into data interoperability [62]. Nevertheless, these efforts would only reach wide usage if the community aims for open datasets. These collections of different case studies and field experiment can be use concurrently as a main database and future research could keep appending to push future works [42, 33]. On the other hand, existing open datasets, while limited, could be exploited to increase their utility. Building simulation is very prevalent in building models but when used in tandem with PMV it fails to capture the HBI component. Agent-based modeling have started to gain momentum in other areas of the built environment beyond transportation research and can be used alongside building models [116]. Moreover, increasing the available data in terms of sample size and variability is actively explored in thermal comfort models [27, 116]. The resulting datasets could then be used with building models for new unseen scenarios. Finally, incorporating this human-generated data and human behaviour into the RL formulation, specifically in the reward design, would need to be standardised too. Existing approaches rely on comfort different comfort proxies. One is a fixed temperature band where occupants are assumed to remain comfortable. The human-comfort portion of the reward is calculated as an arbitrary number if the indoor temperature is outside a given range [51] or a variable number calculated by the difference between the indoor operative temperature and desired set-

point [40]. Another common approach is to factor occupancy, whether there are any occupants in the indoor space, as a binary variable [40, 63]. Directly using occupant-generated data such as comfort label is not widely adopted, but some existing efforts consider human interactions such as thermostat interactions [109]. A more inclusive approach would include the human behaviour, e.g., thermal comfort label, into the reward function through some numerical mapping or aggregation from all current occupants.

Consistency is key to getting a reliable and efficient RL model for HBI. Research has shown that occupants will take behavioral actions to improve their own thermal comfort when the thermal preferences of other occupants in the same space are not known to them [31]. An occupant's ability to change their own thermal comfort range can impact an RL model's ability to correctly decipher their personal comfort model. Considering a heterogeneous workplace, this has an even greater effect when considering the differing thermal preferences of those with differing ages and genders. These preferences can also change over time as well [22]. Occupant behavior becomes increasingly more complex within a shared space with more behaviors between occupants. Research has shown it is difficult to model the interconnected actions that occupants can have between shared spaces with traditional decision trees. RL methods can have similar issues in this regard as well. For instance, a variety of behaviors can occur between occupants that affect their comfort levels, a RL model may interpret this incorrectly the next time these conditions are met again but without these occupant behaviors, resulting in energy waste and discomfort [82]. Personalized RL models for each occupant would be able to generate a space model that can more accurately provide the best comfort.

However, with multiple occupants, thermal fairness is also a concern as minority comfort preferences will never be chosen when using majority rule or the mean temperature preference inflicting unfairness upon some occupants [129]. Models for generalized spaces can be similarly 'unfair' in that context if the reinforcement learning is weighted in the direction of the majority from feedback. As thermal comfort is a very subjective feeling to building occupants, another issue is consistency in thermal comfort feedback. Since existing survey methods provide directional adjustment for temperature (i.e., hotter, or colder) and not exact temperature preferences for each occupant there can be a concern with the intensity of these direction changes between occupants [141]. With RL requiring some form of feedback there will be some uncertainty in whether one occupant's definition of a preference is the same as another occupants. RL models can use smaller steps to account for this at the expense of a longer training period or more surveys.

7. What are available environments, libraries and datasets to support studies of RL for buildings?

Han Li, Silvio Brandi, Giuseppe Pinto, Tianzhen Hong

RL has been exceptional in domains including board games [125], computer games [89, 10] and robotics [166]. Besides the advancement in DRL algorithms, two important commonalities behind the successes are: (1) the problems have well-known environmental settings and rules that thoroughly describe the state transition probabilities in their MDP, and (2) on-policy sampling is safe and economic. In cases like games where the environment is fully understood, the MDP can be simulated without uncertainty and the RL agent can be trained entirely in a virtual environment. For scenarios where the environment and state transition probabilities are approximated, such as robotics and autonomous driving, control policies can be trained in virtual environments and then transferred to real-world scenarios, which is known as *sim2real*. In addition to virtual environments, datasets collected under existing policies are very useful for offline RL training and imitation learning [119] in situations where sampling becomes expensive and risky, such as autonomous driving.

In the building domain, training and deploying RL controllers from scratch in real-time is prohibitive for two reasons. First, buildings have stringent requirements for indoor environments, which prevent RL agents from thoroughly exploring the state and action spaces. Second, buildings have slow dynamics and sparse rewards, which lead to expensive and inefficient sampling.

In this section, we highlight the existing virtual environments, datasets, and software libraries that can be used to bootstrap the development and testing of RL controllers for buildings applications. The overview of how those parts can help training RL building controllers is shown in Figure 3.

