

ORIGINAL RESEARCH

Open access

Practical Obstacles in Applying Multi-Agent Reinforcement Learning to Grid-Responsive Buildings

Jose Martinez^{1*}, Carmen Lopez¹

Abstract

Previous research has highlighted the need for standardized experimental settings in building control studies, and recent work on real-world reinforcement learning (RL) has introduced challenges that remain largely unexplored. Building on these insights, we propose a curated set of nine real-world challenges specifically aimed at RL control in grid-interactive buildings (GIBs). We suggest that studies in this field adopt this challenge-based perspective alongside standardized environments to ensure repeatable results. While advanced control strategies like model predictive control (MPC) and RL each offer unique advantages, both face limitations that hinder real-world deployment, and direct, unbiased comparisons between them are scarce. By framing evaluations around concrete challenges, it becomes possible to systematically test controller performance under diverse conditions and facilitate fairer comparisons. To demonstrate this approach, we focus on the offline learning challenge using CityLearn, an OpenAI Gym environment tailored for RL in demand response scenarios, in which energy storage across a collection of heterogeneous buildings is manipulated to shape the aggregated electricity demand profile. Using this setup, we examine how different levels of domain knowledge and RL algorithm complexity influence outcomes, revealing that the sequence of operations (SOOs) generated by a rule-based controller (RBC)—which provides training logs for offline RL agents—directly affects performance on four energy flexibility metrics. Prolonged offline training using an optimized RBC yields sustained performance gains, whereas training on logs from a simplified RBC can deteriorate agent effectiveness over time. Additionally, sharing information between agents does not appear to affect results significantly. Overall, we call for stronger interdisciplinary collaboration to tackle these real-world challenges and unlock the full potential of GIB control systems.

Keywords Reinforcement learning, Grid-interactive buildings, Benchmarking, Rule-based controller (RBC)

*Correspondence:

Jose Martinez

jose.martinez@gmail.com

¹ Department of Energy Systems Engineering, Faculty of Engineering, University of Salamanca, Salamanca, Spain

Introduction

Buildings are responsible for about 40% of global energy consumption and roughly 30% of associated greenhouse gas emissions, yet retrofitting and optimization can reduce CO₂ emissions by 50% to 90% [1]. Achieving deep decarbonization requires electrifying end-use systems while simultaneously greening electricity supply, optimizing energy use in lighting, heating, ventilation, air conditioning (HVAC), and domestic hot water (DHW) systems, and enhancing building thermal performance [2]. A major factor in grid decarbonization is the integration of renewable energy sources (RESs) into electricity networks, complemented by the adoption of photovoltaic (PV) panels and solar thermal collectors within buildings. Electric vehicles (EVs), due to their storage capacity and network connectivity, further expand

opportunities for synergistic integration with buildings [3]. However, careful coordination is necessary to maintain grid stability and reliability during operation (Figure 1) [4-6].

Demand response (DR) mechanisms enable consumers and prosumers to actively provide grid flexibility by curbing consumption, shifting demand, or generating and storing electricity at strategic times, often in exchange for reduced energy costs [7]. HVAC systems contribute to DR by adjusting temperature setpoints to reduce load, performing pre-heating or pre-cooling to shift load (passive storage), or storing thermal energy directly in dedicated storage units (active storage) [8]. Residential DR-enabled thermostats allow electricity providers to modify setpoints during peak demand periods, offering savings to end users. The proliferation of communication technologies enables diverse components—PV arrays, HVAC units, storage systems, EVs, and smart thermostats—to share operational data, creating a foundation for smart city ecosystems that promote energy efficiency and sustainable urban development [9].



Figure 1. Grid-interactive buildings.

Sophisticated control strategies can significantly enhance demand response (DR) by automatically managing building energy systems while accounting for the unique features of occupants and structures. For demand response to deliver its full potential, however, electrical loads need to be managed in a responsive, flexible, and smart manner. If every electrical load responds simultaneously to identical signals, the result may be a shift in aggregated peak demand rather than an actual reduction. Consequently, more advanced and coordinated approaches are required to synchronize the operation of the various technologies involved.

Advanced control techniques such as Model Predictive Control (MPC) [10] and deep Reinforcement Learning (RL) [11] have been explored for numerous building control tasks. Although each method has drawbacks—for instance, MPC depends on an accurate model while RL demands substantial data—impressive applications and outcomes have been reported in recent years. Moreover, hybrid approaches combining physics-informed neural networks for modeling have started to appear [12].

Unlike MPC, RL offers an adaptive, and often model-free, control strategy that can leverage both real-time and historical data to enable effective demand response. RL is an agent-based machine learning paradigm in which agents discover

optimal behaviors through ongoing interaction with their environment [13, 14]. In contrast to supervised learning, the agent does not rely on large volumes of labeled examples. Unlike unsupervised learning, it receives delayed feedback from the environment. In simple terms, given a current input (state), the agent selects an action, observes the resulting reward signal (immediate or delayed), and updates its understanding of which actions perform best under specific conditions.

Reinforcement learning can be divided into single-agent reinforcement learning (SARL) and multi-agent reinforcement learning (MARL). SARL is typically formulated as a Markov Decision Process (MDP), whereas MARL is usually modeled as a Markov Game (MG) involving multiple interacting agents. SARL satisfies the stationarity assumption of an MDP, but in MARL, interactions among agents can lead to partial observability, potentially breaking the stationarity condition [15]. Nevertheless, MARL is particularly well suited to environments characterized by high-dimensional state and action spaces that involve cooperation or competition among agents—features commonly found in communities of grid-interactive buildings (GIBs). Through MARL, system-wide goals such as peak shaving and ramp-rate reduction can be directly optimized.

A key difficulty when applying RL to demand response is the fair and consistent comparison of different algorithms [4]. As noted in Wölflé et al. [16], it is essential to establish a shared set of representative environments to enable systematic evaluation and comparison of building optimization methods.

Building upon Wölflé et al. [16] and drawing inspiration from Dulac-Arnold et al. [17, 18], this paper has two main objectives. First, we identify and discuss several practical, real-world challenges relevant to grid-interactive buildings that the research community should prioritize. Second, we demonstrate Challenge 8 (C8)—specifically, training offline using fixed logs generated by an external behavior policy—within the CityLearn simulation environment [19].

Real-World Challenges

Dulac-Arnold et al. outlined nine practical challenges for reinforcement learning [17]. They proposed a collection of benchmark environments in Dulac-Arnold et al. [18] to test algorithms addressing those challenges. However, the environments described in Dulac-Arnold et al. [18] are not appropriate for assessing grid-interactive buildings, since they focus on small-scale problems that lack the necessary domain-specific knowledge and context. In the following, we adapt and present these nine challenges within the specific setting of GIBs, retaining the original italicized descriptions from Dulac-Arnold et al. [18] for reference.

C1: Being able to learn on live systems from limited samples. In this challenge, the controller begins with random initialization and must learn solely from the data samples it collects. The available samples can be artificially restricted—for example, by providing data only every three hours instead of every 15 minutes. Algorithms can be assessed based on the speed of convergence (in terms of time or number of samples) and the stability of their exploration behavior. Alternatively, the trade-off between data requirements and final controller performance can be examined.

C2: Dealing with unknown and potentially large delays in the system actuators, sensors, or feedback. The thermal dynamics of buildings introduce significant time delays between control actions (such as adjusting HVAC systems) and their observable effects. This is particularly relevant for strategies like pre-cooling or pre-heating that exploit the building's thermal mass. The controller must learn these dynamics implicitly and automatically. Benchmark datasets featuring buildings with varying levels of thermal mass (from light to heavy) should be developed, allowing comparison of converged controllers to reveal how longer feedback delays affect performance.

C3: Learning and acting in high-dimensional state and action spaces. This challenge evaluates the scalability of control solutions. Because buildings naturally have large state–action spaces, controllers can be tested on selected subsets to determine how performance varies across them. When controlling multiple buildings or multiple zones within a single building, scalability is assessed by increasing the number of controlled entities and observing the resulting control quality.

C4: Reasoning about system constraints that should never or rarely be violated. This represents a core issue in building control, which typically involves balancing energy savings with maintaining comfortable indoor conditions. Additional operational constraints include preserving the minimum battery state of charge (SOC) and keeping temperatures within safe limits. Algorithms should be evaluated both by the number of constraint violations occurring during training and by the behavior of the final policy. The integration of constraint handling directly into the objective function is covered under C6.

