Artificial Intelligence Review (2024) 57:173
https://doi.org/10.1007/s10462-024-10819-x

An experimental evaluation of deep reinforcement learning algorithms for HVAC control

Antonio Manjavacas[1] · Alejandro Campoy-Nieves[1] · Javier Jiménez-Raboso[1] · Miguel Molina-Solana[1] · Juan Gómez-Romero[1]

Accepted: 28 May 2024 / Published online: 13 June 2024
© The Author(s) 2024

Abstract
Heating, ventilation, and air conditioning (HVAC) systems are a major driver of energy consumption in commercial and residential buildings. Recent studies have shown that Deep Reinforcement Learning (DRL) algorithms can outperform traditional reactive controllers. However, DRL-based solutions are generally designed for ad hoc setups and lack standardization for comparison. To fill this gap, this paper provides a critical and reproducible evaluation, in terms of comfort and energy consumption, of several state-of-the-art DRL algorithms for HVAC control. The study examines the controllers' robustness, adaptability, and trade-off between optimization goals by using the Sinergym framework. The results obtained confirm the potential of DRL algorithms, such as SAC and TD3, in complex scenarios and reveal several challenges related to generalization and incremental learning.

Keywords Reinforcement learning · HVAC · Building energy optimization · Sinergym

1 Introduction

In recent decades, both global warming and climate change have been significantly spurred by the growth in energy demand from residential and non-residential buildings. The Global Alliance for Buildings and Construction reported that, in 2021, the building and construction sector was responsible for more than one third of the global energy consumption (IEA

✉ Antonio Manjavacas
manjavacas@ugr.es

Alejandro Campoy-Nieves
alejandrocn7@ugr.es

Javier Jiménez-Raboso
jajimer@correo.ugr.es

Miguel Molina-Solana
miguelmolina@ugr.es

Juan Gómez-Romero
jgomez@decsai.ugr.es

[1] Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18071 Granada, Spain

2021; GlobalABC 2021)—despite the reduction in commercial and industrial activities caused by the COVID-19 pandemic. According to the International Energy Agency (IEA 2021), buildings are responsible for 17% and 10% of global direct and indirect CO_2 emissions, respectively.

Heating, ventilation, and air conditioning (HVAC) systems are one of the main sources of energy consumption in buildings, representing more than 50% of their associated energy demand in developed countries (Pérez-Lombard et al. 2008). This consumption is potentially affected by a lack of precise control over these systems, whose proper and efficient operation is vital to ensure energy savings (Wang et al. 2020; Mawson and Hughes 2021; Gholamzadehmir et al. 2020).

Many strategies have been implemented to improve HVAC control efficiency. In fact, most common control solutions are based on Rule-Based Controllers (RBC) combined with Proportional-Integrative-Derivative (PID) control (Geng and Geary 1993), which stand out for their simplicity and efficiency (Borase et al. 2021). By using heuristic rules, these controllers can guarantee proper comfort temperature ranges while reducing energy consumption (ASHRAE 2021). However, this type of controllers are far from offering optimal behavior, as they are mostly reactive, hardly cover the complexity of environments influenced by many variables—and therefore, many rules, and are guided by fixed and predetermined control sequences that heavily rely on expert knowledge (Wang and Hong 2020; Salsbury 2005). Generally, RBCs lack scalability, as they do not address building energy optimization on a global level, but on a local one. This is because a control where so many variables are considered would result in an overly complex RBC, being practically impossible to generalize their rules at a building level (Privara et al. 2013; Serale et al. 2018).

Model-based solutions, such as Model Predictive Control (MPC) (Yao and Shekhar 2021; Salakij et al. 2016; Morari and Lee 1999), are an alternative to reactive controllers. These controllers use physical models of buildings to simulate their thermal dynamics and analytically derive optimal HVAC control. Usually, MPC consider not only the characteristics of the environment but also other constraints and contextual data such as occupancy or weather. This ability to characterize and predict environmental conditions makes MPC outperform reactive controllers (Yao and Shekhar 2021; Kümpel et al. 2021; Efheij et al. 2019). Nevertheless, MPC also suffers from certain limitations. In addition to the high computational power it demands, precise system calibration is necessary to achieve the expected performance, which is not easily scalable since each building is unique in its layout and thermodynamic properties (Gomez-Romero et al. 2019). This poses a challenge in terms of cost-effectiveness, as many factors must be considered: materials, occupancy, end-use, location, orientation, etc. Consequently, this has resulted in a relatively small number of buildings currently implementing MPC strategies compared to 'if-then-else', 'on/off' or 'bang-bang' RBC controllers, as well as PID where digital control and variable frequency drives are available (Serale et al. 2018).

Given the shortcomings of these methods, Reinforcement Learning (RL) has been recently proposed as a viable alternative for complex control problems. RL is a computational learning method focused on the interaction of an agent with its environment, either real or simulated. This is an iterative learning method, based on trial and error, where a reward function makes the agent lean towards preferable actions or states. Therefore, the agent's goal will be to discover which actions lead to the maximization of the expected reward (Sutton and Barto 2018). The combination of RL with deep neural networks has led to a growing application of Deep Reinforcement Learning (DRL) in numerous domains (Mnih et al. 2015; Gibney 2016; Gupta et al. 2021), including HVAC control (Barrett and

Linder 2015; Azuatalam et al. 2020; Perera and Kamalaruban 2021; Yang et al. 2021; Fu et al. 2022). Accordingly, DRL can learn sophisticated control strategies from data (Biemann et al. 2021), generally obtained from building simulations, while using more computationally efficient building models than MPCs (Deng et al. 2022).

Nevertheless, as highlighted in Vázquez-Canteli and Nagy (2019), Findeis et al. (2022), most of the DRL proposals for HVAC control in the literature pick one or few algorithms without substantial motivation, lack a comprehensive analysis of them under controlled and assorted conditions, and cannot be easily reproduced. Motivated by this gap, this paper performs a comprehensive experimental evaluation of state-of-the-art DRL algorithms applied to HVAC control. The main contribution of this work is offering insights on which algorithms are most promising in different building energy control scenarios, and what possibilities arise for their further improvement. To this aim, we focus on algorithms performance (evaluation metrics, comfort-consumption trade-off, convergence, etc.), but also in their robustness against changing conditions and capabilities for transference to different scenarios. The study relies on SINERGYM,[1] an open-source building simulation and control framework for training DRL agents (Jiménez-Raboso et al. 2021). SINERGYM offers a standardized and flexible design that facilitates the comparison of algorithms under different environments, reward functions, states and actions spaces, as well as experiments replication.

The remainder of this paper is structured as follows. Section 2 details the theoretical background in which our research is framed. Section 3 introduces the main fundamentals of DRL and its application in HVAC control. Sections 4 and 5 will describe the environment and the experiments conducted, with the subsequent discussion on the results in Sect. 6. Finally, Sect. 7 will detail the main conclusions derived from this research.

2 Related work and novelty

The application of DRL in HVAC control is a developing field that has generated significant interest and growth within a broad community of researchers. Although the first approach to applying RL in building control dates back to 1998 by Mozer (1998), the development of this field really began to be remarkable in the last few years, together with the rise of DRL.

Extensive research about the application of DRL in building energy control and HVAC can be found in recent literature reviews, such as Vázquez-Canteli and Nagy (2019), Yu et al. (2021), Wang and Hong (2020), Han et al. (2019), Leitao et al. (2020), Mason and Grijalva (2019), Rajasekhar et al. (2020), Zhang et al. (2018), Yang et al. (2020), Mocanu et al. (2018), where several applications, perspectives, and objectives within this field are summarized and studied in detail.

Focused on the DRL methods being used, we find many applications of well-established algorithms, such as Deep Q-Networks (DQN) (Lissa et al. 2021; Yoon et al. 2019; Gupta et al. 2021; Sakuma et al. 2020), Deep Deterministic Policy Gradient (DDPG) (Gao et al. 2020; Zou et al. 2020), or actor-critic methods (Morinibu et al. 2019; Wang et al. 2017). However, we share the opinion of Biemann et al. (2021) on the current situation in this field: (1) the widespread use of DQN—instead of more modern algorithms such as Twin

[1] https://github.com/ugr-sail/sinergym.

Delayed DDPG (TD3), Soft Actor-Critic (SAC) or Proximal Policy Optimization (PPO)—in this continuous control problem could be hindering significant progress; and (2) these are generally isolated implementations that compare the performance of the chosen algorithm against a simplistic baseline and solve specific problems in well-delimited domains. In fact, there is little work benchmarking multiple DRL algorithms under standard conditions, as is common in other fields of machine learning (Vázquez-Canteli and Nagy 2019; Wölfle et al. 2020; Islam et al. 2017).