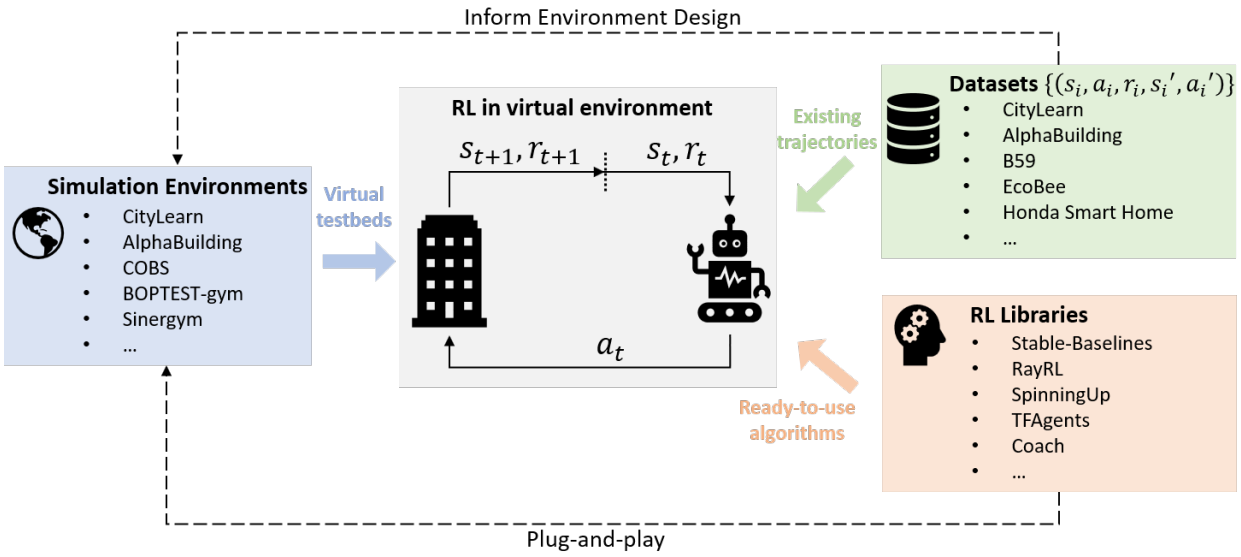


Figure 3: Simulation environments, datasets, and libraries for RL in buildings

7.1. Computational Environments

As indicated in Question 5, due to the challenges of implementing RL in real buildings, most researchers chose to use simulated environments for control policy development. While building performance modeling started decades ago, it was only in the past few years that the research community started to adopt it in creating virtual environments to train RL control policies for buildings. However, there is already a diverse list of environments created for various purposes at different scopes. Depending on the scopes of application, most existing environments focus on single building control, while AlphaBuilding ResCommunity [145] and CityLearn [138] support multi-building coordinations at community scale. In terms of the control levels, most existing environments are supervisory control at system-level, meaning the control actions are usually setpoint adjustment rather than component level controls. Although virtual environments provide great opportunities for offline RL training, it is always a non-trivial challenge when transferring and tuning the control policy into real buildings due to distribution shifts. To mitigate this issue, efforts have been dedicated to active data collection and model calibration[159], so that the virtual environments could approximate the state transition as closely as possible. The most commonly used building and system dynamics models in existing environments are developed with EnergyPlus and Modelica where the goals are to develop system level controllers in single buildings. In CityLearn and AlphaBuilding ResCommunity where the goal is community level load coordination, less computationally expensive reduced-order models are used. Environments like the BOPTEST-gym [4] and Advanced Controls Test Bed (ACTB) [85] allow different types of models to emulate the system dynamics. The coupling between building simulation and RL algorithms are mainly realized in three approaches: Functional Mock-up Interface (FMI) [96], Building Controls Virtual Test Bed (BCVTB) [151], and the EnergyPlus Python API [1]. Researchers adopt standard software frameworks and APIs to develop virtual environments for buildings. Most existing virtual environments adopted the OpenAI gym [13] to encapsulate the building simulation. Recently, researchers have been trying to make the virtual environments more realistic and extend to more complex operation scenarios. For example, FlexDRL is a simulation environment for RL algorithm development for office buildings with Distributed Energy Resources (DERs) [134]. In addition to traditional HVAC system control, it allows the RL controller to interact with electric battery and solar PV systems. In COBS [140], stochastic occupancy simulation is included alongside building simulation, which adds occupant-related variabilities and enables occupant-centric control policy development. A summary of existing popular virtual

environments for buildings can be found in Table 3.

7.2. Datasets

As previously mentioned, datasets collected under existing policies can support the pre-training of RL agents before their deployment in real buildings, providing useful experience to speed-up the online learning process. The advantages of employing existing datasets during the pre-training phase lie into an increase of generalizability and stability of RL agents, while reducing the time needed to converge towards a near optimal policy. In this context, even simulated or synthetic datasets can be effectively employed to provide useful knowledge to initialize the control policy before further training or direct implementation of the agent. However, their application must be carefully considered, as a dataset must include a perfect overlap between the state-action-reward tuples for a given control problem and the available information. As an example, it is not enough to have information related to the energy consumption of a building or its main sub-systems, but it is necessary to include information of the indoor environment, operational parameters of the HVAC system, and control actions. This becomes even harder to obtain when multiple buildings are considered, as the dataset should contain enough information to allow the controller to understand the underlying dynamics and relationships between buildings. Furthermore, the quality of the dataset directly influences the performance of the deployed agent, that may suffer from performance decrease if the dataset employed during pre-training does not match the dynamics of the real environment.

The practice of sharing datasets to pre-train RL agents is fairly recent, most open-source datasets concerning building energy management data do not contain enough information to be used during the pre-training phase of RL agents. In this context, the main open-source datasets authors found available to pre-train RL control policies are:

- AlphaBuilding dataset [68]: includes synthetic building operation dataset for commercial office buildings in three U.S. climates.
- LBNL Building 59 dataset [81]: includes a three-year monitored dataset of a medium-sized office building constructed in 2015 in Berkeley, California.
- CU-BEMS dataset [113]: includes 18 months of one-minute interval monitored data of detailed building operation including electricity consumption and indoor environmental measurements of a seven-story office building in Bangkok, Thailand.
- B2RL dataset [73]: includes data collected from both real world building energy management systems and simulation environment for commercial and school buildings.
- The CityLearn Challenge 2020 dataset [137]: includes hourly one-year synthetic data from nine multifamily and commercial buildings in four United States climate zones.
- The CityLearn Challenge 2021 datasets [93, 98]: includes four-year hourly synthetic data from active storage control in nine multifamily and commercial buildings modeled for New Orleans, Louisiana.
- The CityLearn Challenge 2022 dataset [101]: includes hourly one-year preprocessed data from 17 single-family homes in the Sierra Crest Zero Net Energy community in Fontana, California.

7.3. RL Libraries

RL controllers can be built from scratch using any programming language. However, using libraries specifically designed for RL development can simplify the development process and provide state-of-the-art solutions. A variety of libraries for RL development are available in Python including Stable-Baselines[117], Keras-RL[114], Tensorforce[60], TFAgents[48], Coach[14], and RLLib[70]. Besides Python, the Reinforcement Learning Toolbox supports RL implementation in MATLAB and Simulink, the Reinforcement Learning ToolKit (RLTK)[2] offers fast and efficient implementation of popular algorithms in C++, and the swift-rl library supports the implementation of common algorithms in Swift.

In summary, multiple environments have been developed by different groups for different use cases and with different approaches (e.g., building type, energy systems, climate zones, co-simulation techniques). A limited number of environments integrate the virtual simulation environment with real building/facility operations or controls. The interoperability or standardization between these environments needs further development. Although

simulated datasets are becoming more available, datasets of real buildings with detailed monitored data points and metadata are still very limited with adequate coverage and resolution to support RL applications.

8. What are examples of real-world implementations of RL for building energy management?

Draguna Vrabie, Helia Zandi

Reinforcement learning is a data-driven approach, specifically model-free RL, that needs a large number of interactions with the environment to learn better policies. Also, varying preferences, overriding set points, delayed control, invalid state information, missing data, etc. are a few other challenges related to the real-world implementations of RL in a built environment [97]. Due to these challenges, the current practice in this space is to use offline training in the simulated environment and deploy a trained RL model in the real-world built environment. Kurte et al., 2020 [61] trained a Deep-Q-Network (DQN) model to control Heat Ventilation and Air Conditioning (HVAC) system in a simulated environment and deployed the pre-trained model in a single family 2 zone research house located in Knoxville, TN, USA. In a similar research, Naug et al., 2020 [95] proposed a continual learning approach to relearn the policy in non-stationary building operations. They deployed the new RL model and relearning framework to control the indoor climate of a large three-story building at Vanderbilt University, Nashville, TN, USA campus.

Chen et al., 2019 [18] mentioned the fact that the offline training of RL agents requires additional efforts in developing simulated environments. They proposed Gnu-RL, an approach that used a differential MPC approach and imitation learning to pre-train RL agents using historical data. Once deployed the model used the learned policy to control the operations as well as continued to improve the policy. The Gnu-RL showed 16.7% reduction in cooling demand when deployed in a conference room for a period of three weeks. Authors in [80] have reported successful real-world deployment of deep RL agents for control cooling systems in two commercial buildings with 9% to 13% energy savings. Based on these experiments, the authors summarize a set of open challenges, including control policy evaluation, learning from limited data, offline learning, dealing with non-stationary dynamics with multiple time scales, the satisfaction of constraints, and handling different operating scenarios. Park et al. reviewed the real-world implementations of occupant-centric building controls [110]. One important finding is that current information and communication technological (ICT) environment is not supportive enough to implement advanced control algorithms (e.g., RL) in terms of sensor integration, memory, learning, communication, and actuating relevant building systems. To overcome this challenge, some researchers utilized open source devices (e.g., Arduino, Raspberry Pi) to implement advanced control algorithms. For example, *LightLearn* is a Q-Learning based occupant-centric control system which utilize Raspberry Pi as a main control node to interact with occupants in a form of monitoring light switch usage and daylight level [106]. This approach was successfully implemented in academic offices at Austin, TX. As the learning complexity would increase, the existing BAS computing platform should provide such abilities for building control researchers.

It is clear from these examples, that real-world implementations of RL can differ whether they are research experiments in real buildings, or actual deployments in real buildings. For instance, [106] uses research grade tools to experiment in a real building, which may not be suitable for actual buidling operation due to reliability concerns. On the other hand [61] and [18] implements the RL agent in a BMS that runs an experimental room/home with real occupancy.