C5: Interacting with partially observable systems, which can alternatively be viewed as systems that are non-stationary or stochastic. This challenge consists of two aspects. First, observations can be deliberately corrupted with realistic failures such as sensor noise or missing values, which are common in actual building and HVAC systems. Controller performance can then be analyzed across increasing levels of such disturbances. Second, the controller's robustness can be tested on perturbed versions of the system—for instance, after building retrofits (improved envelope or windows), equipment upgrades, changes in occupant behavior, or shifts in climate conditions. The ability to maintain performance on these altered systems serves as a key evaluation metric.

C6: Learning from multiple or poorly specified objective functions. Building energy management is inherently multi-objective, especially when dealing with multiple zones or multiple buildings. There may also be a combination of global objectives (e.g., total building energy consumption) and local objectives (e.g., individual equipment performance). As noted in C4, constraints can be embedded directly into the objective function. For fair evaluation, the contributions of individual objectives should be analyzed separately.

C7: Being able to provide actions quickly, especially for systems requiring low latencies. Latency is the delay between acquiring a measurement and executing the corresponding control action, often due to lengthy computation. In real-world smart buildings and microgrids, additional latency may arise when computation occurs in the cloud, compounded by data transmission delays or network issues. To study this, artificial timestep delays of varying durations can be introduced during control execution, and their effect on overall performance can be quantified.

C8: Training offline from the fixed logs of an external behavior policy. The objective here is to derive an improved control policy from historical data logs produced by a suboptimal reference controller, such as a rule-based controller (RBC), which is frequently available in practice. In addition to the simulation environment, datasets of varying lengths (e.g., 2 weeks, 1 month, or 6 months) generated by a known reference RBC should be provided. Controllers are then judged on their capacity to outperform these baseline policies.

C9: Providing system operators with explainable policies. In contrast to the original proposal in Dulac-Arnold *et al.* [18], which emphasized the generation of figures for better interpretability, the building domain requires control decisions that can be communicated in straightforward terms to facility managers. Progress in explainable artificial intelligence (XAI) is therefore essential. Algorithms that may be slightly suboptimal but are more transparent and easier to understand are often preferred, as they are more likely to gain acceptance and be deployed in practice. Developing shared standards and expected outcomes between researchers and operators would further improve communication among stakeholders.

Each of these challenges demands a tailored experimental setup within a suitable simulation platform to investigate and measure the relevant influencing factors properly. As an illustration, we specifically design an experiment for and demonstrate Challenge 8 (C8) using the CityLearn environment [19].

Background

We briefly review the fundamentals of reinforcement learning and multi-agent reinforcement learning. Comprehensive introductions are available in Sutton and Barto [13].

Reinforcement learning

In reinforcement learning, an agent interacts continuously with an environment to maximize the cumulative reward it receives. The problem is commonly modeled as a Markov Decision Process (MDP). An MDP is formally defined by the tuple $M = (S, A, T, \gamma, R)$, where S and A represent the state and action spaces, respectively. At each time step t , the agent observes a state $s_t \in S$, selects an action $a_t \in A$, and transitions to a new state $s_{t+1} \sim T(\cdot | s_t, a_t)$, where T is the (usually unknown) transition function. The agent then receives a scalar reward $r_t \sim R(s_t, a_t)$. The central aim of RL is to discover a policy $\pi: S \rightarrow A$ that maximizes the expected cumulative discounted return:

$$\max_{\pi} \mathbb{E}_{s_t, a_t \sim \pi(\cdot | s_t)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \quad (1)$$

It can be shown that, for any stationary policy π , this objective leads to a well-defined value function that depends on the starting state. The state-value function is expressed as:

$$V^{\pi}(s) = \mathbb{E}_{s_0=s, \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \quad (2)$$

Where $r(s_t, a_t) = r_t \sim R(s_t, a_t)$ denotes expectation over trajectories generated by policy π . Similarly, the action-value function is defined as:

$$Q^{\pi}(s, a) = \mathbb{E}_{s_0=s, a_0=a, \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \quad (3)$$

The reinforcement learning objective expressed in [Eq. 1](#) can be reformulated as:

$$\max_{\pi} V^{\pi}(s), \forall s. \quad (4)$$

To optimize the objective described above, reinforcement learning algorithms generally fall into two categories: value-based and policy-based methods. Value-based approaches rely on the classical Bellman equation for the action-value function. Let the optimal action-value function be denoted as Q^* ; it is established that Q^* satisfies the following relationship:

$$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot | s, a)} \max_{a'} Q^*(s', a'). \quad (5)$$

By minimizing the difference between the left and right-hand sides of the above equation, we reach the Q-learning algorithm [\[20\]](#).

Multi-agent reinforcement learning

Multi-agent reinforcement learning (MARL) generalizes the standard RL paradigm to environments with multiple interacting agents. The typical MARL framework can be classified into cooperative, competitive, or hybrid settings. In this study, we concentrate on the cooperative scenario, as the primary goal is to align the actions of multiple buildings to smooth the overall electricity demand profile, which constitutes a shared objective among all agents. Formally, the MARL problem considered here is expressed as a Markov decision process (MDP) described by the tuple $M = (S, A, T, \gamma, R)$, with key distinctions from single-agent RL: (1) the action space A now represents the combination of all agents' actions,

i.e., $A = U_1 \times U_2 \times \dots \times U_n$, where U_i denotes the action set available to agent i ; (2) the state space S corresponds to the joint observations of all agents, $S = O_1 \times O_2 \times \dots \times O_n$, with O_i representing the observation for agent i . The overall workflow for RL and MARL is illustrated in **Figure 2**; further details on the algorithms can be found in Li [21].

Although a multi-agent system could, in theory, be treated as a single-agent problem with a central controller determining the actions of all agents, this approach is often impractical due to the rapid expansion of state and action spaces as the number of agents increases [22], as well as the high computational cost of training such a centralized agent. Moreover, centralized control structures tend to reduce system resilience against adversarial attacks [23]. Consequently, decentralized methods, in which each agent learns its own policy independently, offer a more feasible solution. However, a fully decentralized setup, in which agents lack information about others' strategies, can lead to suboptimal coordination.