One of the few exceptions is the aforementioned work by Biemann et al. (2021), which develops a comparison between different DRL algorithms in a two-zone data center[2] environment. This is a well-known and commonly used test scenario in DRL-based HVAC control (Moriyama et al. 2018; Zhang et al. 2019b), in which the objective is the management of temperature setpoints and fan mass flow rates of a data center. Specifically, the algorithms they compared were SAC, TD3, PPO, and Trust Region Policy Optimization (TRPO), evaluating energy savings, thermal stability, robustness, and data efficiency. Their results revealed the predominance of off-policy algorithms and, specifically, SAC, achieving consumption reductions of approximately 15% while guaranteeing comfort and data efficiency. Finally, the robustness was remarkable when the DRL algorithms were trained using different weather models, which increased their generalization and adaptability.

Another remarkable and recent study is that of Brandi et al. (2022), where the control of an HVAC system based on the charging and discharging of a refrigerated storage facility is presented. In this case, two RBCs and an MPC controller were compared with two SAC implementations: one trained offline—in simulation, and the other one online—on the real building. The authors obtained the best results for MPC and offline DRL, as well as an interesting approximation of the online trained SAC to the performance of its offline variant over time. A Long Short-Term Memory (LSTM) neural network architecture was used to predict the value of the variables composing the agents' observations over a given prediction horizon. The authors conclude that DRL-based solutions are more unstable than MPC solutions, although they exhibit adaptation to recurrent patterns without explicit supervision.

Regarding generalization capacity and robustness, few papers address this issue. In Xu et al. (2020), transfer learning (Torrey and Shavlik 2010; Taylor and Stone 2009) is proposed to train and evaluate a DQN-based agent in a building layout, and then transfer it to a slightly different one. In this work, the controller uses two sub-networks: a front-end network, which captures building-agnostic behavior, and a back-end network, specifically trained for each building. This way, transfer learning avoids cold-starts and tuning in the second building, allowing for rapid deployment and greater efficiency. Different experiments are carried out, such as (1) transfer from n-zone to n-zone buildings with different materials and layouts; (2) transfer from n-zone to m-zone; and (3) transfer from n-zone to n-zone with different HVAC equipment. A second related work is Lissa et al. (2021), which focuses on spatial and geographical variations in HVAC control systems. Among their main achievements, we can find a significant reduction in the time required to reach comfort temperatures, although the authors do not delve into energy efficiency.

At this point, we can identify certain limitations and research gaps in the relevant but scarce literature addressing the experimental evaluation of DRL algorithms in HVAC control:

- Studies such as Biemann et al. (2021) and Brandi et al. (2022) only provide results in a single environment. In contrast, benchmarking is desirable to be performed using a representative enough set of scenarios and test configurations.

[2] https://github.com/NREL/EnergyPlus/blob/develop/testfiles/2ZoneDataCenterHVAC_wEconomizer.idf.

- Something similar happens when it comes to the application of transfer learning: the existing case studies are limited and use different environments, so they do not enable comparisons to be made. We believe that further deepening of its application is necessary, concretely by promoting, as far as possible, the generalization capacity and robustness of different agents in a broader set of testing environments.
- There are additional opportunities close to transfer learning that have not yet been addressed, such as the application of sequential learning. The objective pursued is to compare if a controller that learns progressively to manage an HVAC system under different weather conditions can be more efficient than one directly trained on a fixed setting.
- Finally, we consider it relevant to study how the manipulation of the reward function used by DRL agents affects their performance, and what consequences these changes may have in terms of consumption and comfort violation.

3 Theoretical background

3.1 Deep reinforcement learning

A reinforcement learning problem is defined as a Markov Decision Process (MDP) consisting of the following elements:

- An agent, which learns from the interaction with its environment over a discrete sequence of time steps $\mathcal{T} = \{0, 1, 2, ...\}$ and that pursues a certain objective.
- An environment, defined as any element external to the agent. It is a dynamic process that produces relevant data for the agent. A state $s_t \in \mathcal{S}$ represents the current situation of the environment at time step t.
- A set of actions $\mathcal{A}$, which determine the dynamics of the environment. Every action $a_t \in \mathcal{A}$ allows the agent to transit between different states.
- A reward function that evaluates the goodness of an action or state for the agent. The reward signal $r_t \in \mathbb{R}$ drives the agent's training.
- A policy function $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which determines the actions to be taken by the agent in the face of different states.

The goal of an RL agent is to achieve optimal behavior towards accomplishing a task, which involves learning an optimal policy that maximizes cumulative reward (Zhang and Yu 2020; Sutton and Barto 2018). Learning this optimal policy can be solved through dynamic programming in problems with not-particularly-large action and/or state spaces. For more complex cases, there are a variety of algorithms based on iterative improvement of the agent's policy or successive approximation of the expected reward of states and actions, e.g. Monte Carlo, SARSA, or the widely used Q-learning (Watkins and Dayan 1992).

However, these methods generally become infeasible as the complexity of the environment increases. This may be due to considerable growth in the number of actions or states that compose the environment, which can be mitigated by ad hoc methods that are difficult to generalize. Here is where deep learning comes into play, as neural networks can be applied to learn abstract representations of states and actions, as well as to approximate the expected rewards based on historical data. The combination of RL with deep neural

Table 1 Some of the most widely used DRL algorithms, including their main features and limitations

Algorithm	Main features and limitations
DQN	+ Simple value-based off-policy algorithm + Implements a replay buffer and a target network – Limited to discrete action spaces and prone to overestimating action values – Improved by variants like DDPG
DDPG	+ Off-policy algorithm + Improves DQN by allowing continuous action spaces – Highly sensitive to hyperparameterisation
A2C/A3C	+ On-policy gradient-based algorithms focused on parallelisation + Based on the joint learning of multiple agents implementing local policies + A2C includes a coordinator to synchronise all agents – Underperforms more complex policy-gradient methods
TRPO	+ On-policy gradient-based algorithm + Employs a trust region defined by a KL-divergence constraint to limit policy updates – More complex and less stable than PPO
PPO	+ On-policy algorithm based on policy gradient + Improves TRPO by adding a clipped objective to prevent large policy updates + Reduced computational costs – Data-inefficient
TD3	+ Off-policy actor-critic algorithm + Extends DDPG with improved stability and convergence guarantees + Clipped double-Q learning, delayed policy updates and target policy smoothing – May experience action saturation
SAC	+ Off-policy actor-critic algorithm + Employs entropy regularization to promote stability and exploration + Less sensible to hyperparameterisation – Slow training

networks has given rise to Deep Reinforcement Learning (DRL), which has lead to numerous applications of reinforcement learning in real environments. Thus, DRL algorithms combine the best of both worlds, endowing reinforcement learning with the representational power, efficiency, and flexibility of deep learning (Zai and Brown 2020).

In recent years, we have observed several advances in the development of DRL algorithms that represent the state of the art, such as Deep Q-Networks (DQN) (Mnih et al. 2013), Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. 2015), Asynchronous Advantage Actor-Critic (A3C) and Advantage Actor-Critic (A2C) (Mnih et al. 2016), Trust Region Policy Optimization (TRPO) (Schulman et al. 2017b), Proximal Policy Optimization (PPO) (Schulman et al. 2017a), Twin Delayed DDPG (TD3) (Fujimoto et al. 2018), and Soft Actor-Critic (SAC) (Haarnoja et al. 2018). Table 1 summarises the main features and limitations of these algorithms.

As we will see in the following subsection, some of these algorithms are powerful tools for solving continuous control problems with an indefinite number of states and actions (Duan et al. 2016), as is the case of HVAC control.

3.2 HVAC control problem formulation

The feasibility of HVAC control through DRL was firstly demonstrated by using reduced action spaces and simplified building models (Zhang et al. 2019a; Moriyama

et al. 2018). The objective of this problem is to find an optimal control policy that maximises the comfort of occupants while minimising energy consumption. The problem is formulated as a Partially Observable MDP (POMDP), and the goal is to find a policy that includes as many desirable states as possible while avoiding energetically-costly actions. Additional parameters, such as energy price and the use of renewable energy sources, can also be considered in the optimization process. However, for the sake of simplicity, we will focus on comfort and consumption as the primary targets.