To summarize, real-world implementation of RL for built environments is still in a nascent stage. More research is required to address various challenges associated with real-world RL as elaborated in the next section.

9. What are challenges in real-world implementation of RL in the built environment?

Ahmed Zamzam, Sourav Dey, Tianzhen Hong, Kuldeep Kurte, Ján Drgoňa, Draguna Vrabie, Helia Zandi

The challenges of real-world implementation of RL in the built environment are categorized into four main areas, a) learning challenges, b) infrastructure, c) cost and benefit, and d) safety, security, and trust.

Learning challenges.

1. *Sample inefficiency and initial training instability*: RL requires a lot of training data, sometimes in the scale of years, for it to have a near-optimal policy. The exploratory process in the initial stages might is not acceptable for the building control managers and the occupants. Moreover, the initial behavior can be potentially unstable or violate equipment constraints which can cause damage to the mechanical components. Thus, they cannot be directly applied to the building and need a physics-informed solution, additional safety filters and fallback strategies, pre-training, or transfer learning from another trained agent to a similar building environment.
2. *State-space complexity*: The complexity of the states can increase exponentially for a large building with multiple inputs and outputs, a phenomenon known as *the curse of dimensionality*. This may require using larger amounts of data to train the RL controllers which increases the data collection and training times, or resorting to reduced-order models which may lead to gross errors when applying the controllers to the real buildings. In addition, the dynamical systems of any building can have different state-space representations with possibly varying dimensions. This is more pronounced in RL controllers which can have different levels of observability and forecasts availability. Thus, it is required to make choices regarding the state-space representation of the buildings to be controlled based on data availability and the performance targets. In addition, this issue may hinder transferring RL controllers between buildings; refer to Q4 for more discussions on this.
3. *Reward shaping*: RL performance and learning are sensitive to the shaping of the reward function. Sometimes from a purely cost-based approach if thermal comfort cost is usually too high compared to the energy costs, and this causes the RL agent to place very little importance on the energy costs.
4. *Delayed feedback*: Some environments do not have immediate consequences and exhibit delayed feedback to control actions. This is sometimes difficult to learn and may take a large number of actions to estimate the long-term consequences of an action. A careful trade-off between exploration and exploitation is essential to converge to address this issue.
5. *Transfer learning difficulty*: Transfer learning in the context of building controllers with RL is difficult as buildings and their systems are widely varied and require different states and control spaces. A different building cannot always adapt a trained RL agent with a particular input and output architecture.

Infrastructure.

1. *Lack of real-time sensing and control*: Most conventional buildings lack open access to real-time sensing that would allow for easy extension of the implemented control system. Most current energy management control systems (EMCS) in buildings are closed systems that depend on vendors to program or overwrite the control logic. This setup makes it difficult to take external control signals from RL controllers.
2. *Coverage and resolution of building data*: Most buildings may not collect or archive the necessary data points for training the RL controllers. Quite often, the available data lack the time resolution or the resolution of the individual components in the building data for proper training of the RL algorithm. For example, a whole building's energy use data is not beneficial without the plug loads and internal gains for an RL agent controlling the HVAC system.

Costs and benefits.

1. *Costs*: Additional costs of sensing, data collection, and system integration to enable the RL controls may pose a significant burden to users and stakeholders. Reducing the costs through best practices, training, and standardization can reduce the costs of scaling up the adoption of RL controls. Expertise and efforts required for building operating staff to support and maintain RL controllers can be extra costs that need to be considered.
2. *Benefits*: Energy and non-energy benefits should be accounted for. Energy benefits including energy savings and utility cost reduction are straightforward to include, but non-energy benefits, including GHG emissions reduction, peak demand decrease, and improvements to occupant comfort, health, and well-being are hard and usually not considered.

RL controls have the potential to improve the IEQ in addition to energy benefits through the multi-criteria award/cost function.

Safety, security, and trust.

1. *Safety and Security*: Ensure the RL controllers are safe, i.e., the control policy recommended by RL would not lead to abnormal damage to equipment or lead to an extreme indoor environment for occupants. When RL controllers fail, the default controllers should be automatically deployed. The RL controllers should be designed and tested to be cyber-secure. A safety layer (middleware) can be helpful to sit between the RL controllers and the building EMCS - the building operator or manager can choose the automatic mode for RL controllers to directly communicate with the EMCS or disable it due to unexpected reasons regarding safety or security issues.
2. *Explainable AI*: It is essential to ensure the RL control policy can be explained to building operators or managers in human language. Opaque decisions or recommendations may lead to a lack of trust in the RL deployment.

For more in-depth reviews, analysis, and case studies of the RL challenges in building controls, we refer the reader to [146, 97, 80]. While the real-world challenges associated with generic RL methods can be found in [39].

10. What are open research questions and possible future research directions?

Zoltan Nagy, Kingsley Nweye, Mario Berges

In addition to the technical challenges in RL discussed in Q9, there are also some more fundamental research questions and directions that can be explored in the future.

In Q6, we explored RL for human-building-interaction, and showed some potential applications and recent works. From a more fundamental perspective, it helps to recall that ultimately, we control buildings to optimize social/human objectives (e.g., comfort, productivity, health, etc.). And more research is needed on how best to incorporate these into the RL framework. To date, most work is balancing energy and comfort (which is modeled very crudely). But even within this limited view, there are various different ways of balancing this trade-off: hard constraints that need to be imposed on the actions, or soft constraints made evident in the design of the reward function, etc. This begs the question of what the *right* approach is for each situation. Moreover, as we measure more detailed occupant behavior, and use it to refine modeling of thermal comfort and incorporate other models of human behavior and its synergistic relationship with building behavior (e.g., not just how different policies map to expected rewards in terms of PMV-type mean comfort, but individual comfort at higher spatio/temporal resolutions, as well how the policy maps to rewards in terms of worker productivity, sick days, etc.) what innovations are needed to accommodate these changes?

Since buildings are engineered systems, there is a wealth of information that can be accessed about their form, function and behavior in the digital assets that are generated throughout their design and construction process (e.g., building information/energy models, sequence of operations documents, etc.). Model-based RL approaches, in particular (though model-free approaches too if one can leverage these digital assets to quickly generate simulation environments on which to train them) have a lot to benefit from these digital assets. For instance, in Chen et al. (Gnu-RL) a simple lumped parameter model of the thermodynamics of the building showed a drastic improvement in sample efficiency compared to model-free approaches. Arguably, significantly better results could be achieved if these thermodynamic models were of higher order and could account for the system's multiple time constants. Given the diversity of the building stock, this could only be achieved if we had scalable methods to synthesize these reduced-order models directly from available digital assets.

As stressed in Q2, high expectations arise from merging MPC and RL approaches in stochastic applications (e.g. buildings) enabling both learning and constraints handling. Both MPC and RL pursue the same goal while following radically different mechanisms. Machine learning and model-based adepts can collaborate at the level of control or at the modeling level. Although hybrid methods for HVAC control are very promising, it is a nascent research field. Collaboration at the modeling level is more apparent but has been less explored. Grey-box modeling can be seen as supervised learning which is a type of machine learning. Grey-box modeling is very popular for its application in MPC because it has several advantages. However, a trigger-

ing issue within the practice of grey-box modeling for MPC is that it is severely limited when scaling to detailed physical models because of the non-convex nature of its parameter estimation process. Dozens of physically meaningful parameters can be estimated when using this technique. Remarkably, machine learning techniques commonly estimate thousands of parameters for their deep neural networks, indicating that the building control community has a lot to learn from machine learning in this aspect. In this realm are also recent advances in physics constrained machine learning (PIML) or physics constrained neural networks (PINNs). They offer a novel pathway whereby models can be learned from data but with the added constraint of maintaining physical consistency (energy conservation, etc), which reduces data requirements.