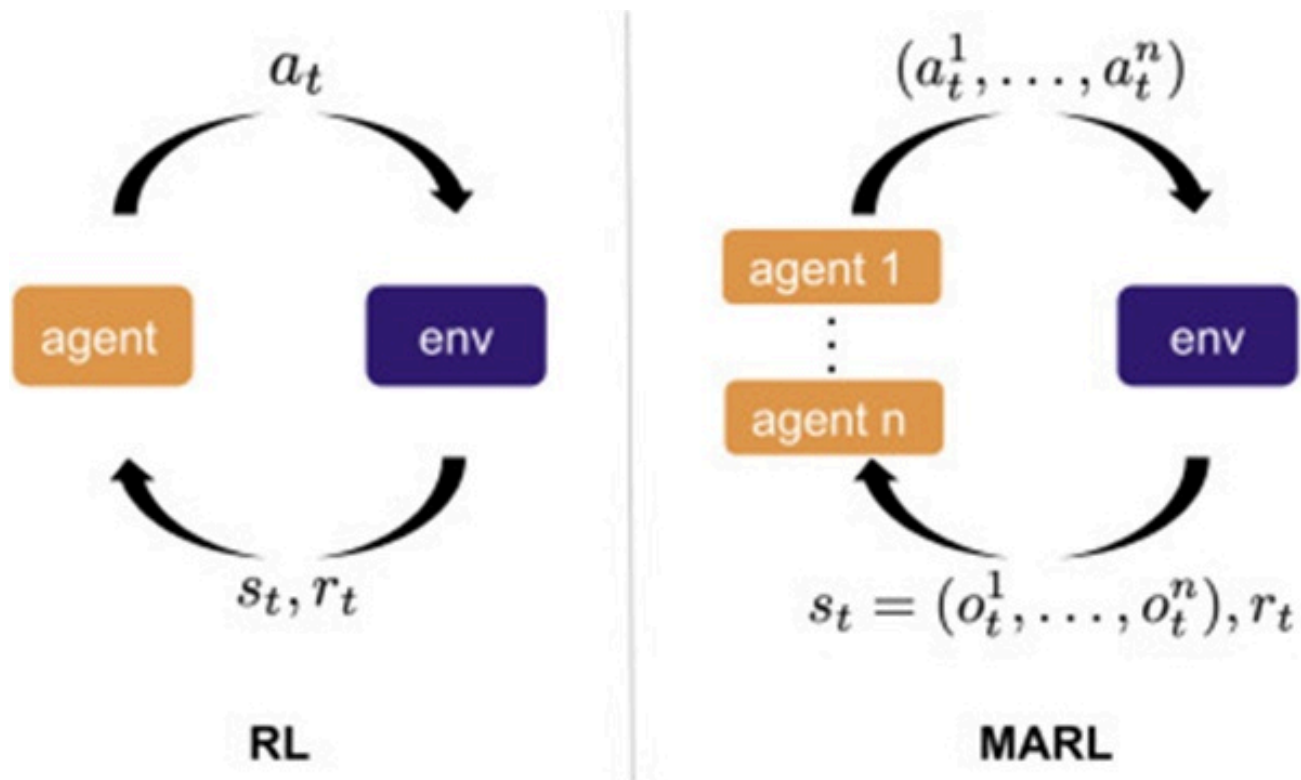


Figure 2. The pipeline of RL and MARL.

CityLearn

CityLearn is an OpenAI Gym-based simulation environment designed to simplify the development and testing of reinforcement learning (RL) agents for demand response (DR) applications. It enables agents to reshape the overall district-level electricity demand profile by intelligently managing energy storage systems across a variety of buildings [19, 24, 25]. The primary goal of CityLearn is to provide a standardized platform for consistently evaluating and benchmarking RL algorithms.

The environment incorporates detailed energy models for several key devices, including air-to-water heat pumps, electric heaters, chilled water (CHW) storage, domestic hot water (DHW) storage, and electrical energy storage systems, as illustrated in **Figure 3**. In each building, an air-to-water heat pump handles cooling. In contrast, an electric heater fulfills the DHW heating needs. Buildings may also be equipped with various combinations of CHW, DHW, and electricity storage units to help reduce reliance on grid electricity for cooling, DHW heating, and general electrical loads. The capacities of the CHW and DHW storage systems are specified as multiples of the number of hours these devices can fully cover the building's peak annual cooling and DHW demands, respectively, when operating at full charge.

All devices, along with other non-controllable electric equipment and household appliances (non-shiftable loads), draw electricity from the main power grid. Additionally, photovoltaic (PV) systems can be integrated into the buildings to generate on-site electricity and thereby offset a portion of the grid consumption.

The RL agents are responsible for managing the charge and discharge operations of the CHW, DHW, and electricity storage systems by determining the amount of energy to store or release at each time step. CityLearn ensures that the building's heating and cooling energy demands are always met, regardless of the agent's decisions, by relying on pre-calculated building load profiles. These profiles account for space cooling, dehumidification, appliance usage, DHW requirements, and solar power generation. A backup controller is implemented to prioritize fulfilling the building's immediate energy needs before any excess energy is directed toward storage.

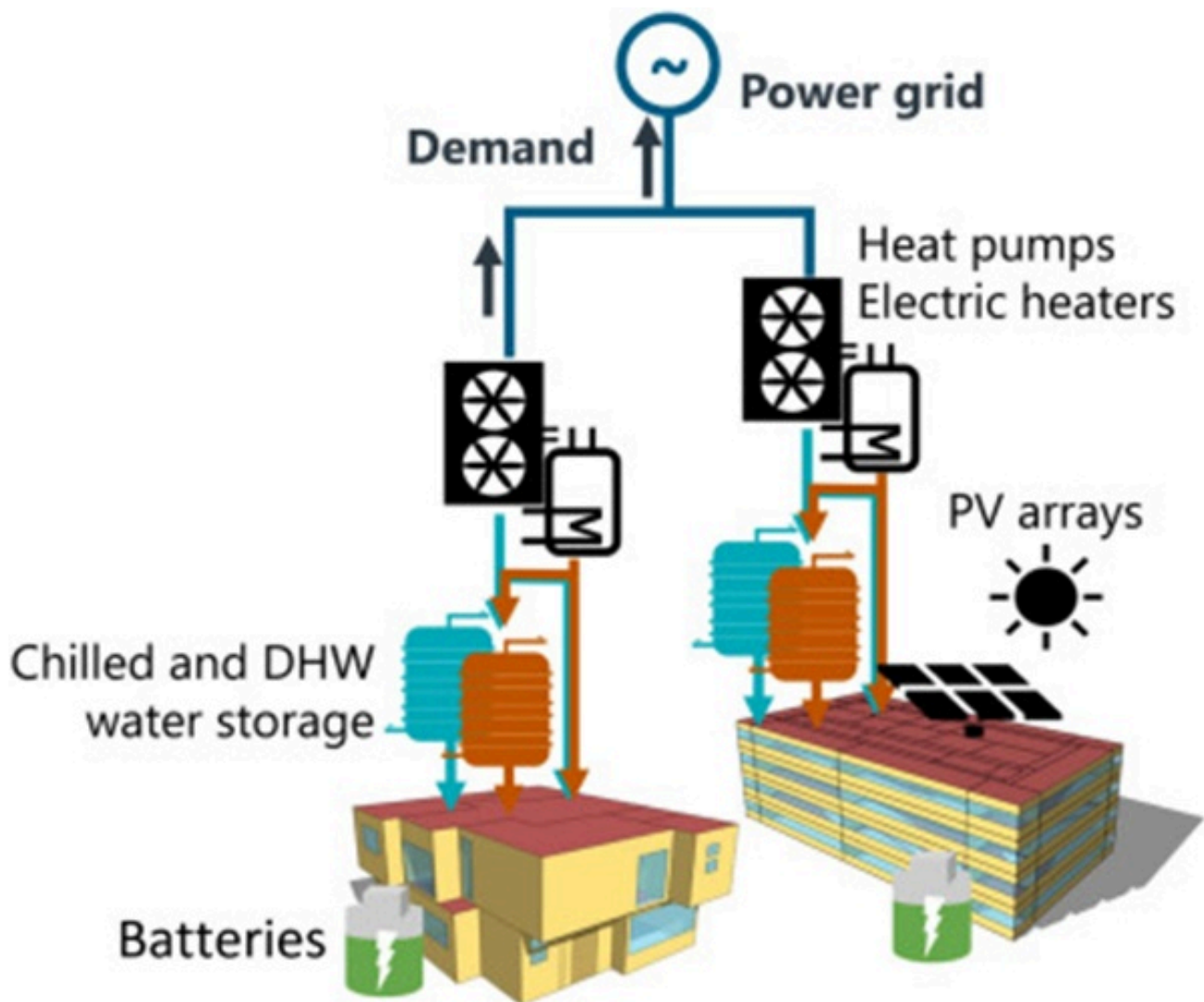


Figure 3. General architecture of the CityLearn simulation platform.

Researchers have frequently employed CityLearn as a common testbed when exploring different demand response techniques. The environment has supported work on incentive-driven load management [26], joint demand response programs involving multiple parties [27], synchronized control of energy resources across buildings [28, 29], and systematic performance comparisons among various reinforcement learning methods [30, 31].

Table 1. Storage capacities for chilled water (CHW), domestic hot water (DHW), and electricity (ELE), along with the size of photovoltaic (PV) systems installed in each building. The capacities of the CHW and DHW tanks are given as the number of hours during which the storage could cover the building's highest annual hourly cooling demand and DHW demand, respectively, if fully charged at the start.