Based on a set of observed variables that define the ambient conditions of the environment—i.e., outdoor/indoor temperature, CO_2 concentration, humidity, occupancy, the following objectives are considered:

- Regarding power demand P, we seek a policy that leads to its minimization:

$$\pi^* = \underset{\pi_\theta}{argmin} \sum_{t=1}^{T} P_t \tag{1}$$

- In the case of comfort, we try to minimize the distance C between the current state of the building, S_t, and a target state, S_{target}:

$$\pi^* = \underset{\pi_\theta}{argmin} \sum_{t=1}^{T} C(S_t, S_{target}) \tag{2}$$

A state will be desirable if the variables that compose it are within certain pre-established preferences. Therefore, we say that a comfort violation occurs if the value of these variables differs from the established limits.

In this problem, P corresponds to the consumption of the building's thermal equipment—e.g., heat pump, boiler—while $C(S_t, S_{target})$ refers to the difference between the actual and desired temperatures in the controlled zones of the building.

Therefore, we can combine the minimization of power demand and the maximization of comfort in a single expression:

$$\pi^* = \underset{\pi_\theta}{argmin} \sum_{t=1}^{T} \omega\, C(S_t, S_{target}) + (1-\omega)\, P_t \tag{3}$$

with ω and $(1-\omega)$ being the weights assigned to comfort and power demand, respectively. We can now define our reward function such that:

$$r(S_t, A_t) = (1-\omega)\, \lambda_P\, P_t + \omega\, \lambda_C\, C(S_t, S_{target}) \tag{4}$$

where P_t is the power demand of the HVAC system at time step t; $C(S_t, S_{target})$ represents the distance from the current zone temperature to the desired comfort limits; ω represent the weighting assigned to each part of the reward, and λ_P and λ_C, two constant factors used for scaling the magnitudes of power demand and temperature.

It is common to express this reward in negative terms, since its maximization is sought, which leads us to rewrite it as:

$$r(S_t, A_t) = -(1-\omega)\, \lambda_P\, P_t - \omega\, \lambda_C\, C(S_t, S_{target}) \tag{5}$$

Note how the reward function employed will vary depending on the variables to be considered in the problem. As discussed later, the details of this function is one of the main design decisions in DRL.

Finally, the actions to be performed by the agent will depend on the problem we are facing with. For example, in problems where the actions consist of adjusting the heating and cooling setpoints of the HVAC system, the action space may be discrete—there is a finite number of actions where each action is a tuple with fixed setpoint values, or continuous—each setpoint is a real number to be adjusted. Other similar problems can be addressed with a similar setup, such as air flow regulation (Raman et al. 2020) or lighting control (Chen et al. 2018).

3.3 Simulation models

Most of the DRL solutions proposed in the literature are not directly trained on a real building, but they make use of different simulation tools. In the HVAC context, https://energyplus.net/ and https://modelica.org/ are the most widely used frameworks. The use of this type of simulators is motivated by the fact that training a DRL agent in a real scenario would be too inefficient, as there is a need to establish a complete mapping between states, actions, and rewards, also considering extreme cases (Brandi et al. 2020). In fact, it has been observed that it takes 20–50 days of training to converge on an acceptable control policy (Fazenda et al. 2014; Costanzo et al. 2016; Vázquez-Canteli et al. 2019), thus making impractical to directly train DRL agents while they are deployed.

Typically, simulators are combined with deep learning (e.g., TensorFlow, Keras, PyTorch) or DRL libraries (e.g., Stable Baselines, RLlib, TensorFlow Agents) to pre-train and test algorithms in simulated environments prior to deployment (Valladares et al. 2019; Vázquez-Canteli et al. 2019). However, communication between DRL agents and simulation engines is rarely straightforward, which leads us to tools such as Boptest (Blum et al. 2021), Energym (Scharnhorst et al. 2021), RL-testbed (Moriyama et al. 2018) or Sinergym. These tools enable the communication between DRL agents and energy simulators, providing the necessary tools to perform the training and evaluation of the algorithms in different settings.

In the case of Sinergym, this software acts as a wrapper for EnergyPlus, offering several tools that enable DRL agents execution, data logging, the configuration of simulation environments, as well as the customization of states, actions and rewards. Its similarities and distinctive features with respect to other alternatives are detailed in Table 2.

4 Data and environments

All the experiments were formulated and executed using the Sinergym framework (Jiménez-Raboso et al. 2021). In the following subsections we present the configuration details, including state and action spaces, reward functions and weather data.

4.1 Building models

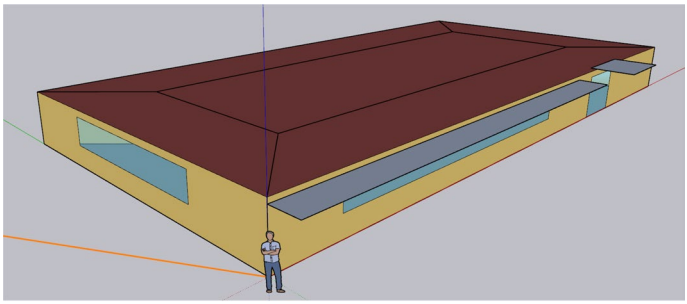
The following building models included in Sinergym, were chosen as test scenarios:

- 5ZoneAutoDXVAV (Wei et al. 2017; Ding et al. 2020). This is a single-story rectangular office building with dimensions of 30.48 m × 15.24 m (see Fig. 1a). The building is

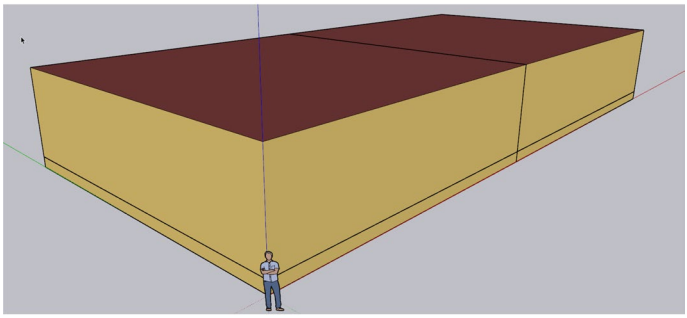
Table 2 Frameworks for DRL-based building energy optimization

Framework	Custom spaces	Actions	Weather config.	Custom rewards	Simulator	Envs.
RL TestBed for EnergyPlus (Moriyama et al. 2018)	✗	Continuous	✗	✗	EnergyPlus 9.5.0	3
Energym (Scharnhorst et al. 2021)	✗	Continuous	✓	✓	EnergyPlus 9.5.0 Modelica 2.14	14
BOPTEST (Blum et al. 2021)	✓	Discrete, continuous	✗	✓	Modelica 4.0.0	7
Sinergym (Jiménez-Raboso et al. 2021)	✓	Discrete, continuous	✓	✓	EnergyPlus 23.1.0	87

Custom spaces observation and actions spaces customization, *Actions* support for continuous and discrete action spaces, *Weather config.* weather customization capabilities, *Custom rewards* support for customized reward functions, *Simulator* building simulation engine, *Envs.* number of environments available. Adapted and updated from Jiménez-Raboso et al. (2021)



(a) 5ZoneAutoDXVAV representation



(b) 2ZoneDataCenterHVAC representation

Fig. 1 Building models used as benchmark environments

divided into five zones: four exterior zones and one interior zone. The average height of the building is 3.048 m and the total surface area is 463.6 m^2. it is equipped with a packaged Variable Air Volume (VAV), Direct Expansion (DX) cooling coil and gas heating coils, with fully auto-sized input as the HVAC system to be controlled.

- 2ZoneDataCenterHVAC (Moriyama et al. 2018; Zhang et al. 2019b; Li et al. 2019; Biemann et al. 2021). A 1-story data center with a surface of 491.3 m^2 (see Fig. 1b). The building is divided into two asymmetrical zones (east and west). Each data center zone has an HVAC system consisting on an air economizer, direct and indirect evaporative coolers, a single speed DX cooling coil, chilled water coil, and VAV with no reheat air terminal unit.

In both cases, we aim to regulate indoor temperatures by balancing comfort and power demand while also considering the influence of one on the other.

4.2 Weather

Given the importance of weather conditions in building control strategies (Ghahramani et al. 2016), the experimentation was conducted considering three types of weathers integrated in Sinergym, based on Typical Meteorological Year (TMY3) data obtained from U.S. Department of Energy: Prototype Building Models | Building Energy Codes Program (2021). Note how these climates vary in their average temperatures and humidity levels, thus providing a diverse and representative test set. These are:

- *Hot dry* Climate corresponding to Arizona (USA), with an average annual temperature of 21.7 °C and an average annual relative humidity of 34.9%.
- *Mixed humid* Climate in New York (USA), with an average annual temperature of 12.6 °C and an average annual relative humidity of 68.5%.
- *Cool marine* Corresponds to Washington (USA). The average annual temperature is 9.3 °C, while the average annual relative humidity is 81.1%.