In Q4 and Q5 we stressed the need for appropriate communication infrastructure, especially when considering coordinating loads between buildings. In addition to the physical infrastructure itself, this will require development of novel privacy aware solutions, especially when third-party providers, often cloud solutions, are accessing and storing a building's data. This is closely related to the need for a proper data sharing practice, which has to start with the use of a common semantic descriptors.

In Q8 we highlight the paucity of real-world implementation of RL in live building systems and in Q9 identity explainable AI as one of the trust challenges facing RL deployment. Thus, what tools, technologies and methodologies exist to provide descriptions of learned RL policies using a language that is commonplace to most stakeholders and varying expertise levels in building control and asset management that can help accelerate the adoption of RL real-world building control problems? The work in [153] has explored rule extraction of RL policies to reduce the communication complexity of the underlying learned policy to simple if-else statements as used by the simpler but more widespread RBC.

Conclusions

In this paper we explored ten questions pertaining to the implementation of reinforcement learning (RL) algorithms for building energy management. After providing a general introduction and overview, we discussed practical, real-world challenges, available datasets and libraries, and real-world RL implementations. We concluded with possible future research directions. As reducing energy demand and associated greenhouse gas emissions from the built environment is fundamental to address the impacts of climate change, advanced control algorithms, such as RL, play a pivotal role. This paper serves as a valuable resource for both early career and experienced researchers, as well as practitioners and policy makers offering a deeper understanding of opportunities and challenges in RL to support building decarbonization.

Acknowledgments

PNNL authors are supported by the U.S. Department of Energy, through the Energy Efficiency and Renewable Energy, Building Technologies Office under the "Advancing Market-Ready Building Energy Management by Cost-Effective Differentiable Predictive Control" project. PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL0-1830. LBNL authors are supported by the U.S. Department of Energy, through the Energy Efficiency and Renewable Energy, Building Technologies Office under Contract No. DE-AC02-05CH11231. The National Renewable Energy Laboratory (NREL) authors are supported by the U.S. Department of Energy, through the Energy Efficiency and Renewable Energy, Building Technologies Office. NREL is operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract DE-AC36-08GO28308. The NUS authors are funded by the NUS-based Singapore MOE Tier 1 Grant titled Ecological Momentary Assessment (EMA) for Built Environment Research (A-0008301-01-00). Part of this research was conducted at the Future Cities Lab Global at Singapore-ETH Centre. Future Cities Lab Global is supported and funded by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme and ETH Zurich (ETHZ), with additional contributions from the National University of Singapore (NUS), Nanyang Technological University (NTU), Singapore and the Singapore University of Technology and Design (SUTD). The KU Leuven authors acknowledge the financial support through the TECHPED - C2 project (C24M/21/021). The TECHPED project investi-

gates TECHnically feasible and effective solutions for Positive Energy Districts.

Appendix

Name	Latest Release	Summary	Building Type	Observation Space	Action Space	State Transition Simulation	Multi-Agent RL Support	Link to Repository
CityLearn [138]	2023	An open source OpenAI Gym environment for the implementation of Multi-Agent Reinforcement Learning (RL) for building energy coordination and demand response in cities.	various	temporal info, weather condition, PV generation, unmet setpoints, electricity consumption, carbon emissions	charging /discharging of DHW, CW, and electric battery	reduced-order model	Yes	https://github.com/intelligent-environments-lab/CityLearn
AlphaBuilding ResCommunity [145]	2021	A multi-agent OpenAI Gym Environment for Thermally Controlled Loads (TCLs) Coordination. The building characteristic distribution is extracted from more than 80,000 households in the United States.	residential	ambient temperature, cloud coverage, hour of day, indoor temperature of arbitrary number of TCL	on/off for arbitrary number of TCL	reduced-order model	Yes	https://github.com/LBNL-ETA/AlphaBuilding-ResCommunity
AlphaBuilding MedOffice [143]	2021	A realistic OpenAI Gym environment that can be used to train, test and benchmark controllers for medium size office (1AHU + 9VAV boxes)	commercial - office	ambient temperature, cloud coverage, hour of day, indoor temperature of 9 zones, HVAC energy consumption	9 VAV and 1 AHU outlet temperature setpoint	EnergyPlus model	Yes	https://github.com/WalterZWang/AlphaBuilding-MedOffice
COBS [160]	2022	An open-source, modular co-simulation platform for developing and comparing building control algorithms, which integrates various simulators and agent models with EnergyPlus and supports fine-grained and occupant-centric control of building subsystems.	various	flexible - users can specify state variables in addition to the default as long as they are EnergyPlus output variable.	flexible - users can specify actions supported by EnergyPlus EMS.	EnergyPlus model	No	https://github.com/sustainable-computing/COBS

Name	Latest Release	Summary	Building Type	Observation Space	Action Space	State Transition Simulation	Multi-Agent RL Support	Link to Repository
BOPTEST-gym [4]	2022	An OpenAI-Gym environment for the BOPTEST framework that exposes the "control points" of building models using a standard, familiar API that allows control algorithms to interact with the models as if they are physical buildings.	various	flexible - users can overwrite the observation space with the available sensors in a BOPOTEST emulator building model	flexible - users can overwrite the action space with the available actuators in a BOPTEST emulator building model	various	No	https://github.com/ibpsa/project1-boptest-gym
Gym-Eplus [165]	2019	An open source Open-AI Gym environment for RL testing using building energy simulations.	various	flexible - users can specify state space from available EnergyPlus output	flexible - users can specify action space from available EnergyPlus actuators	EnergyPlus	No	https://github.com/zhangzhizza/Gym-Eplus
Sinergym [55]	2023	An open source OpenAI Gym environment for wrapping simulation engines for building control using deep reinforcement learning	various	flexible - users can specify state space from available EnergyPlus output	flexible - users can specify action space from available EnergyPlus actuators	EnergyPlus	No	https://github.com/ugr-sail/sinergym
RayRLlib EnergyPlus [46]	2023	An example single zone testbed to train a control policy using Ray RLlib and EnergyPlus Python API	commercial - office	outdoor temperature, zone air temperature, zone CO2 concentration, heating and cooling setpoints	supply air temperature setopint	EnergyPlus	No	https://github.com/airboxlab/rllib-energyplus
Energym [124]	2021	An open source building simulation library designed to test climate control and energy management strategies with several building models embeded.	various	varied by building model	varied by building model	EnergyPlus + Modelica	No	https://github.com/bsl546/energym

Name	Latest Release	Summary	Building Type	Observation Space	Action Space	State Transition Simulation	Multi-Agent RL Support	Link to Repository
rl-testbed for energyplus [91]	2023	A Reinforcement Learning Testbed for Power Consumption Optimization with several building models embeded.	commercial - data-center	varied by building model	varied by building model	EnergyPlus	No	https://github.com/IBM/rl-testbed-for-energyplus
FlexDRL [135]	2022	An open-source simulation environment for DRL algorithm development for office building with distributed energy resources.	commercial - office	weather conditions, zone air temperature, lighting, plug-loads, fan power, solar irradiation, actual hot water and chilled water supply temperature, battery state of charge, total power	AHU supply air temperature and flow rate, chilled water and hot water supply temperature, on/off solar shading, lighting, battery charging and discharging, solar PV shading	EnergyPlus + Modelica	No	https://github.com/LBNL-ETA/FlexDRL

Table 3: Existing popular virtual environments for building controls

References

[1] EnergyPlus Python API — EnergyPlus Live Documentation 0.2 documentation.