ID	CHW	DHW	ELE	PV
Empty cell	Stg. (h)	Stg. (h)	Stg. (kWh)	(kW)
1	2	2	140	120
2	3	3	80	0
3	2	0	50	0
4	1.5	0	75	40
5	3.5	1.5	50	25
6	1.5	3	30	20
7	2	2	40	0
8	3	3	30	0
9	3	3	35	0

C8: Offline Pretraining from Recorded Logs of An External Rule-Based Controller

This part of the study leverages the CityLearn simulation platform to assess Challenge 8 (C8) under the GIB setup. The evaluation focuses on contrasting two different reinforcement learning strategies for managing building energy systems. The first strategy relies on fully independent soft actor-critic (SAC) controllers that operate without coordination. The second approach uses the MARLISA algorithm, which enables collaboration between multiple agents. Both strategies are tested inside the CityLearn environment with the nine-building dataset. We specifically examine how the performance of these agents evolves when they are first trained offline for varying lengths of time using fixed interaction logs collected from a rule-based controller (RBC) policy. Our core assumption is that providing agents with a longer period of offline learning from expert demonstrations will enhance their effectiveness once they begin online operation, as they will have already absorbed knowledge of high-quality control decisions.

Dataset

The experiments draw on the publicly available CityLearn Challenge 2021 dataset [32]. It features nine representative Department of Energy (DOE) prototype buildings: a medium office building (ID = 1), a quick-service restaurant (ID = 2), a standalone retail outlet (ID = 3), a strip mall retail property (ID = 4), and five medium-scale multifamily residential buildings (IDs 5 through 9) [33]. Hourly energy demand profiles for all buildings were generated in advance using EnergyPlus simulations, based on real weather records from Austin, Texas, spanning 2014 to 2017. Information regarding the capacities of chilled water storage, domestic hot water storage, electrical storage, and rooftop photovoltaic systems for each building is summarized in [Table 1](#).

Agent architecture and reward formulation

Standalone SAC agents

In environments with continuous state and action variables, conventional table-based Q-learning quickly becomes infeasible due to the exponential growth in the number of possible state-action pairs. Actor-critic methods overcome this limitation by using neural networks that generalize effectively across large, high-dimensional spaces. Here, the actor component receives the current environmental state and proposes the best actions according to its current policy. The critic component then judges the quality of those actions by predicting the corresponding Q-values, taking into account both the selected action and the state at the time it was chosen.

Soft Actor-Critic (SAC) belongs to the family of model-free, off-policy reinforcement learning algorithms [34]. Its off-policy design allows the agent to learn from past experiences multiple times, making the training process sample-efficient. The method is built around three fundamental principles: the actor-critic architecture, off-policy updates, and an entropy term that encourages broad exploration while stabilizing learning. In practice, SAC maintains and updates three separate neural networks: one for the policy (actor), one for the soft Q-function (critic), and one for the value function V. Readers interested in a deeper technical treatment of SAC are directed to Haarnoja et al. [35].

$$r_i^{\text{SAC}}(t) = \min(0, -e_i(t)^3) \quad (6)$$

Details on the neural network design and the specific hyperparameter values used for the SAC agents are presented in **Table 2**. These choices match the standard default settings recommended by CityLearn for the dataset being used.

The independent SAC reinforcement learning agents are guided using the reward $r_i^{\text{SAC}}(t)$ (Eq. 6), which is formulated as a single-agent metric determined solely by the net electricity usage $e_i(t)$ of agent i at a given time step t ; here, $e_i(t) > 0$ indicates that the building's consumption surpasses its generation, whereas $e_i(t) < 0$ signifies that the building is producing a surplus and is self-sufficient during that period.

Table 2. Hyperparameter settings for the independent SAC agents and the MARLISA multi-agent algorithm.

Variable	Value
Discount	0.99
Decay rate	0.005
Learning rate	0.0003
Batch size	256
NN hidden layer count	2
NN hidden layer size	256
Replay buffer capacity	100,000
Temperature	0.2
Training time steps	(744, 434, 8760)
Training episodes	1
Total time steps	35,040 (4 years)

MARLISA RL agents

MARLISA is based on the Soft Actor-Critic (SAC) algorithm. It enables coordination among multiple agents through reward sharing, collective reward signals, and the mutual exchange of selected information [36]. Each agent forecasts its own future electricity consumption and communicates this forecast to the other agents in a leader–follower structure. Through an iterative process, the agents converge on their chosen actions before implementation.

$$r_i^{\text{MARL}}(t) = \text{sign}(-e_i(t)) \cdot 0.01 \cdot e_i(t)^2 \cdot \max(0, \sum_{i=0}^n e_i(t)) \quad (7)$$

The MARLISA agents use the same network architecture and hyperparameter settings as the single-agent SAC models, as detailed in [Table 2](#).

The reward function for the MARLISA agents, denoted as $r_i^{\text{MARL}}(t)$ and given by [Eq. 7](#), integrates the individual building's net electricity consumption $e_i(t)$ with the aggregate district-level net electricity consumption $\sum e_i(t)$. This formulation facilitates information sharing among agents and incentivizes them to minimize the district's coordinated energy demand. Here, $e_i(t)$ takes a positive value when the building consumes more electricity than it produces, and a negative value when the building generates more electricity than it consumes.

RBC

Rule-based controllers (RBCs) are designed without assuming detailed prior knowledge of each building's energy profile. Two distinct variants of the RBC sequence of operation (SOO) are implemented: RBCBasic, described in [Algorithm 1](#), which follows a simplified and somewhat arbitrary logic, and RBCOptimized, outlined in [Algorithm 2](#), which incorporates domain expertise.

For both rule-based strategies, the inputs consist of the current hour of the day (h) and the time step (t), while the output is the charge or discharge action $a(t)$ applied to the chilled water, domestic hot water (DHW), or electricity storage systems. In [Algorithm 1](#), the action values are selected arbitrarily to represent a sub-optimally tuned controller. In contrast, the values in [Algorithm 2](#) are determined through a grid search over various hourly combinations, selecting the set that yields the strongest performance according to the evaluation metrics.

Both RBC variants operate greedily at the level of each building. They leverage the available storage capacity to shift energy use by charging the storage more intensively during nighttime hours—when the heat pumps achieve higher coefficients of performance—and discharging during daytime periods. Additionally, the RBCs serve as a baseline for normalizing the performance metrics of the reinforcement learning agents.

Input: h, t
Output: $a(t)$
if $9 \leq h \leq 21$ **then**
 | $a(t) = -0.08$;
else
 | $a(t) = 0.091$;
end

Algorithm 1. RBC_{Basic} sequence of operation

Input: h, t
Output: $a(t)$
if $1 \leq h \leq 6$ **then**
 | $a(t) = 0.05532$;
else if $7 \leq h \leq 15$ **then**
 | $a(t) = -0.02$;
else if $16 \leq h \leq 18$ **then**
 | $a(t) = -0.044$;
else if $19 \leq h \leq 22$ **then**
 | $a(t) = -0.024$;
else
 | $a(t) = 0.034$;
end

Algorithm 2. RBC_{Optimized} sequence of operation

Action-space design

The action space for each building is defined by the number of controllable energy storage systems available, including chilled water (CHW), domestic hot water (DHW), and electricity storage. Consequently, the dimensionality of the action space varies from 1 to 3, depending on whether a building is equipped with 1, 2, or all 3 of these storage systems. Each action value is constrained within the range of -1 to 1 , where positive values correspond to charging commands and negative values correspond to discharging commands.

State-space design

The state space consists of 27 observable variables encompassing temporal, weather, district-level, and building-specific information, as outlined in **Table 3**. To facilitate the learning process, these states undergo several preprocessing transformations: cyclical encoding for the month and hour variables, one-hot encoding for the day variable, and min–max normalization for all remaining states. The state of charge (SOC) for the storage systems is included only when the corresponding storage system is present in a given building. In contrast, the rule-based controllers (RBCs) rely exclusively on the hour variable when determining control actions.

Table 3. The unified state space for all agents.

State	Unit
Temporal	
Month	–
Day	–
Hour	–
Weather	
Dry-bulb temperature	°C
Dry-bulb temperature (+6 h)	°C
Dry-bulb temperature (+12 h)	°C
Dry-bulb temperature (+24 h)	°C
Relative humidity	%
Relative humidity (6 h)	%
Relative humidity (12 h)	%
Relative humidity (24 h)	%
Diffuse solar irradiance	W/m ²
Diffuse solar irradiance (6 h)	W/m ²
Diffuse solar irradiance (12 h)	W/m ²
Diffuse solar irradiance (24 h)	W/m ²
Direct solar irradiance	W/m ²
Direct solar irradiance (6 h)	W/m ²
Direct solar irradiance (12 h)	W/m ²

Direct solar irradiance (24 h)	W/m ²
District	
Net electricity consumption	kWh
Carbon intensity	kgCO ₂ /kWh
Building	
Zone dry-bulb temperature	°C
Zone relative humidity	%
Non-shiftable load	kWh
Solar generation	W
Chilled water storage SOC	–
Domestic hot water storage SOC	–
Electricity storage SOC	–

Performance metrics and cost functions

Agent effectiveness is measured by how well they can reduce four specific cost functions that reflect the district's overall energy flexibility. Let E_t stand for the net electricity use of the entire district at each time step t .