Sinergym offers the possibility of adding random variations to the weather data between episodes by using Ornstein–Uhlenbeck (OU) processes (Benth and Šaltytė-Benth 2005; Jiménez-Raboso et al. 2021). This enables a more varied training aimed at preventing overfitting and improving generalization. Thus, in this work, stochasticity was achieved by applying OU with the following parameters: $\sigma = 1.0$, $\mu = 0$ and $\tau = 0.001$.

4.3 Observation and action spaces

At each simulation step, the agent receives information from the environment in the form of an observation $o_t \in \mathcal{O} \subset \mathcal{S}$. The information received consists of a subset of variables that provide information about the current state of the building, including comfort and consumption metrics, occupancy, time information, indoor and outdoor temperatures, humidity, wind, or solar radiation. These variables may vary in number or zone depending on the building we are trying to control. However, EnergyPlus uses common identifiers shared by different buildings, which facilitates the management of this information.

Table 3 Observation variables in 5ZoneAutoDXVAV and 2ZoneDataCenterHVAC

Variable name	Min.	Max.
Site outdoor air drybulb temperature	− 5e6	5e6
Site outdoor air relative humidity	− 5e6	5e6
Site wind speed	− 5e6	5e6
Site wind direction	− 5e6	5e6
Site diffuse solar radiation rate per area	− 5e6	5e6
Site direct solar radiation rate per area	− 5e6	5e6
Zone thermostat heating setpoint temperature*	− 5e6	5e6
Zone thermostat cooling setpoint temperature*	− 5e6	5e6
Zone air temperature*	− 5e6	5e6
Zone thermal comfort mean radiant temperature*	− 5e6	5e6
Zone air relative humidity*	− 5e6	5e6
Zone thermal comfort clothing value*	− 5e6	5e6
Zone thermal comfort Fanger model PPD*	− 5e6	5e6
Zone people occupant count*	− 5e6	5e6
People air temperature*	− 5e6	5e6
Facility total HVAC electricity demand rate	− 5e6	5e6
Current year	1	2050
Current month	1	31
Current day	1	12
Current hour	0	23

In 2ZoneDataCenterHVAC, variables marked with an asterisk are recorded for both east and west zones

Therefore, Table 3 contain the variables that make up an observation for 5ZoneAutoDXVAV and 2ZoneDataCenterHVAC, as well as their ranges.[3] Both the observed and controlled variables are extracted from the simulation of each building, with 20 for 5ZoneAutoDXVAV and 29 for 2ZoneDataCenterHVAC. Depending on the building and simulation conditions, each variable can take values in a wide range that we are unable to determine a priori. Therefore, most of the variables are assumed to take values in $[-5e6, 5e6]$, as these are the minimum and maximum limits of the simulator outputs. However, all the observation values were normalized in [0, 1] using the normalization wrapper provided by Sinergym.

Note that a comprehensive analysis of the variables required for training will depend on the problem's context and the scope of the controller's application. In this case, to train all agents with the most information available, we did not explicitly conduct a detailed study of these observations.

Regarding the action space, each action consists of a set of values representing the HVAC heating and cooling setpoints. In the case of 5ZoneAutoDXVAV, an action $a_t \in \mathbb{R}^2$ involves the selection of two values representing the heating and cooling setpoints for the whole building. Conversely, for 2ZoneDataCenterHVAC, an action

[3] For an in-depth study of these variables, please refer to EnergyPlus official documentation: https://energyplus.net/documentation.

Table 4 Action variables for 5ZoneAutoDXVAV and 2ZoneDataCenterHVAC

Building	Variable name	Min.	Max.
5ZoneAutoDXVAV	Zone heating setpoint temperature	15	22.5
	Zone cooling setpoint temperature	22.5	30
2ZoneDataCenterHVAC	East zone heating setpoint temperature	15	22.5
	East zone cooling setpoint temperature	22.5	30
	West zone heating setpoint temperature	15	22.5
	West zone cooling setpoint temperature	22.5	30

$a_t \in \mathbb{R}^4$ consists of four values corresponding to the selected setpoints for the east and west zones of the facility. Their identifier, definition, and ranges are displayed in Table 4.

Finally, based on several tests, we consider a continuous action space, as opposed to a discrete one. This is because a discrete action space can overly constrain the agent's behaviour if the number of allowed actions is limited. In addition, the use of fixed setpoints can worsen the agent's flexibility by explicitly imposing expert knowledge. In our case, while expert knowledge is present in the range of values that these variables can take, there are no restrictions within this range, allowing each setpoint to take arbitrary continuous values—as long as the heating setpoint remains below the cooling setpoint. Finally, we must consider that a continuous action space covers the action range of a discrete one. This means that the possibility of choosing similar actions is not lost, while offering a wide margin for improvement.

4.4 Reward

For the reward calculation, we chose the linear reward function included by default in Sinergym:

$$r_t = -\omega \, \lambda_P \, P_t - (1 - \omega) \, \lambda_C \, (\mid T_t - T_{up} \mid_+ + \mid T_t - T_{low} \mid_+) \tag{6}$$

This is a similar formula to the one shown in Eq. 5, where ω is the weight assigned to power consumption and consequently, $1 - \omega$ the weight assigned to comfort; λ_P and λ_C are both scaling constants; P_t is the facility total HVAC power demand rate (measured in W); T_t is the current inner temperature (in °C), and T_{up} and T_{low} are the limits of the target comfort range. Thus, the greater the consumption, or distance from the interior of the comfort range, the worse the reward, and the more the agent is penalized.

The linear reward allows us to maintain a real trade-off between comfort and consumption. Thus, assuming precise scaling factors, the preference for each reward term is determined solely by the value of ω, and no term is more important than the other by default.

Recent studies such as (Kadamala et al. 2024) address the issue of reward selection without reaching a clear consensus on which option is more effective in practice, so we opt for the most equitable reward definition. Other alternatives may lead to a bias towards a particular reward term. For example, this is the case with Sinergym's exponential reward, where deviations from target temperatures have a greater impact than increases in power consumption.

For the majority of the experiments in Sect. 5, we use the default value of $\omega = 0.5$, which involves seeking a complete balance between comfort and power requirements. We also analyse the effect of using different weightings in Sect. 5.6.

4.5 Algorithms

We briefly summarize the DRL algorithms and rule-based controllers used throughout the experimentation. The proposed control problem is addressed from a single-agent perspective, since multi-agent HVAC control is mostly applied in building communities (Pinto et al. 2021; Vazquez-Canteli et al. 2020) or problems where the action space is too large for a single agent (Fu et al. 2022). For the environments considered in this work—with a maximum of two controlled zones in 2ZoneDataCenterHVAC, such control is not strictly required.

The DRL algorithms selected were PPO, TD3, and SAC. As it is beyond the scope of this paper to go into their theoretical details, we refer the reader to the original publications where these are presented in detail: Schulman et al. (2017a) (PPO), Fujimoto et al. (2018) (TD3), and Haarnoja et al. (2018) (SAC).

PPO is an improvement over the on-policy algorithm TRPO. On-policy algorithms have a single policy that improves progressively according to the exploration of states and/or actions. This algorithms are generally sample-inefficient due to the data loss that occurs when updating their policy. This may hinder their performance in domains such as HVAC control, where data collection is slow, as detailed in Biemann et al. (2021). This is an assumption that we will empirically test in our experiments.

On the other hand, we have off-policy algorithms such as TD3 and SAC, which are more sample-efficient and can therefore be expected to produce better results in the HVAC control problem. Off-policy algorithms are based on two policies: a behaviour policy, which is exploratory, and a target policy, which is the one actually used by the agent and is based on the knowledge gathered by the exploratory one. Thus, while TD3 is an improved version of DDPG commonly used in HVAC control (Li et al. 2019; Gao et al. 2020; Biemann et al. 2021; Fu and Zhang 2021), SAC offers a promising alternative in this domain, as shown by recent studies such as (Brandi et al. 2022; Yu et al. 2020; Biemann et al. 2021; Coraci et al. 2021).