[2] Reinforcement Learning Toolbox.

[3] Alavi, H. S., Churchill, E. F., Wiberg, M., Lalanne, D., Dalsgaard, P., Fatah gen Schieck, A., and Rogers, Y. Introduction to human-building interaction (hbi) interfacing hci with architecture and urban design, 2019.

[4] Arroyo, J., Manna, C., Spiessens, F., and Helsen, L. An OpenAI-Gym Environment for the Building Optimization Testing (BOPTEST) Framework. In *Proceedings of the 17th IBPSA Conference* (2021).

[5] Arroyo, J., Manna, C., Spiessens, F., and Helsen, L. Reinforced model predictive control (RL-MPC) for building energy management. *Applied Energy 309* (3 2022).

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

[7] ASHRAE. Ashrae guideline 36-2018 high-performance sequences of operation for hvac systems, 6 2018.

[8] Azuatalam, D., Lee, W.-L., de Nijs, F., and Liebman, A. Reinforcement learning for whole-building hvac control and demand response. *Energy and AI 2* (2020), 100020.

[9] Becerik-Gerber, B., Lucas, G., Aryal, A., Awada, M., Bergés, M., Billington, S. L., Boric-Lubecke, O., Ghahramani, A., Heydarian, A., Jazizadeh, F., et al. Ten questions concerning human-building interaction research for improving the quality of life. *Building and Environment 226* (2022), 109681.

[10] Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[11] Blum, D., Arroyo, J., Huang, S., Drgoňa, J., Jorissen, F., Walnum, H. T., Chen, Y., Benne, K., Vrabie, D., Wetter, M., and Helsen, L. Building optimization testing framework (boptest) for simulation-based benchmarking of control strategies in buildings. *Journal of Building Performance Simulation 14*, 5 (2021), 586–610.

[12] Brandi, S., Gallo, A., and Capozzoli, A. A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings. *Energy Reports 8* (2022), 1550–1567.

[13] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[14] Caspi, I., Leibovich, G., Novik, G., and Endrawis, S. Reinforcement learning coach, Dec. 2017.

[15] Chen, B., Cai, Z., and Bergés, M. Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable MPC policy. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 316–325.

[16] Chen, B., Cai, Z., and Bergés, M. Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable mpc policy. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, 2019), pp. 316–325.

[17] Chen, B., Cai, Z., and Bergés, M. Gnu-rl: A practical and scalable reinforcement learning solution for building hvac control using a differentiable mpc policy. *Frontiers in Built Environment 6* (2020), 562239.

[18] Chen, B., Cai, Z., and Bergés, M. Gnu-rl: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), 316–325.

[19] Chen, B., Donti, P. L., Baker, K., Kolter, J. Z., and Bergés, M. Enforcing policy feasibility constraints through differentiable projection for energy optimization. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems* (2021), pp. 199–210.

[20] Chen, B., Yao, W., Francis, J., and Bergés, M. Learning a distributed control scheme for demand flexibility in thermostatically controlled loads. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* (2020), IEEE, pp. 1–7.

[21] Cheng, Z., Zhao, Q., Wang, F., Jiang, Y., Xia, L., and Ding, J. Satisfaction based q-learning for integrated lighting and blind control. *Energy and Buildings 127* (2016), 43–55.

[22] Choi, J., Aziz, A., and Loftness, V. Investigation on the impacts of different genders and ages on satisfaction with thermal environments in office buildings. *Building and Environment 45*, 6 (2010), 1529–1535.

[23] Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., Red Hook, NY, 2018, pp. 4754–4765.

[24] Costanzo, G. T., Iacovella, S., Ruelens, F., Leurs, T., and Claessens, B. J. Experimental analysis of data-driven control for a building heating system. *Sustainable Energy, Grids and Networks 6* (2016), 81–90.

[25] Costanzo, G. T., Iacovella, S., Ruelens, F., Leurs, T., and Claessens, B. J. Experimental analysis of data-driven control for a building heating system. *Sustainable Energy, Grids and Networks 6* (2016), 81–90.

[26] Dalamagkidis, K., Kolokotsa, D., Kalaitzakis, K., and Stavrakakis, G. S. Reinforcement learning for energy conservation and comfort in buildings. *Building and environment 42*, 7 (2007), 2686–2698.

[27] Das, H. P., and Spanos, C. J. Conditional Synthetic Data Generation for Personal Thermal Comfort Models, Mar. 2022.

[28] Day, J. K., McIlvennie, C., Brackley, C., Tarantini, M., Piselli, C., Hahn, J., O'Brien, W., Rajus, V. S., De Simone, M., Kjærgaard, M. B., Pritoni, M., Schlüter, A., Peng, Y., Schweiker, M., Fajilla, G., Becchio, C., Fabi, V., Spigliantini, G., Derbas, G., and Pisello, A. L. A review of select human-building interfaces and their relationship to human behavior, energy use and occupant comfort.

Building and Environment 178, May (2020), 106920.

[29] Degris, T., White, M., and Sutton, R. S. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839* (2012).

[30] Dempster, M. A., and Leemans, V. An automated fx trading system using adaptive reinforcement learning. *Expert systems with applications 30*, 3 (2006), 543–552.

[31] Deng, Z., and Chen, Q. Artificial neural network models using thermal sensations and occupants' behavior for predicting thermal comfort. *Energy and Buildings 174* (2018), 587–602.

[32] Didden, D., Wiesé, N., Kazmi, H., and Driesen, J. Sample efficient reinforcement learning with domain randomization for automated demand response in low-voltage grids. *IEEE Journal of Emerging and Selected Topics in Industrial Electronics 3*, 4 (2022), 891–900.

[33] Dong, B., Liu, Y., Mu, W., Jiang, Z., Pandey, P., Hong, T., Olesen, B., Lawrence, T., O'Neil, Z., Andrews, C., Azar, E., Bandurski, K., Bardhan, R., Bavaresco, M., Berger, C., Burry, J., Carlucci, S., Chvatal, K., De Simone, M., Erba, S., Gao, N., Graham, L. T., Grassi, C., Jain, R., Kumar, S., Kjærgaard, M., Korsavi, S., Langevin, J., Li, Z., Lipczynska, A., Mahdavi, A., Malik, J., Marschall, M., Nagy, Z., Neves, L., O'Brien, W., Pan, S., Park, J. Y., Pigliautile, I., Piselli, C., Pisello, A. L., Rafsanjani, H. N., Rupp, R. F., Salim, F., Schiavon, S., Schwee, J., Sonta, A., Touchie, M., Wagner, A., Walsh, S., Wang, Z., Webber, D. M., Yan, D., Zangheri, P., Zhang, J., Zhou, X., and Zhou, X. A Global Building Occupant Behavior Database. *Scientific Data 9*, 1 (June 2022), 369.

[34] Dong, B., Markovic, R., Carlucci, S., Liu, Y., Wagner, A., Kim, J., Vellei, M., Simone, M. D., Shamsaiee, M., Dabirian, S., Yan, D., and Kang, X. A guideline to document occupant behavior models for advanced building controls ´ n Drgo n.

[35] Drgoňa, J., Arroyo, J., Cupeiro Figueroa, I., Blum, D., Arendt, K., Kim, D., Ollé, E. P., Oravec, J., Wetter, M., Vrabie, D. L., and Helsen, L. All you need to know about model predictive control for buildings. *Annual Reviews in Control 50* (2020), 190–232. `https://doi.org/10.1016/j.arcontrol.2020.09.001`.