Average Daily Peak represents the yearly average of the highest daily electricity demand. It is calculated according to [Eq. 8](#), where d denotes the day of the year and i denotes the number of time steps within a day. For hourly simulations, i is fixed at 24.

$$\text{Average Daily Peak} = \frac{\sum_{d=0}^{364} \max(E_{i \times d}, \dots, E_{i \times (1+d) - 1})}{365} \quad (8)$$

1 - Load Factor equals one minus the ratio of average monthly electricity demand to the highest monthly demand. This is expressed in [Eq. 9](#), where m denotes the month and k is the total number of time steps in that month. In the present hourly simulations, k equals 730.

$$1 - \text{Load Factor} = \left(\sum_{m=0}^{11} 1 - \frac{\sum_{t=m \times k}^{k \times (1+m) - 1} E_t}{k \times \max(E_t, \dots, E_{k \times (1+m) - 1})} \right) \times \frac{1}{12} \quad (9)$$

Net Electricity Demand, formulated in [Eq. 10](#), sums only the positive components of net electricity use. The goal is to lower the district's actual energy consumption rather than relying on surplus production, which effectively favors self-sufficient (islanded) operation.

$$\text{Net Electricity Demand} = \sum_{t=0}^{n-1} \max(0, E_t) \quad (10)$$

Ramping, defined in [Eq. 11](#), measures the absolute change in net electricity consumption between successive time steps, where n is the total number of time steps satisfying $0 \leq t < n$.

$$\text{Ramping} = \sum_{t=1}^{n-1} |E_t - E_{t-1}| \quad (11)$$

Experimental setup

The experiments designed to examine the hypothesis involve two main variations: the type of rule-based controller sequence of operation (SOO) used to create the offline training logs, and the length of the offline training phase.

In each training run, the RL algorithms first undergo offline training for either the first 744 time steps (two weeks), 4,344 time steps (six months), or 8,760 time steps (one year). During this initial phase, control actions are taken directly from pre-generated fixed logs produced by either RBCBasic or RBCOptimized. After this period, the system switches to online learning, where the SAC or MARLISA agents take over action selection for the remaining time steps up to a total of 35,040 steps (four years).

The four RL agent configurations tested are:

- SACRBCBasic
- SACRBCOptimized
- MARLISARBCBasic
- MARLISARBCOptimized

This setup enables investigation of the influence of algorithm sophistication (single-agent SAC vs. coordinated MARLISA) and the benefit of basic vs. more advanced domain knowledge in the underlying rule-based controllers.

Every simulation consists of a single complete episode of 35,040 time steps, spanning the years 2014 through 2017. Each specific combination of offline training length and agent type is repeated three times within the CityLearn simulator using the nine-building dataset, each run starting with a unique random seed. Final results are computed as the average of these three repetitions.

The complete source code for this study is openly accessible in the CityLearn repository [\[37\]](#).

Results

Performance metrics

Figure 4 compares district-wide performance when all storage systems are managed solely by RBCBasic or RBCOptimized across the full four-year period. RBCOptimized achieves clearly better outcomes than RBCBasic on average daily peak, 1-load factor, and ramping. In contrast, the two rule-based methods show almost no difference in net electricity consumption, with RBCOptimized offering only a minor long-term benefit in this metric.

Figure 5 displays the district-level results for the different offline training durations and RL agents. All values are normalized against the performance of the RBC used during the offline phase (shown as a dashed black line). Scores below one indicate improvement over the RBC baseline, while scores above one indicate degradation.

The richer domain knowledge embedded in RBCOptimized yields stronger baseline performance, leading both SACRBCOptimized and MARLISARBCOptimized agents to fall short of it. Consequently, longer offline training periods with RBCOptimized slow down early progress but deliver superior final results. On the other hand, the simpler logic of RBCBasic creates a weaker starting point, so that RL agents trained on it generally perform worse than the RBC itself; in this case, longer training actually leads to poorer outcomes than shorter training.

For agents pretrained with RBCOptimized, the net electricity demand metric remains remarkably stable across different offline training lengths. A noteworthy pattern emerges during the two-week offline period: it produces a strong initial drop in net electricity consumption right after switching online, but this advantage fades within the first year and converges toward the weaker results observed with the six-month and one-year training durations.

When SAC and MARLISA are trained using the same RBC, algorithm complexity has little to no impact on average daily peak or 1 - load factor. For the ramping metric, MARLISARBCOptimized shows noticeably worse performance during short offline training but gradually recovers. Meanwhile, SACRBCOptimized agents maintain ramping levels close to those of the RBCOptimized baseline throughout.

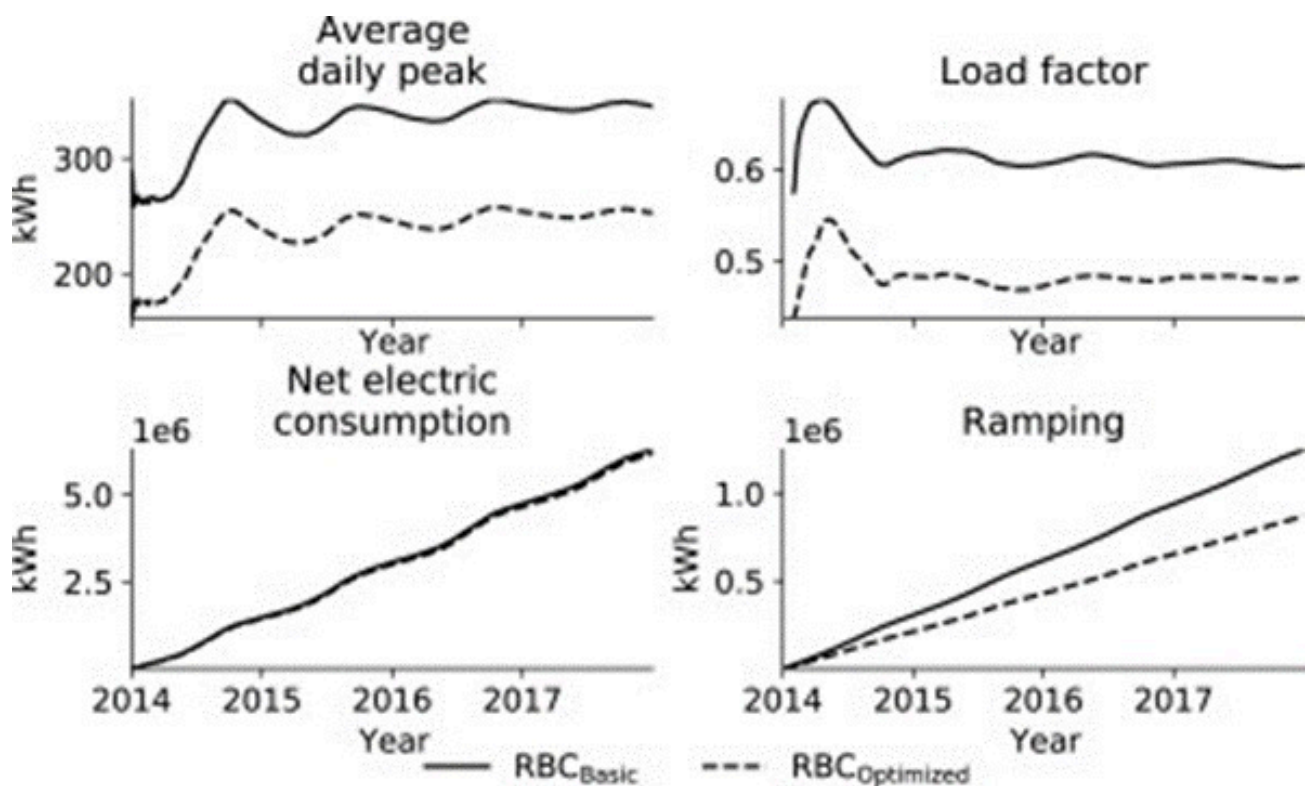


Figure 4. Comparison of district-wide energy flexibility indicators achieved when storage systems across all buildings follow the unchanging action sequences from either the basic or the optimized rule-based controller over the full four-year simulation horizon.