Regarding the rule-based controllers used as baselines, the following approaches produced the best results for each building:

- For 5ZoneAutoDXVAV, a control based on static setpoints depending on the season of the year, based on the ASHRAE standard (ASHRAE 2004) for thermal comfort in dwellings. In this way, the static setpoints [26, 29] °C are set for the period from June to September, while setpoints [20, 23.5] °C are maintained for the rest of the year.
- In the case of 2ZoneDataCenterHVAC, an integral control based on degree-by-degree correction of the current setpoints, depending on whether the temperature of each zone is within the specified range. This range corresponds to ASHRAE standard (ASHRAE 2016) for recommended temperature ranges in data centers, which establishes [18, 27] °C as reference setpoint values in these facilities.

Although these RBCs are able to guarantee proper control in the proposed scenarios, it should be noted that (1) they are not easily scalable, as they have to be defined manually; (2) they are not able to consider wide optimization horizons, as the rules used are

essentially reactive; and (3) in more complex scenarios (e.g. larger number of variables to be controlled), their performance could be significantly compromised.

For an in-depth study on how these RBCs are implemented, we refer the reader to the corresponding SINERGYM https://github.com/ugr-sail/sinergym/blob/main/sinergym/utils/controllers.py module, where their source code can be consulted.

5 Experiments

The following subsections detail the different experiments conducted. The objective pursued was to thoroughly study (1) which algorithms are able to guarantee better thermal control; (2) which DRL algorithms offer higher robustness to weather conditions for which they have not been trained; (3) the application of sequential learning and its effectiveness to obtain better controllers, and (4) the comfort-consumption trade-off and its influence on the performance of the controllers.

Although the most significant results will be illustrated graphically, space limitations have led us to transfer part of the complementary graphic material to an external repository: https://github.com/ugr-sail/paper-drl_building, available for detailed analysis of simulation data, code, and additional charts.

5.1 Methodology

Figure 2 summarizes the experimentation phases followed in this work. First, the performance between RBCs and DRL algorithms were compared in the 5ZONEAUTODXVAV and 2ZONEDATACENTERHVAC buildings with three different types of weather conditions, that is, 6 different environments.

Following this comparison, further experiments were carried out under more complex settings. On the one hand, the best DRL agent obtained in the previous step was used to test how it adapted to different weather conditions than those used during its training. We refer to this experiment as the "robustness test", which allowed us to evaluate the agent's ability to generalise to situations not experienced during training.

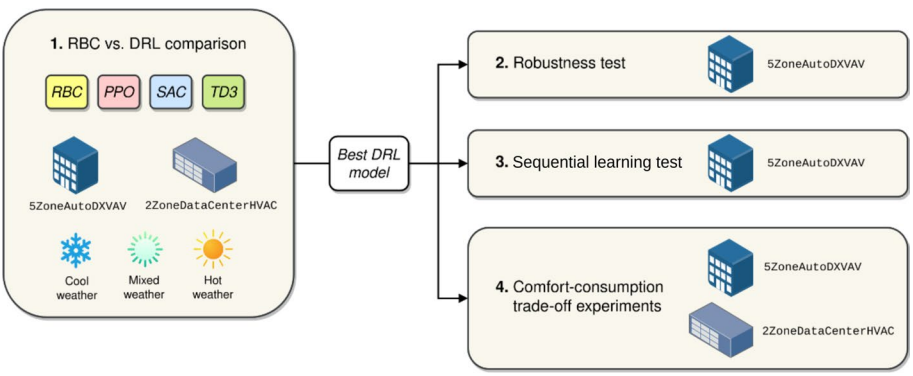


Fig. 2 Proposed experiments

Subsequently, we explored the application of sequential learning by progressively training an agent over multiple weather conditions, comparing its performance with that of an agent specialised in a single environment

Finally, we compared the performance of the best DRL controllers of each building under different definitions of the reward function, mainly by varying the weights of the comfort and consumption terms and observing the results obtained.

5.2 Experimental settings

Both the training and evaluation of the DRL algorithms were executed using Google Cloud computing platform and Sinergym version https://pypi.org/project/sinergym/1.8.2/. Regarding the handling of simulation time, each episode corresponds to 1 year of building simulation. Each episode consists of 35,040 time steps (i.e., 365 days × 24 h/day × 4 timesteps/hour), resulting in an observation sample and subsequent setpoint adjustment every 15 min. This is an appropriate value for the problem addressed, although it can be configured as desired within Sinergym settings.

In addition, the evaluation metrics common to all experiments and used to compare the agents are the following:

- Mean episode reward: calculated as the arithmetic mean of the rewards obtained in each time step of an episode.
- Mean power demand: mean HVAC electricity demand rate of the building's HVAC system (in W), provided by EnergyPlus.
- Comfort violation: percentage of time during which the ambient temperature is outside a desired comfort range, and mean value (in °C) of temperature deviations from comfort limits. In the case of 5ZoneAutoDXVAV, the comfort ranges used were: [23, 26] °C for June to September, and [20, 23.5] °C for the rest of the year. Meanwhile, for 2ZoneDataCenterHVAC, a single comfort range was used: [18, 27] °C, as recommended by ASHRAE's standard (ASHRAE 2016), in order to ensure the safety of the building's equipment.

Other metrics, such as the evolution of indoor and outdoor temperatures, will also give an insight into how the control adjusts between setpoints and how performance evolves, as detailed below.

5.3 Comparison between control algorithms

The first experiment conducted was a comparison between DRL algorithms and RBCs. This involve the training and subsequent evaluation of the DRL algorithms in different environments, in order to identify which algorithms perform better for each environment. The environments used for training and evaluation consisted in all the combinations between buildings (5ZoneAutoDXVAV, 2ZoneDataCenterHVAC) and weathers (hot dry, mixed humid, and cool marine), resulting in 6 different scenarios.

The configuration settings for this experiment were as follows:

- Machine used for training: Google Cloud's https://cloud.google.com/compute/docs/machine-types machine, equipped with a 2.2 GHz *Intel Cascade Lake* processor, 8 vCPU, and 8192 MiB of RAM.

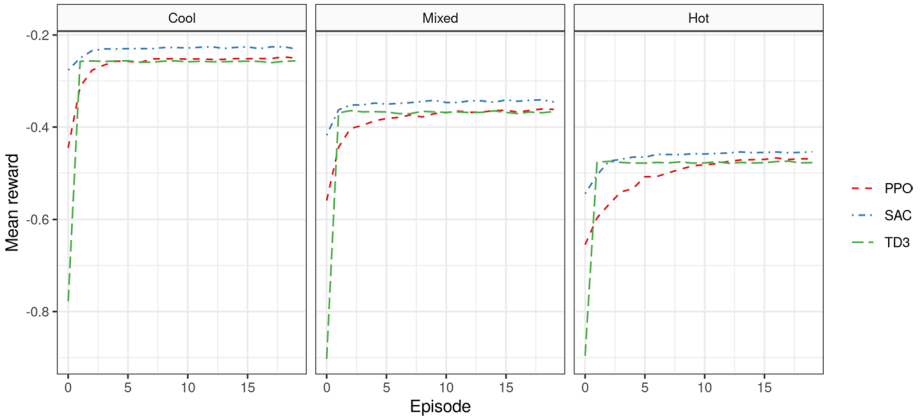


Fig. 3 Mean episode rewards during 20 training episodes in 5ZoneAutoDXVAV

- Number of training episodes: 20. This was found to be a sufficient number of episodes for all algorithms to achieve convergence.
- Frequency of evaluation: 4. Specifies the number of training episodes after which an evaluation and selection of the best model is performed.
- Evaluation length: 3. Refers to the number of episodes used to perform the evaluations.
- Random seed: 42. This seed was used in all experiments to facilitate the replication of the results.
- Normalization of the observations in [0, 1], using the Sinergym normalization wrapper.
- Multiple combinations of **hyperparameters** were tested until finding the ones that offered the best results for each DRL algorithm and environment combination, as detailed in Appendix 1. Endorsing the idea of (Brandi et al. 2020; Agarwal et al. 2021), the choice of hyperparameters in DRL is a complex task that greatly affects the training process of the agents. Therefore, it is necessary to follow a testing process during several episodes to compare the performance of different configurations. This philosophy was followed until acceptable sets of hyperparameters were found.

Regarding the training process, Fig. 3 shows how SAC offers the best results in all 5ZoneAutoDXVAV scenarios, with early convergence at its optimal values. Moreover, TD3 shows fast convergence, although without surpassing SAC, and offering similar results—or even slightly lower—than those of PPO, whose convergence requires a greater number of training episodes.