[36] Drgoňa, J., Picard, D., Kvasnica, M., and Helsen, L. Approximate model predictive building control via machine learning. *Applied Energy 218* (2018), 199–216.

[37] Drgoňa, J., Tuor, A., Skomski, E., Vasisht, S., and Vrabie, D. Deep learning explicit differentiable predictive control laws for buildings. *IFAC-PapersOnLine 54*, 6 (2021), 14–19. 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021.

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

[39] Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901* (2019).

[40] Esrafilian-Najafabadi, M., and Haghighat, F. Towards self-learning control of HVAC systems with the consideration of dynamic occupancy patterns: Application of model-free deep reinforcement learning. 109747.

[41] Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence* (2018), vol. 32.

[42] Földváry Ličina, V., Cheung, T., Zhang, H., de Dear, R., Parkinson, T., Arens, E., Chun, C., Schiavon, S., Luo, M., Brager, G., Li, P., and Kaam, S. ASHRAE Global Thermal Comfort Database II. *Dataset v4* (2018), 1–4.

[43] Folkers, A., Rick, M., and Büskens, C. Controlling an autonomous vehicle with deep reinforcement learning. In *2019 IEEE Intelligent Vehicles Symposium (IV)* (2019), IEEE, pp. 2025–2031.

[44] Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (2018), PMLR, pp. 1587–1596.

[45] Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning* (2019), PMLR, pp. 2052–2062.

[46] Galataud, A. Ray RLlib - EnergyPlus Python API sample, Mar. 2023. original-date: 2022-08-08T15:50:03Z.

[47] Gao, G., Li, J., and Wen, Y. Energy-efficient thermal comfort control in smart buildings via deep reinforcement learning. *arXiv preprint arXiv:1901.04693* (2019).

[48] Guadarrama, S., Korattikara, A., Ramirez, O., Castro, P., Holly, E., Fishman, S., Wang, K., Gonina, E., Wu, N., Kokiopoulou, E., Sbaiz, L., Smith, J., Bartók, G., Berent, J., Harris, C., Vanhoucke, V., and Brevdo, E. TF-Agents: A library for reinforcement learning in tensorflow. `https://github.com/tensorflow/agents`, 2018. [Online; accessed 25-June-2019].

[49] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (2018).

[50] Hahn, J., Heiler, S., Kane, M. B., Park, S., and Jensch, W. The Information Gap in Occupant-Centric Building Operations: Lessons Learned from Interviews with Building Operators in Germany. *Frontiers in Built Environment 8* (May 2022), 838859.

[51] Heidari, A., Maréchal, F., and Khovalyg, D. Reinforcement Learning for proactive operation of residential energy systems by learning stochastic occupant behavior and fluctuating solar energy: Balancing comfort, hygiene and energy use. 119206.

[52] Henze, G. P., and Schoenmann, J. Evaluation of reinforcement learning control for thermal energy storage systems. *HVAC and R Research 9* (2003), 259–275.

[53] Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence* (2018).

[54] Jebessa, E., Olana, K., Getachew, K., Isteefanos, S., and Mohd, T. K. Analysis of reinforcement learning in autonomous vehicles. In *2022 IEEE 12th Annual Computing and Communication Workshop and*

Conference (CCWC) (2022), IEEE, pp. 0087–0091.

[55] Jiménez-Raboso, J., Campoy-Nieves, A., Manjavacas-Lucas, A., Gómez-Romero, J., and Molina-Solana, M. Sinergym: A building simulation and control framework for training reinforcement learning agents. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, USA, 2021), Association for Computing Machinery, p. 319–323.

[56] Kakade, S. M. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, Cambridge, MA, 2002, pp. 1531–1538.

[57] Kamthe, S., and Deisenroth, M. P. Data-efficient reinforcement learning with probabilistic model predictive control. *arXiv preprint arXiv:1706.06491* (2017).

[58] Kim, J., Schiavon, S., and Brager, G. Personal comfort models–a new paradigm in thermal comfort for occupant-centric environmental control. *Building and Environment 132* (2018), 114–124.

[59] Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., Behar, J. V., Hern, S. C., and Engelmann, W. H. The national human activity pattern survey (nhaps): A resource for assessing exposure to environmental pollutants. *Journal of Exposure Analysis and Environmental Epidemiology 11* (2001), 231–252.

[60] Kuhnle, A., Schaarschmidt, M., and Fricke, K. Tensorforce: a tensorflow library for applied reinforcement learning. Web page, 2017.

[61] Kurte, K., Munk, J., Kotevska, O., Amasyali, K., Smith, R., McKee, E., Du, Y., Cui, B., Kuruganti, T., and Zandi, H. Evaluating the adaptability of reinforcement learning based hvac control for residential houses. *Sustainability 12*, 18 (2020).

[62] Lei, B., Janssen, P., Stoter, J., and Biljecki, F. Challenges of urban digital twins: A systematic review and a Delphi expert survey. *Automation in Construction 147* (Mar. 2023), 104716.

[63] Lei, Y., Zhan, S., Ono, E., Peng, Y., Zhang, Z., Hasama, T., and Chong, A. A practical deep reinforcement learning framework for multivariate occupant-centric control in buildings. *Applied Energy 324* (Oct. 2022), 119742.

[64] Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research 17*, 1 (2016), 1334–1373.

[65] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research 37*, 4-5 (2018), 421–436.

[66] Li, B., and Xia, L. A multi-grid reinforcement learning method for energy conservation and comfort of hvac in buildings. *IEEE International Conference on Automation Science and Engineering 2015-Octob* (2015), 444–449.

[67] Li, H., Wan, Z., and He, H. Real-time residential demand response. *IEEE Transactions on Smart Grid 11*, 5 (2020), 4144–4154.

[68] Li, H., Wang, Z., and Hong, T. A synthetic building operation dataset. *Scientific Data 8*, 1 (Aug 2021), 213.

[69] Li, Y., Wen, Y., Guan, K., and Tao, D. Transforming cooling optimization for green data center via deep reinforcement learning. *arXiv preprint arXiv:1709.05077* (2017).

[70] Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J. E., Jordan, M. I., and Stoica, I. RLlib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)* (2018).

[71] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[72] Lissa, P., Schukat, M., and Barrett, E. Transfer learning applied to reinforcement learning-based hvac control. *SN Comput. Sci. 1* (2020), 127.

[73] Liu, H.-Y., Fu, X., Balaji, B., Gupta, R., and Hong, D. B2rl: An open-source dataset for building batch reinforcement learning. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, USA, 2022), BuildSys '22, Association for Computing Machinery, p. 462–465.

[74] Liu, S., and Henze, G. Investigation of reinforcement learning for building thermal mass control. *Proceedings of SimBuild Conference 2004: 1st conference of IBPSA-USA 1* (2004).

[75] Liu, S., and Henze, G. P. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 1. theoretical foundation. *Energy and Buildings 38* (2006), 142–147.

[76] Liu, S., and Henze, G. P. Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2. results and analysis. *Energy and Buildings 38* (2 2006), 148–161.

[77] Liu, S., and Henze, G. P. Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory. *Journal of solar energy engineering 129*, 2 (2007), 215–225.

[78] Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems 30* (2017).

[79] Lu, X., Fu, Y., and O'Neill, Z. Benchmarking high performance hvac rule-based controls with advanced intelligent controllers: A case study in a multi-zone system in modelica. *Energy and Buildings 284* (4 2023).