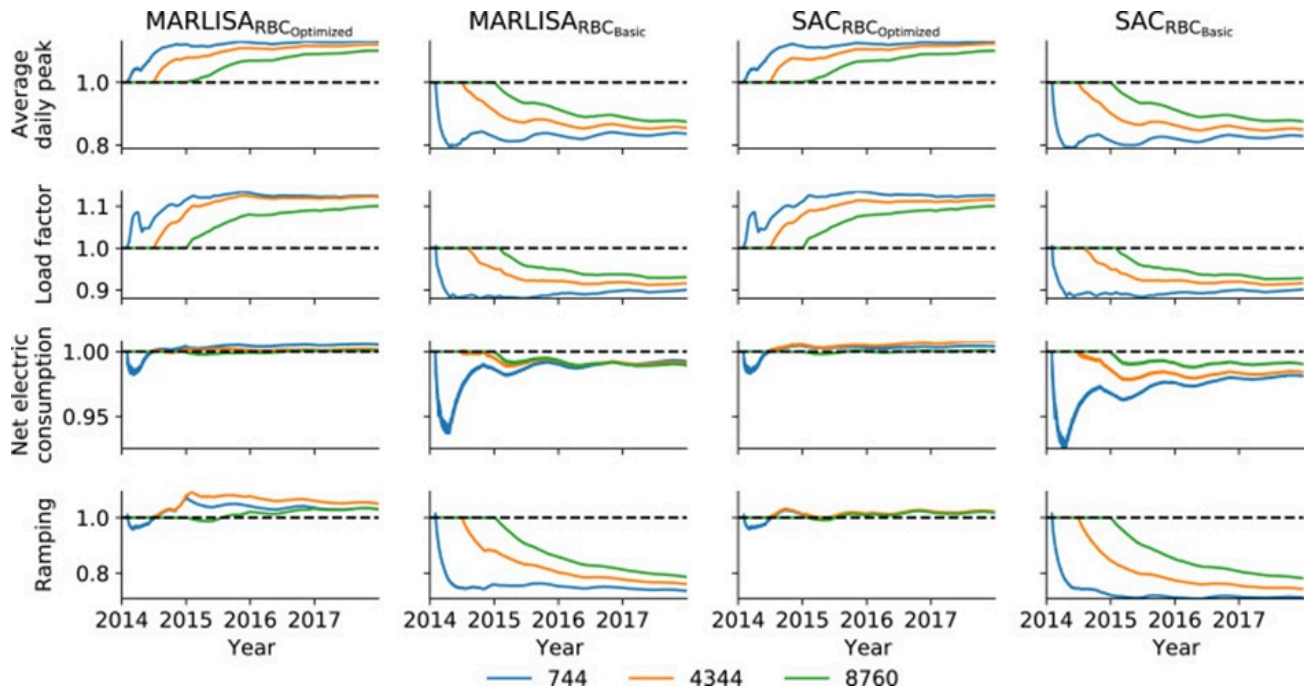


Figure 5. Energy flexibility results from CityLearn runs examining the effects of different offline training lengths and various RL agent setups. Blue, orange, and green traces correspond to offline training durations of 744 time steps (two weeks), 4,344 time steps (six months), and 8,760 time steps (one year), respectively. The four RL configurations tested are SACRBCBasic, SACRBCOptimized, MARLISARBCBasic, and MARLISARBCOptimized.

Every metric has been scaled relative to the performance of the particular rule-based controller that supplied the offline training data (represented by the dashed black reference line), with the RBC type noted in the agent label subscript. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

District electricity consumption

Figure 6 illustrates the district's net electricity usage patterns for the four RL agents after their respective offline training phases, along with a reference profile that assumes neither photovoltaic panels nor storage management. The selected window displays the first seven days of 2014 right after the end of a six-month offline period, and repeats the same seven-day window one year later in 2015.

In the 2014 window, agents pretrained for only two weeks or six months have already begun online learning, while those assigned a full year of pretraining remain under rule-based control. Immediately upon switching to online mode, the six-month pretrained agents closely mirror the behavior of the two-week pretrained ones, leading to virtually identical consumption traces six months later for both groups.

Still in 2014, the configuration continued offline training for one year, recording noticeably higher electricity use during early mornings and late nights, yet registers lower demand around midday than the agents already operating online. When the same seven-day period is examined in 2015, the consumption shapes look almost identical, no matter which rule-based controller provided the initial training, which RL method was applied, or how long the offline phase lasted.

Taken as a whole, the addition of solar generation combined with active storage management creates clear energy flexibility benefits compared with the fully uncontrolled case (no PV and no storage). The most evident improvement appears as effective peak reduction from late morning through the afternoon hours.

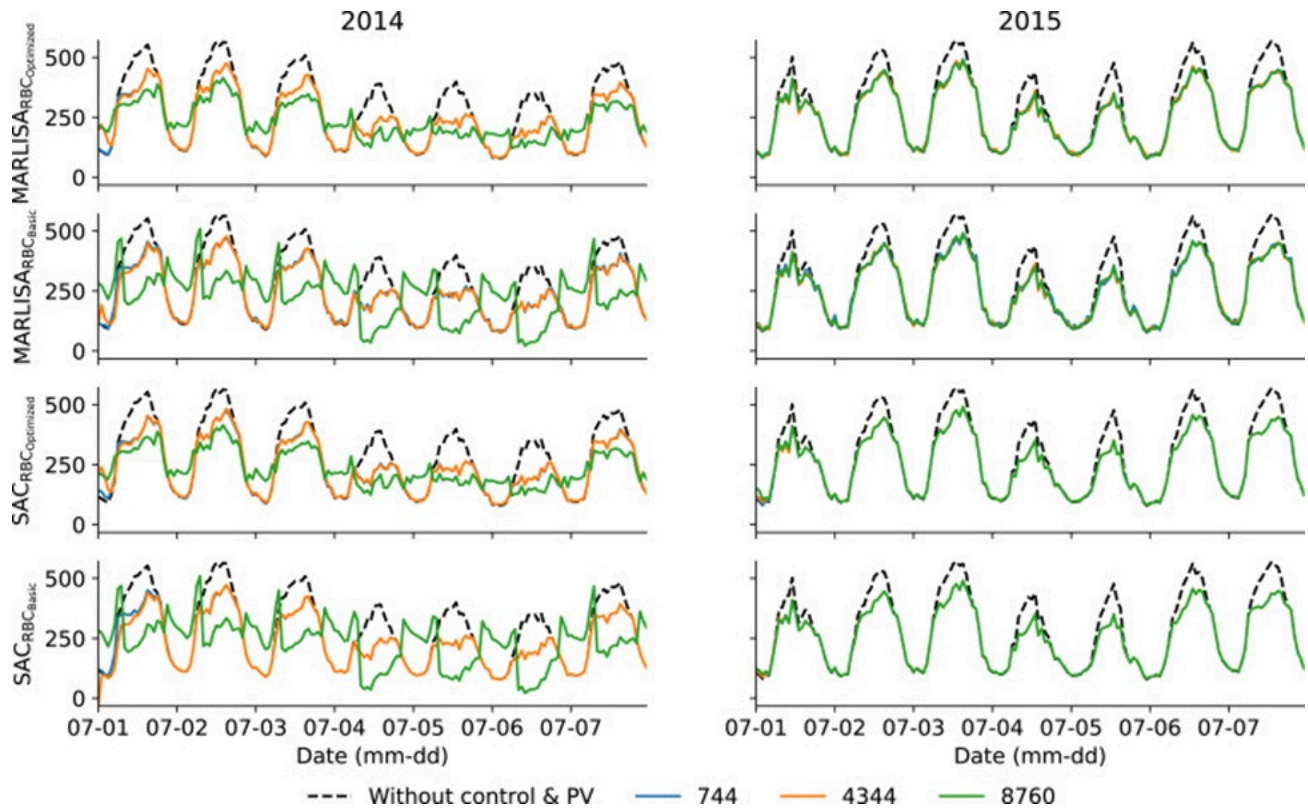


Figure 6. Net electricity usage at the district scale when employing RL-controlled chilled water (CHW), domestic hot water (DHW), and electrical storage systems together with photovoltaic (PV) generation, compared against the baseline electricity demand without any control or PV (shown as a dashed black line) across different durations of offline training. Offline training lengths tested were 744 timesteps (two weeks), 4344 timesteps (six months), and 8760 timesteps (one year), represented by the blue, orange, and green curves, respectively. The evaluated RL agents were SACRBCBasic, SACRBCOptimized, MARLISARBCBasic, and MARLISARBCOptimized. Results cover the first seven days following six months of simulation in 2014 (left panel) and the matching timeframe in 2015 (right panel). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Sophisticated building control strategies