Regarding 2ZoneDataCenterHVAC, in Fig. 4 we can see a higher instability of TD3 in the first episodes, and a later better performance against SAC and PPO. In this environment, SAC again demonstrates early convergence, outperforming PPO in cool and hot climates, and equaling it in the mixed case.

Once the algorithms were trained and reached convergence, they were evaluated during 20 episodes using the best model obtained during training. Figures 5 and 6 show the results obtained for each environment, considering the metrics already presented in Sect. 5.2: mean rewards per episode, power consumption and comfort violation. We will analyze the results obtained taking as baseline the performance of both RBCs and random agents (RAND). The values represented in the boxplots are summarized in Appendix 2.

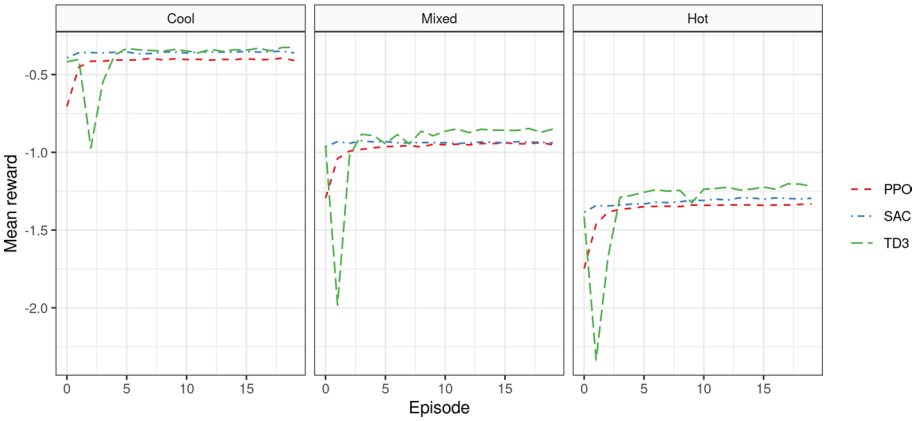


Fig. 4 Mean episode rewards during 20 training episodes in 2ZoneDataCenterHVAC

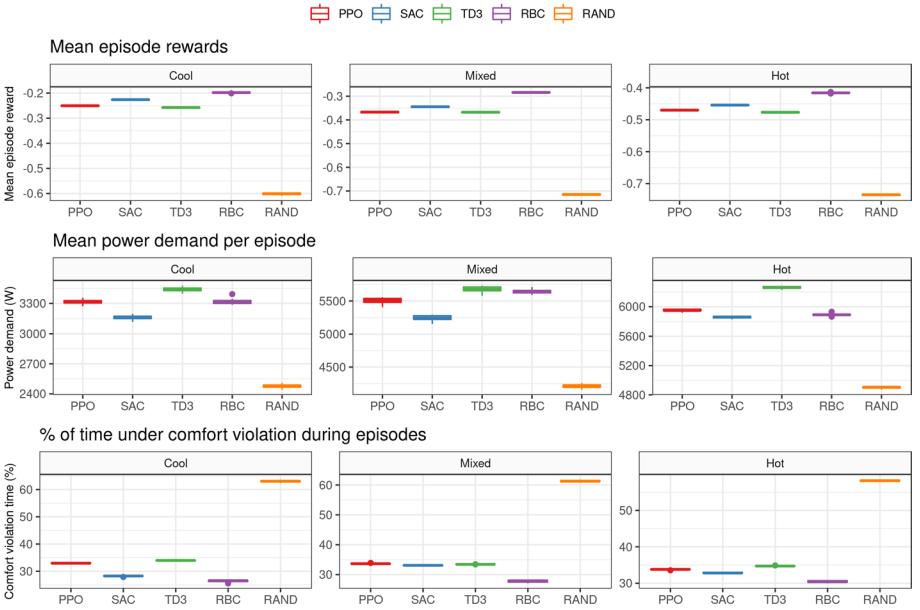


Fig. 5 Results for 20 evaluation episodes in 5ZoneAutoDXVAV

In the case of 5ZoneAutoDXVAV, as shown in Fig. 5, the best comfort-consumption balance is guaranteed by the RBC. Being this an ad hoc solution implemented for each building, we are interested in knowing which DRL algorithm comes closest to its performance. In this case, as we could see during training, SAC is the closest agent to RBC control in all climates, followed by PPO and TD3, with similar mean rewards in all scenarios. Thus, none of the agents manages to outperform RBC regarding this metric.

Considering the comfort and consumption values, we observe that RBC's main competitor, SAC, manages to reduce power demand without major penalties in comfort

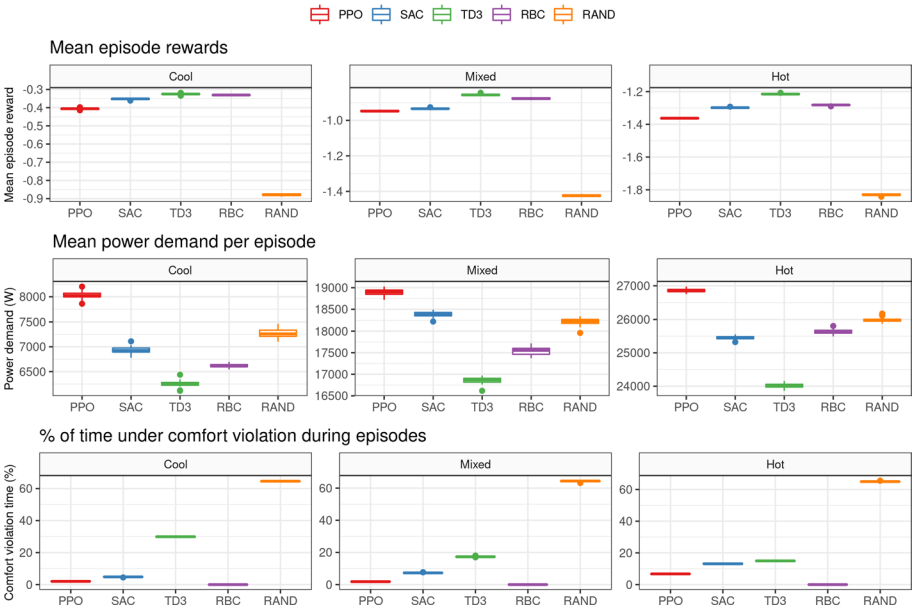


Fig. 6 Results for 20 evaluation episodes in 2ZoneDataCenterHVAC

violation, especially in hot weather. On the other hand, TD3 is the most energetically costly agent, while PPO maintains a significant balance between both metrics without demonstrating outstanding performance.

Looking at Fig. 6, corresponding to 2ZoneDataCenterHVAC, we find a higher competitiveness with respect to RBC, which is outperformed in all scenarios by TD3 in terms of reward. We see, in turn, significant power savings with respect to the other agents, albeit at the cost of a greater comfort violation. This could potentially be less interesting if we are dealing with a building where the temperature must always be kept in a specific range for safety reasons, so more conservative options such as SAC, PPO or RBC itself could be more suitable.

When we refer to conservative solutions, we mean those that offer indoor temperatures further away from the limits of comfort ranges. This makes it possible to deal with temperature inertia at the cost of investing more power in ensuring stable temperatures. For example, if we compare the indoor temperatures throughout the simulation for TD3 (Fig. 7) and RBC (Fig. 8), we observe that the former temperatures are closer to the upper comfort limit than those of the RBC. This behavior may be interpreted as the DRL agent trying to optimize consumption by taking a higher risk in ensuring comfort, thus leading to indoor temperatures closer to the limit allowed by the reward function.

A more feasible alternative in ensuring compliance with comfort limits might involve modifying the reward function used, giving greater importance to comfort versus consumption. This approach will be specifically addressed in Sect. 5.6.