[80] Luo, J., Paduraru, C., Voicu, O., Chervonyi, Y., Munns, S., Li, J., Qian, C., Dutta, P., Davis, J. Q., Wu, N., Yang, X., Chang, C.-M., Li, T., Rose, R., Fan, M., Nakhost, H., Liu, T., Kirkman, B., Altamura, F., Cline, L., Tonker, P., Gouker, J., Uden, D., Bryan, W. B., Law, J., Fatiha, D., Satra, N., Rothenberg, J., Waraich, M., Carlin, M., Tallapaka, S., Witherspoon, S., Parish, D., Dolan, P., Zhao, C., and Mankowitz, D. J. Controlling commercial cooling systems using reinforcement learning, 2022.

[81] Luo, N., Wang, Z., Blum, D., Weyandt, C., Bourassa, N., Piette, M. A., and Hong, T. A

three-year dataset supporting research on building energy management and occupancy analytics. *Scientific Data 9*, 1 (Apr 2022), 156.

[82] Malik, J., Mahdavi, A., Azar, E., Putra, H. C., Berger, C., Andrews, C., and Hong, T. Ten questions concerning agent-based modeling of occupant behavior for energy and environmental performance of buildings. *Building and Environment 217* (2022), 109016.

[83] Mania, H., Guy, A., and Recht, B. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055* (2018).

[84] Martins, L. A., Soebarto, V., and Williamson, T. A systematic review of personal thermal comfort models. *Building and Environment 207* (2022), 108502.

[85] Marzullo, T., Dey, S., Long, N., Leiva Vilaplana, J., and Henze, G. A high-fidelity building performance simulation test bed for the development and evaluation of advanced controls. *Journal of Building Performance Simulation 15*, 3 (2022), 379–397.

[86] Mason, K., and Grijalva, S. Building hvac control via neural networks and natural evolution strategies. In *2021 IEEE Congress on Evolutionary Computation (CEC)* (2021), IEEE, pp. 2483–2490.

[87] Miller, C., Abdelrahman, M., Chong, A., Biljecki, F., Quintana, M., Frei, M., Chew, M., and Daniel, W. The Internet-of-Buildings (IoB) – Digital twin convergence of wearable and IoT data with GIS / BIM. *CISBAT 2021 - Carbon Neutral Cities - Energy Efficiency & Renewables in the Digital Era, EPFL*, July (2021).

[88] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (New York, NY, 2016), pp. 1928–1937.

[89] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[90] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature 518*, 7540 (2015), 529–533.

[91] Moriyama, T., De Magistris, G., Tatsubori, M., Pham, T.-H., Munawar, A., and Tachibana, R. Reinforcement learning testbed for power-consumption optimization. In *Methods and Applications for Modeling and Simulation of Complex Systems* (Singapore, 2018), Springer Singapore, pp. 45–59.

[92] Mozer, M. C. The neural network house: An environment hat adapts to its inhabitants. In *Proc. AAAI Spring Symp. Intelligent Environments* (1998), vol. 58.

[93] Nagy, G. Z. The CityLearn Challenge 2021, 2021.

[94] Nagy, Z., Park, J. Y., and Vázquez-Canteli, J. R. Reinforcement learning for intelligent environments: A tutorial. *Routledge Handbook of Sustainable and Resilient Infrastructure* (2018), 733–746.

[95] Naug, A., Q'uiñones Grueiro, M., and Biswas, G. A Relearning Approach to Reinforcement Learning for control of Smart Buildings. *Annual Conference of the PHM Society 12*, 1 (Nov. 2020), 14–14. Number: 1.

[96] Nouidui, T., Wetter, M., and Zuo, W. Functional mock-up unit for co-simulation import in energyplus. *Journal of Building Performance Simulation 7*, 3 (2014), 192–202.

[97] Nweye, K., Liu, B., Stone, P., and Nagy, Z. Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings. *Energy and AI 10* (2022), 100202.

[98] Nweye, K., and Nagy, G. Z. The CityLearn Challenge 2021 Benchmark Results, 2023.

[99] Nweye, K., Nagy, Z., Liu, B., and Stone, P. Offline training of multi-agent reinforcement agents for grid-interactive buildings control. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems* (New York, NY, USA, 2022), e-Energy '22, Association for Computing Machinery, p. 442–443.

[100] Nweye, K., Sankaranarayanan, S., and Nagy, Z. Merlin: Multi-agent offline and transfer learning for occupant-centric energy flexible operation of grid-interactive communities using smart meter data and citylearn.

[101] Nweye, K., Siva, S., and Nagy, G. Z. The CityLearn Challenge 2022, 2023.

[102] O'Brien, W., Gaetani, I., Carlucci, S., Hoes, P. J., and Hensen, J. L. On occupant-centric building performance metrics. *Building and Environment 122* (2017), 373–385.

[103] O'Brien, W., Wagner, A., Schweiker, M., Mahdavi, A., Day, J., Kjærgaard, M. B., Carlucci, S., Dong, B., Tahmasebi, F., Yan, D., et al. Introducing iea ebc annex 79: Key challenges and opportunities in the field of occupant-centric building design and operation. *Building and Environment 178* (2020), 106738.

[104] of Energy (DOE), D. Chapter 5: Increasing efficiency of building systems and technologies. *Quadrennial Technology Review, An Assessment of Energy Technologies and Research Opportunities* (2015), 143–181.

[105] OpenAI. ChatGPT: Optimizing Language Models for Dialogue.

[106] Park, J. Y., Dougherty, T., Fritz, H., and Nagy, Z. Lightlearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment 147* (2019), 397–414.

[107] Park, J. Y., and Nagy, Z. Comprehensive analysis of the relationship between thermal comfort and building control research-a data-driven literature review. *Renewable and Sustainable Energy Reviews 82* (2018), 2664–2679.

[108] Park, J. Y., and Nagy, Z. Hvaclearn: A reinforcement learning based occupant-centric control for thermostat set-points, 6 2020.

[109] Park, J. Y., and Nagy, Z. Hvaclearn: A reinforcement learning based occupant-centric control for thermostat set-points. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems* (2020), pp. 434–437.

[110] Park, J. Y., Ouf, M. M., Gunay, B., Peng, Y., O'Brien, W., Kjærgaard, M. B., and Nagy, Z. A critical review of field implementations of occupant-

centric building controls. *Building and Environment 165* (2019), 106351.

[111] Peirelinck, T., Hermans, C., Spiessens, F., and Deconinck, G. Domain randomization for demand response of an electric water heater. *IEEE Transactions on Smart Grid 12*, 2 (2021), 1370–1379.

[112] Peng, K. S., and Morrison, C. T. Model predictive prior reinforcement learning for a heat pump thermostat. *IEEE International Conference on Automatic Computing: Feedback Computing* (2016), 189–190.

[113] Pipattanasomporn, M., Chitalia, G., Songsiri, J., Aswakul, C., Pora, W., Suwankawin, S., Audomvongseree, K., and HoonchareonWang, N. Cu-bems, smart building electricity consumption and indoor environmental sensor datasets. *Scientific Data 7*, 241 (July 2020).

[114] Plappert, M. keras-rl. `https://github.com/keras-rl/keras-rl`, 2016.

[115] Pritoni, M., Paine, D., Fierro, G., Mosiman, C., Poplawski, M., Saha, A., Bender, J., and Granderson, J. Metadata schemas and ontologies for building energy applications: A critical review and use case analysis. *Energies 14*, 7 (2021).

[116] Quintana, M., Nagy, Z., Tartarini, F., Schiavon, S., and Miller, C. Comfortlearn: enabling agent-based occupant-centric building controls. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2022), pp. 475–478.