More effective control systems for buildings are essential to move beyond conventional fixed setpoint approaches, which ignore forecasts and fail to optimize operational sequences [11]. Model Predictive Control (MPC) originated in the petrochemical sector during the 1970s and has since been implemented in various fields [38]. MPC relies on creating a detailed mathematical representation of the system under control, which performs effectively in standardized applications such as vehicles or aircraft. However, the distinctive characteristics of buildings and their associated energy setups, combined with the high costs of model creation and tuning, have prevented widespread adoption of MPC in the building sector despite ongoing progress [39, 40]. Reinforcement learning (RL) methods have emerged as a potential solution to MPC limitations by operating in a model-free manner. That said, RL typically demands substantially more data and computation time than MPC. Existing comparisons between the two are frequently one-sided and thus of limited value. The challenges outlined in this work deliberately target a wide range of realistic scenarios rather than isolated issues, enabling more equitable evaluations. Naturally, these challenges are equally suitable for benchmarking across different algorithmic categories.

A notable development within multi-agent reinforcement learning (MARL) is the centralized training with decentralized execution (CTDE) paradigm. Under CTDE, each agent's policy can be trained using the complete global state (i.e., combined observations from all agents), yet agents operate autonomously at runtime. This setup allows agents to

develop cooperative behaviors during training without requiring access to others' observations during deployment. A CTDE variant of MARLISA has demonstrated smoother control trajectories than the standard MARLISA method [27]. Nonetheless, any increase in algorithmic sophistication must be balanced with increased data needs, communication overhead, and concerns about data privacy.

Standardizing simulation environments

There is a strong case for establishing common benchmarks in building simulation platforms, including tools such as the COmprehensive Building Simulator (COBS) [41], Sinergym [42], BOPTeST [43], the Advanced Controls TestBed (ACTB), and CityLearn [19]. These should adopt a shared interface, such as OpenAI Gym [44], and provide open-source access to both datasets and code. Such standardization could trigger rapid innovation comparable to the surge in deep learning following the release of the ImageNet dataset [45]. Unlike the ImageNet-driven progress, however, greater interdisciplinary collaboration between building science experts and computer scientists would be advantageous. This would promote the integration of domain-specific building knowledge into controller development while helping translate algorithmic advances into real-world applications. Joint workshops, cross-community invitations, or collaborative events among groups such as ACM's BuildSys/e-Energy and ASHRAE/IBPSA could be developed to fully realize the built environment's capacity to cut greenhouse gas emissions.

Offline learning challenge (C8)

Our primary assumption for Challenge 8 (C8) was that extended offline training would improve agent performance by equipping them with comprehensive knowledge of optimal behaviors before online deployment. The experimental outcomes support this assumption, although specific implementation decisions influence results. Findings indicate that the single-objective optimization (SOO) applied within the rule-based controller (RBC) to produce the offline datasets plays a decisive role in determining subsequent RL performance across the four evaluation metrics. Prolonged offline training using data from an optimized RBC leads to slower initial adaptation once online but ultimately yields stronger long-term results. In contrast, agents trained on data from a basic RBC tend to exhibit declining performance as the offline period lengthens.

The optimized RBC substantially surpassed the RL-based controllers in lowering district-level average daily peaks, improving the 1 - load factor metric, and reducing ramping. This highlights the considerable demand-side flexibility gains achievable simply by refining current RBC implementations, often outperforming the deployment of more intricate control systems. However, traditional RBCs cannot adapt to unexpected changes in the operating environment (C5), a capability inherent to RL methods that can influence overall objective fulfillment. Future research will examine C5 in greater detail.

No meaningful performance gaps were observed between the SAC and MARLISA algorithms across the four key indicators. This implies that the simpler SAC approach is adequate for the task, and that the additional overhead associated with inter-agent information exchange in MARL may not be justified.

Differences in net electricity consumption were insignificant across offline training duration, RBC optimization level, and RL algorithm type. This outcome can be traced back to the underlying RBC strategy. Both the basic and optimized RBC variants prioritize charging storage systems during nighttime and early morning hours to exploit improved heat pump coefficients of performance (COP). Such logic also aligns with residential demand response (DR) programs that reward consumption during off-peak times. In the simulated setting without active DR incentives, this approach primarily benefits the CHW storage (served by a heat pump). DHW and electrical storage charging draw directly from the grid but can be partially offset by daytime solar production. Given the variable nature of solar availability, these storage units could benefit from opportunistic "free" charging during periods when the RBC initiates discharge.

One key difficulty in offline RL is the risk of training on uniform, low-diversity datasets generated by a fixed policy, which can produce policies that fail to generalize in live operation. Research by Yarats *et al.* underscores the significant effect of

the variety of exploration data on offline RL outcomes [46]. Our findings align with this, confirming that extended training on logs from a well-tuned RBC yields better results.

Conclusion

This paper presents a collection of challenges to investigate realistic grid-interactive buildings (GIBs). Although numerous open questions persist in this domain, we stress the importance of a coordinated community effort that tackles core computational issues while ensuring relevance to broader challenges in the built environment. As an illustration, we analyzed Challenge 8 (C8) by examining offline training from fixed logs of an external policy across three different training lengths, two degrees of domain expertise embedded in the behavior policy, varying RL algorithm complexities, and four distinct performance criteria. The provided list is not meant to represent all possible challenges. Instead, by focusing on common practical difficulties, we hope to encourage researchers to document and disseminate their simulation environments and associated problems, using these challenges as a shared reference framework.

Acronyms

AI = Artificial intelligence.
CHW = Chilled water.
COP = Coefficient of performance.
CTDE = Centralized training with decentralized execution.
DHW = Domestic hot water.
DOE = Department of Energy.
DR = Demand response.
EV = Electric vehicle.
GIB = Grid-interactive building.
HVAC = Heating, ventilation, and air conditioning.
MARL = Multi-agent reinforcement learning.
MDP = Markov Decision Process.
MG = Markov Game.
MPC = Model predictive control.
PV = Photovoltaic.
RBC = Rule-based controller.
RES = Renewable energy system.
RL = Reinforcement learning.
SAC = Soft actor–critic.
SARL = Single-agent reinforcement learning.
SOC = State of charge.
SOO = Sequence of operation.

Acknowledgements

None

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 02 Apr 2025

Revised: 14 May 2025

Accepted: 26 Jun 2025

Published online: 20 September 2025

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

Ürge-Vorsatz D. Fifth assessment report, mitigation of climate change. Intergov Panel Clim Change 2014;2014:674–738.

Leibowicz BD, Lanham CM, Brozynski MT, Vazquez-Canteli JR, Castejon NC, Nagy Z. Optimal decarbonization pathways for urban residential building energy services. *Appl Energy* 2018;230:1311–25.

<https://doi.org/10.1016/j.apenergy.2018.09.046>.

Mohagheghi S, Stoupis J, Wang Z, Li Z. Demand response architecture-integration into the distribution management system. *SmartGridComm* 2010;2010:501–6.

Vazquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Appl Energy* 2019;235:1072–89.

<https://doi.org/10.1016/j.apenergy.2018.11.002>.

Dupont B, Dietrich K, De Jonghe C. *Appl Energy* 2014;122:1–10.

<https://doi.org/10.1016/j.apenergy.2014.02.022>.

Department of Energy. A national roadmap for grid-interactive efficient buildings. Technical report. Washington, DC: Department of Energy; 2021. 166 p.