Finally, it is worth noting the difficulties of the DRL algorithms in adapting to comfort ranges that vary at specific periods of the episode. This is the case of 5ZoneAutoDXVAV and the variations in the comfort ranges defined for the warm and cold months, as explained in Sect. 5.2. For instance, Fig. 9 shows how the sudden change

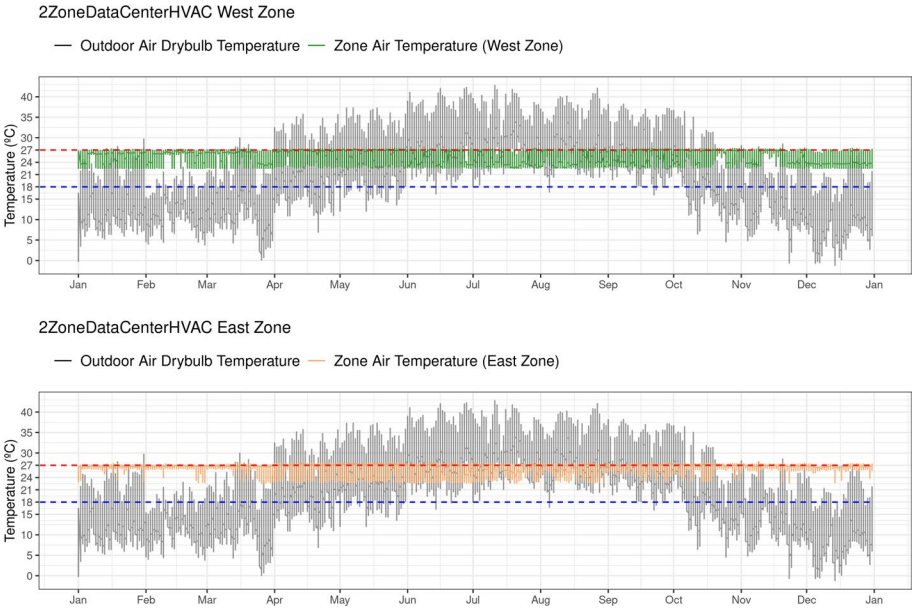


Fig. 7 Evolution of temperatures during a year of simulation using TD3 in 2ZoneDataCenterHVAC-hot. Comfort thresholds are marked with the horizontal dotted red and blue lines. (Color figure online)

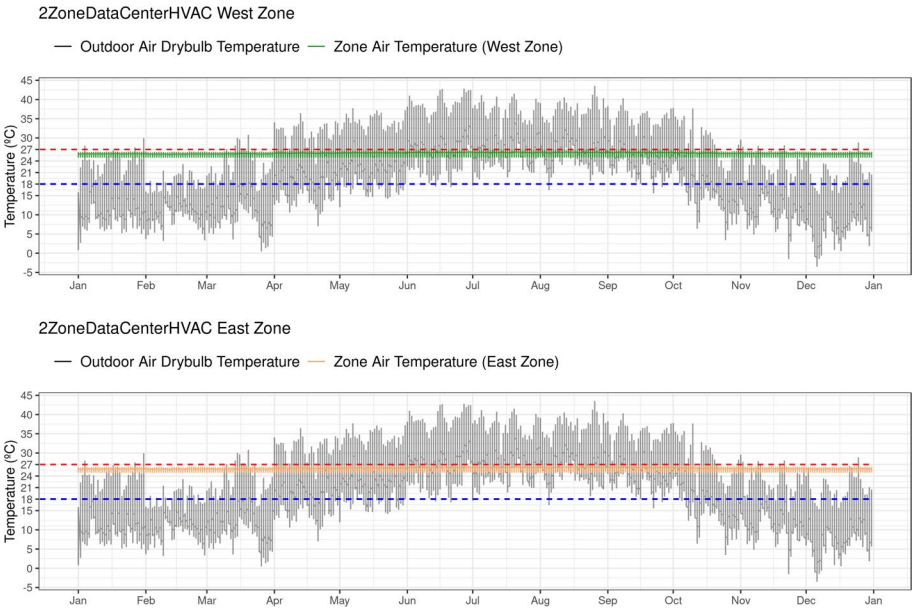


Fig. 8 Evolution of temperatures during a year of simulation using RBC in 2ZoneDataCenterHVAC-hot. Comfort thresholds are marked with the horizontal dotted red and blue lines. (Color figure online)

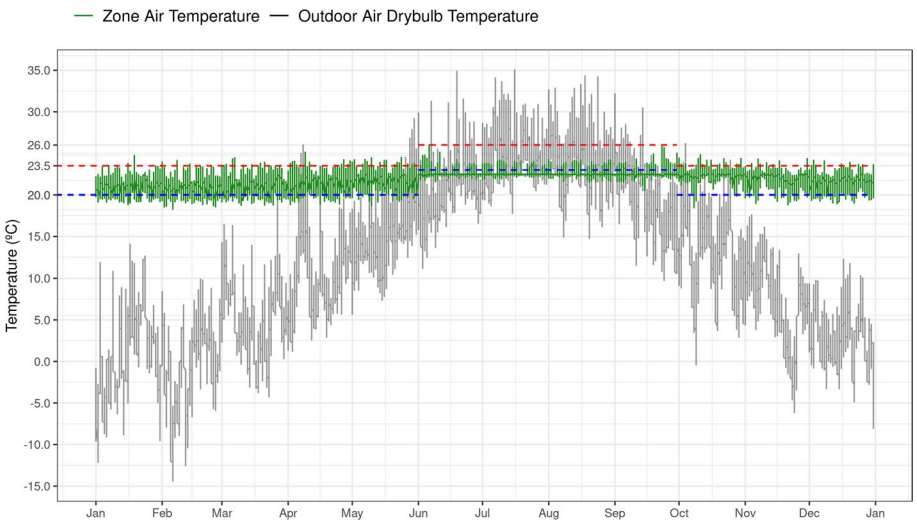


Fig. 9 Evolution of temperatures during a year of simulation using SAC in 5ZoneAutoDXVAV-mixed. Comfort thresholds are marked with the horizontal dotted red and blue lines. (Color figure online)

in the desired temperature range poses difficulties for SAC, which does not adapt well enough to the new comfort requirements.

Having provided a first insight into the performance of the different algorithms, in the following subsections we will address issues related to performance improvement and generalization capabilities of DRL agents.

5.4 Robustness test

We will test the performance of agents executed in environments that differ from those in which they have been trained. This will enable to know how far an agent is able to generalize and extrapolate the knowledge acquired in a given climate to be applied in a different one.

According to Fig. 10, we observe that the agents that perform best in each climate are those that have been trained in that same climate, as would be expected. After applying Mann–Whitney U tests, we obtain significant differences between the rewards obtained in each test environment ($p < 0.05$ in all cases).

In all cases, the following order of performance holds true: cool is the climate for which the greatest rewards are obtained, followed by mixed and hot. Even agents trained in mixed and hot get a higher reward in cool than in their respective training climates. The same happens for the agent trained in hot climate, whose reward is worst in its own climate than in the rest.

As a summary, Table 5 gathers the results obtained for each SAC training and evaluation combination in 5ZoneAutoDXVAV, which we will further discuss in Sect. 6.2.

5.5 Sequential learning

Another objective proposed in this work is to address the application of sequential learning: an approach that involves the progressive training of an agent under different weather conditions. On this basis, our goal in applying this incremental learning is to

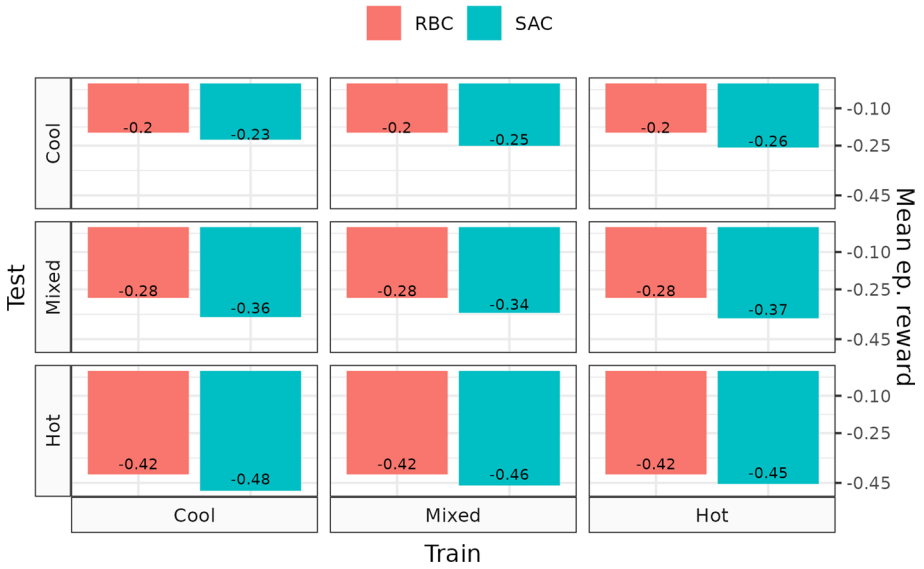


Fig. 10 Evaluation rewards for SAC agents trained in different weathers. Compared with RBC mean rewards in 5ZoneAutoDXVAV

Table 5 Rewards for every training and evaluation combination using SAC in 5ZoneAutoDXVAV

	Train		
	Cool	Mixed	Hot
Test			
Cool	Mean = −0.226 SD = 0.00129	Mean= −0.251 SD = 0.00129	Mean = −0.258 SD = 0.00135
Mixed	Mean = −0.361 SD = 0.00217	Mean = −0.344 SD = 0.00191	Mean = −0.366 SD = 0.00198
Hot	Mean = −0.482 SD = 0.0012	Mean = −0.46 SD = 0.00117	Mean = −0.454 SD = 0.00111

verify whether an agent trained progressively through different environments is capable of achieving better, or at least similar, performance to an agent trained in a single problem.