[117] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research 22*, 268 (2021), 1–8.

[118] Richalet, J., Rault, A., Testud, J. L., and Papon, J. Model predictive heuristic control. applications to industrial processes. *Automatica 14* (1978), 413–428.

[119] Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (2011), JMLR Workshop and Conference Proceedings, pp. 627–635.

[120] Roth, A., and Reyna, J. Grid-interactive efficient buildings technical report series: Whole-building controls, sensors, modeling, and analytics, 2019.

[121] Rummery, G. A., and Niranjan, M. *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[122] Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864* (2017).

[123] Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532* (2017).

[124] Scharnhorst, P., Schubnel, B., Fernández Bandera, C., Salom, J., Taddeo, P., Boegli, M., Gorecki, T., Stauffer, Y., Peppas, A., and Politi, C. Energym: A building model library for controller benchmarking. *Applied Sciences 11*, 8 (2021), 3518.

[125] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature 588*, 7839 (2020), 604–609.

[126] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning* (Lille, France, 2015), pp. 1889–1897.

[127] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015).

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

[129] Shin, E.-J., Yus, R., Mehrotra, S., and Venkatasubramanian, N. Exploring fairness in participatory thermal comfort control in smart buildings. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments* (2017), pp. 1–10.

[130] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Driessche, G. V. D., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. Mastering the game of go with deep neural networks and tree search. *Nature 529* (2016), 484–489.

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

[132] Sutton, Richard S. Barto, A. G. *Reinforcement Learning : An Introduction*. The MIT Press, 2014.

[133] Szita, I., and Lörincz, A. Learning tetris using the noisy cross-entropy method. *Neural computation 18*, 12 (2006), 2936–2941.

[134] Touzani, S., Granderson, J., Pritoni, M., Kiran, M., Krishnan Prakash, A., Wang, Z., Agarwal, S., and USDOE. Flexdrl v1.0, 2 2021.

[135] Touzani, S., Prakash, A. K., Wang, Z., Agarwal, S., Pritoni, M., Kiran, M., Brown, R., and Granderson, J. Controlling distributed energy resources via deep reinforcement learning for load flexibility and energy efficiency. *Applied Energy 304* (2021), 117733.

[136] Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (2016), vol. 30.

[137] Vazquez Canteli, J., and Nagy, Z. The CityLearn Challenge 2020, 2020.

[138] Vázquez-Canteli, J. R., Kämpf, J., Henze, G., and Nagy, Z. Citylearn v1. 0: An openai gym environment for demand response with deep reinforcement learning. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2019), pp. 356–357.

[139] Vázquez-Canteli, J. R., and Nagy, Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied energy 235* (2019), 1072–1089.

[140] Voloshin, C., Le, H. M., Jiang, N., and Yue, Y. Empirical study of off-policy policy evaluation for reinforcement learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and*

Benchmarks Track (Round 1) (2021).

[141] Von Frankenberg, N., Loftness, V., and Bruegge, B. I want it that way: Thermal desirability in shared spaces. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2021), pp. 204–207.

[142] Vázquez-Canteli, J. R., Henze, G., and Nagy, Z. Marlisa: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2020), 170–179.

[143] Wang, W. Z. AlphaBuilding-MedOffice, Mar. 2023. original-date: 2021-04-19T17:15:55Z.

[144] Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224* (2016).

[145] Wang, Z., Chen, B., Li, H., and Hong, T. Alphabuilding rescommunity: A multi-agent virtual testbed for community-level load coordination. *Advances in Applied Energy 4* (2021), 100061.

[146] Wang, Z., and Hong, T. Reinforcement learning for building controls: The opportunities and challenges. *Applied Energy 269*, February (2020), 115036.

[147] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning* (2016), PMLR, pp. 1995–2003.

[148] Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2746–2754.

[149] Wei, T., Wang, Y., and Zhu, Q. Deep reinforcement learning for building hvac control, 6 2017.

[150] Wei, T., Wang, Y., and Zhu, Q. Deep reinforcement learning for building HVAC control. In *Proceedings of the 54th Annual Design Automation Conference 2017* (New York, NY, 2017), ACM, p. 22.

[151] Wetter, M., Haves, P., and Coffey, B. Building controls virtual test bed. Tech. rep., Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2008.

[152] Wurman, P. R., Barrett, S., Kawamoto, K., MacGlashan, J., Subramanian, K., Walsh, T. J., Capobianco, R., Devlic, A., Eckert, F., Fuchs, F., Gilpin, L., Khandelwal, P., Kompella, V., Lin, H., MacAlpine, P., Oller, D., Seno, T., Sherstan, C., Thomure, M. D., Aghabozorgi, H., Barrett, L., Douglas, R., Whitehead, D., Dürr, P., Stone, P., Spranger, M., and Kitano, H. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature 602*, 7896 (2022), 223–228.

[153] Xilei, D., Cheng, S., and Chong, A. Deciphering optimal mixed-mode ventilation in the tropics using reinforcement learning with explainable artificial intelligence. *Energy and Buildings 278* (11 2022), 112629.

[154] Xu, S., Wang, Y., Wang, Y., O'Neill, Z., and Zhu, Q. One for many: Transfer learning for building hvac control. *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (2020).

[155] Yan, D., Hong, T., Dong, B., Mahdavi, A., D'Oca, S., Gaetani, I., and Feng, X. Iea ebc annex 66: Definition and simulation of occupant behavior in buildings. *Energy and Buildings 156* (2017), 258–270.

[156] Yang, L., Nagy, Z., Goffin, P., and Schlueter, A. Reinforcement learning for optimal control of low exergy buildings. *Applied Energy 156* (2015), 577–586.

[157] Yang, L., Nagy, Z., Goffin, P., and Schlueter, A. Reinforcement learning for optimal control of low exergy buildings. *Applied Energy 156* (2015), 577–586.

[158] Zhang, B., Hu, W., Ghias, A. M., Xu, X., and Chen, Z. Multi-agent deep reinforcement learning-based coordination control for grid-aware multi-buildings. *Applied Energy 328* (2022), 120215.

[159] Zhang, L. Data-driven building energy modeling with feature selection and active learning for data predictive control. *Energy and Buildings 252* (2021), 111436.

[160] Zhang, T., and Ardakanian, O. Cobs: Comprehensive building simulator. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, USA, 2020), BuildSys '20, Association for Computing Machinery, pp. 314–315.

[161] Zhang, X., Biagioni, D., Cai, M., Graf, P., and Rahman, S. An edge-cloud integrated solution for buildings demand response using reinforcement learning. *IEEE Transactions on Smart Grid 12*, 1 (2020), 420–431.

[162] Zhang, X., Chen, Y., Bernstein, A., Chintala, R., Graf, P., Jin, X., and Biagioni, D. Two-stage reinforcement learning policy search for grid-interactive building control. *IEEE Transactions on Smart Grid 13*, 3 (2022), 1976–1987.

[163] Zhang, Z., Chong, A., Pan, Y., Zhang, C., and Lam, K. P. Whole building energy model for hvac optimal control: A practical framework based on deep reinforcement learning. *Energy and Buildings 199* (2019), 472–490.

[164] Zhang, Z., and Lam, K. P. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments* (New York, NY, USA, 2018), BuildSys '18, ACM, pp. 148–157.

[165] Zhang, Z., and Lam, K. P. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments* (2018), pp. 148–157.

[166] Zhao, W., Queralta, J. P., and Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* (2020), pp. 737–744.

[167] Zhu, Z., Lin, K., and Zhou, J. Transfer learning in deep reinforcement learning: A survey. *CoRR abs/2009.07888* (2020).