Siano P. Demand response and smart grids - a survey. *Renew Sustain Energy Rev* 2014;30:461–78.
<https://doi.org/10.1016/j.rser.2013.10.022>.

Bruninx K, Patteeuw D, Delarue E, Helsen L, D'Haeseleer W. Short-term demand response of flexible electric heating systems: The need for integrated simulations. In: *International Conference on the European Energy Market (EEM)*. 2013; p. 28–30.
<https://doi.org/10.1109/EEM.2013.6607333>.

Chourabi H, Nam T, Walker S, Gil-Garcia JR, Mellouli S, Nahon K, et al. Understanding smart cities: An integrative framework. In: *Proceedings of the Annual Hawaii International Conference on System Sciences*. 2011; p. 2289–97.
<https://doi.org/10.1109/HICSS.2012.615>.

Drgoña J, Arroyo J, Figueroa IC, Blum D, Arendt K, Kim D, et al. All you need to know about model predictive control for buildings. *Annu Rev Control* 2020;50:190–232.
<https://doi.org/10.1016/j.arcontrol.2020.09.001>.

Wang Z, Hong T. Reinforcement learning for building controls: The opportunities and challenges. *Appl Energy* 2020;269:115036.
<https://doi.org/10.1016/j.apenergy.2020.115036>.

Drgoña J, Tuor AR, Chandan V, Vrabie DL. Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy Build* 2021;243:110992.
<https://doi.org/10.1016/j.enbuild.2021.110992>.

Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge (MA): MIT Press; 2018. 550 p.

Nagy Z, Park JY, Vazquez-Canteli J. Reinforcement learning for intelligent environments: A tutorial. In: Gardoni P, editor. *Handbook of Sustainable and Resilient Infrastructure*. 1st ed. Routledge; 2018.

Busoniu L, Babuska R, Schutter BD. Multi-agent reinforcement learning: A survey. In: *2006 9th International Conference on Control, Automation, Robotics and Vision*. 2006; p. 1–6.
<https://doi.org/10.1109/ICARCV.2006.345353>.

Wölflé D, Vishwanath A, Schmeck H. A guide for the design of benchmark environments for building energy optimization. In: *BuildSys 2020 - Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 2020; p. 220–9.
<https://doi.org/10.1145/3408308.3427614>.

Dulac-Arnold G, Mankowitz D, Hester T. Challenges of real-world reinforcement learning. *arXiv* 2019.
<http://arxiv.org/abs/1904.12901>

Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Goyal S, et al. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Mach Learn* 2021.
<https://doi.org/10.1007/s10994-021-05961-4>.

Vázquez-Canteli JR, Kämpf J, Henze G, Nagy Z. Citylearn v1.0: An OpenAI Gym environment for demand response with deep reinforcement learning. In: *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 2019; p. 356–7.
<https://doi.org/10.1145/3360322.3360998>.

Watkins C, Dayan P. Technical note: Q-learning. *Mach Learn* 1992;8(3):279–92.
<https://doi.org/10.1023/A:1022676722315>.

Li Y. Deep reinforcement learning: An overview. *arXiv* 2017.
<https://doi.org/10.48550/ARXIV.1701.07274>.

Rabbat M, Nowak R. Distributed optimization in sensor networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks. Berkeley, California, USA. New York: ACM; 2004; p. 20–7.
<https://doi.org/10.1145/984622.984626>.

Zhang K, Yang Z, Liu H, Zhang T, Basar T. Fully decentralized multi-agent reinforcement learning with networked agents. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. PMLR; 2018;80:5872–81.
<https://proceedings.mlr.press/v80/zhang18n.html>

Vazquez-Canteli JR, Dey S, Henze G, Nagy Z. Citylearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management. arXiv 2020. <http://arxiv.org/abs/2012.10504>

Github. [n.d.]. <https://github.com/intelligent-environments-lab/CityLearn>

Deltetto D, Coraci D, Pinto G, Piscitelli MS, Capozzoli A. Exploring the potentialities of deep reinforcement learning for incentive-based demand response in a cluster of small commercial buildings. *Energies* 2021;14(10):2933.
<https://doi.org/10.3390/en14102933>.

Glatt R, Silva FL da, Soper B, Dawson WA, Rusu E, Goldhahn RA. Collaborative energy demand response with decentralized actor and centralized critic. In: Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. New York: ACM; 2021; p. 333–7.
<https://doi.org/10.1145/3486611.3488732>.

Pinto G, Piscitelli MS, Vázquez-Canteli JR, Nagy Z, Capozzoli A. Coordinated energy management for a cluster of buildings through deep reinforcement learning. *Energy* 2021;229:120725.
<https://doi.org/10.1016/j.energy.2021.120725>.

Kathirgamanathan A, Twardowski K, Mangina E, Finn DP. A centralised soft actor critic deep reinforcement learning approach to district demand side management through CityLearn. In: Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities. New York: ACM; 2020; p. 11–4.
<https://doi.org/10.1145/3427773.3427869>.

Dhamankar G, Vazquez-Canteli JR, Nagy Z. Benchmarking multi-agent deep reinforcement learning algorithms on a building energy demand coordination task. In: RLEM 2020 - Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings and Cities. 2020; p. 15–9.
<https://doi.org/10.1145/3427773.3427870>.

Qin R, Gao S, Zhang X, Xu Z, Huang S, Li Z, et al. Neorl: A near real-world benchmark for offline reinforcement learning. arXiv 2021. <http://arxiv.org/abs/2102.00714>

Vázquez-Canteli JR, Nagy Z. The CityLearn Challenge 2021. IQC; 2021.

Deru M, Field K, Studer D, Benne K, Griffith B, Torcellini P, et al. U.S. Department of Energy commercial reference building models of the national building stock. Golden (CO): National Renewable Energy Laboratory; 2011.

Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: ICML. 2018. <http://arxiv.org/abs/1801.01290>

Haarnoja T, Zhou A, Hartikainen K, Ha S, Tan J, Kumar V, et al. Soft actor-critic algorithms and applications. arXiv 2018. <http://arxiv.org/abs/1812.05905>

Vazquez-Canteli JR, Henze G, Nagy Z. MARLISA: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. In: BuildSys 2020 - Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. 2020; p. 170–9.
<https://doi.org/10.1145/3408308.3427604>.

GitHub. Citylearn. 2022. Retrieved September 2, 2022 from <https://github.com/intelligent-environments-lab/CityLearn/releases/tag/rl.challenges.paper.2022.0>

Morari M, Lee JH. Model predictive control: past, present and future. *Comput Chem Eng* 1999;23(4-5):667–82.
[https://doi.org/10.1016/S0098-1354\(98\)00301-9](https://doi.org/10.1016/S0098-1354(98)00301-9).

Kontes GD, Giannakis GI, Sánchez V, Agustin-Camacho P de, Romero-Amorrortu A, Panagiotidou N, et al. Simulation-based evaluation and optimization of control strategies in buildings. *Energies* 2018;11(12):3376.
<https://doi.org/10.3390/en11123376>.

Príara S, Cigler J, Váňa Z, Oldewurtel F, Sagerschnig C, Žáčková E. Building modeling as a crucial part for building predictive control. *Energy Build* 2013;56:8–22.
<https://doi.org/10.1016/j.enbuild.2012.10.024>.

Zhang T, 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: ACM; 2020; p. 314–5.
<https://doi.org/10.1145/3408308.3431119>.

Jiménez-Raboso J, Campoy-Nieves A, Manjavacas-Lucas A, Gómez-Romero J, 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: ACM; 2021; p. 319–23.
<https://doi.org/10.1145/3486611.3488729>.

Blum D, Arroyo J, Huang S, Drgoňa J, Jorissen F, Walnum HT, et al. Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings. *J Build Perform Simul* 2021;14(5):586–610.
<https://doi.org/10.1080/19401493.2021.1986574>.

Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. OpenAI Gym. *arXiv* 2016.
<http://arxiv.org/abs/1606.01540>

Deng J, Dong W, Li L-J, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE; 2009; p. 248–55.
<https://doi.org/10.1109/CVPRW.2009.5206848>.

Yarats D, Brandfonbrener D, Liu H, Laskin M, Abbeel P, Lazaric A, et al. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv* 2022.
<https://doi.org/10.48550/arXiv.2201.13425>.