Thus, based on the results obtained in Sect. 5.3, we will test to compare the performance of an agent directly trained on a single climate with one trained on every available climate. We will use SAC under the same configurations described in the previous experiments. The sequence of training considered is: (1) cool, (2) mixed, and (3) hot.

Therefore, in Fig. 11 we observe that the performance of SAC trained under sequential learning is mostly inferior to that of standard SAC according to the metrics proposed.

It is noteworthy that the best performance obtained by SAC with sequential learning occurs in hot weather, which is actually the last weather used to train the agent. This could lead us to believe that the phenomenon of "catastrophic forgetting" (French 1999) might occur, as we will discuss in Sect. 6.3.

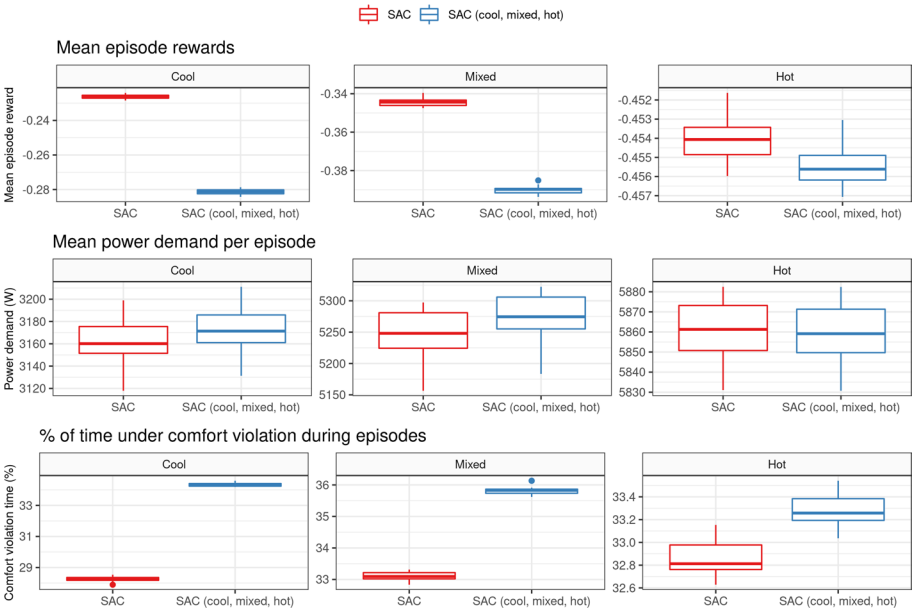


Fig. 11 Results for SAC and sequential learning SAC in 5ZoneAutoDXVAV

5.6 Comfort-consumption trade-off

As we already anticipated in the experimentation of Sect. 5.3, the reward function chosen, as well as the weights given to comfort and consumption in the training of the agents should influence their performance.

In this section, we will compare how different weights for comfort and consumption influence the performance of the agents under the same rest conditions. Specifically, building on the results described in Sect. 5.3, we will use the best performing agents on each building—SAC in 5ZoneAutoDXVAV, and TD3 in 2ZoneDataCenterHVAC—to compare the performance of the following agents:

- An agent trained only on the basis of the comfort requirements ($\omega = 1$ in Eq. 6), regardless of consumption.
- Another trained with a weight of 75% for consumption and 25% for comfort ($\omega = 0.25$).
- An agent trained with the same weight for consumption and comfort ($\omega = 0.5$).
- And, finally, an agent trained with a weight of 75% for comfort and 25% for consumption ($\omega = 0.75$).

Thus, the results obtained are shown in Figs. 12 and 13. From the figures we observe that a greater emphasis on comfort in the reward function reduces comfort violations, as would be expected. Improvements are especially significant from 25 to 50% comfort weight, and decrease as the comfort weight increases (although always improving).

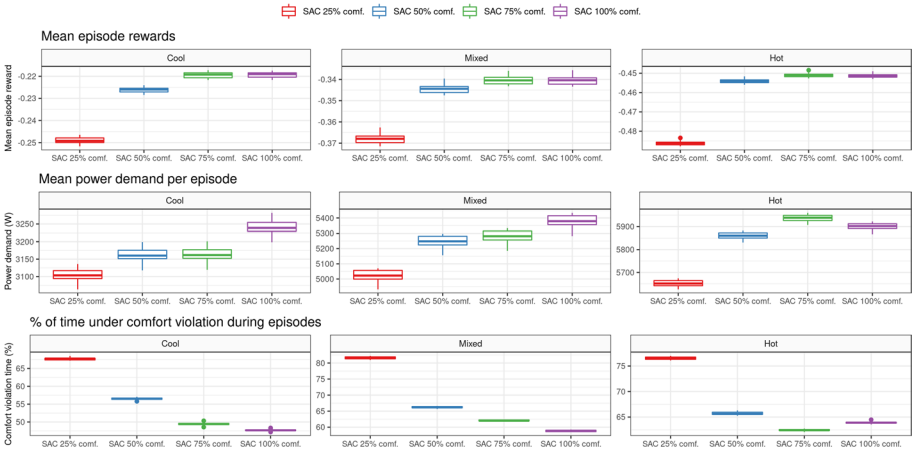


Fig. 12 Results for SAC with different comfort weights in 5ZoneAutoDXVAV

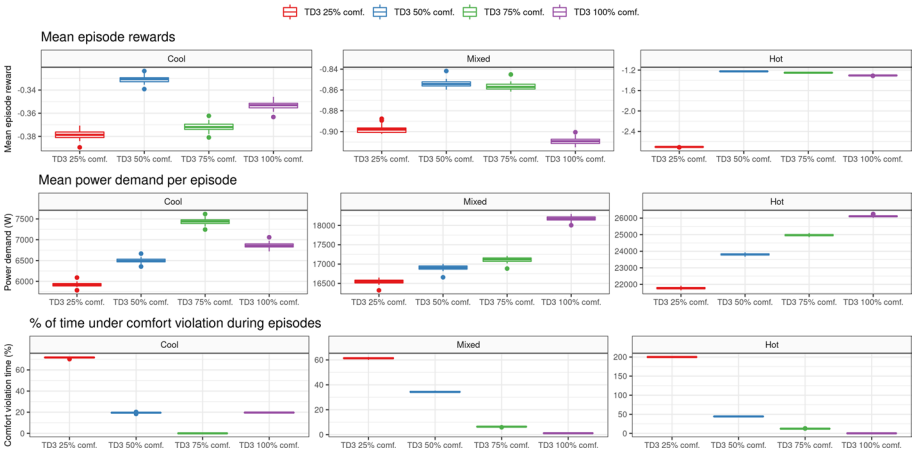


Fig. 13 Results for TD3 with different comfort weights in 2ZoneDataCenterHVAC

However, there are two exceptions where assigning 100% weight to comfort means some worsening from 75%. This is the case for SAC in hot climate and TD3 in cool climate. In Sect. 6.4 we will discuss possible causes for these exceptions.

Finally, improving comfort means—apart from the exceptions mentioned above—an increase in power demand. In this multi-objective problem, the trade-off is disrupted as soon as one objective is given greater importance than the other.

6 Discussion

In this section, we will discuss the results obtained after experimentation, not only describing them but also identifying gaps to be addressed in future work.

6.1 Rewards and comparison between algorithms

We begin by analyzing the results obtained by the different control algorithms in the set of proposed environments. Overall, the results obtained are consistently stable with minimal variation over the 20 evaluation episodes. Stable results from multiple evaluation episodes allows us to confirm that there are no random factors significantly affecting the agents' results (see Appendix 2).

As shown in Sect. 5.2, in the case of 5ZoneAutoDXVAV, no DRL algorithm was able to outperform RBC considering a reward which gives equal importance to comfort and consumption (50–50%). It should also be noted that the buildings used in this experiment have a relatively simple control, which can be easily translated into a limited set of rules. However, as we discussed in the Sect. 5, the results are particularly promising when we consider that there has been neither an exhaustive selection of observed variables nor an in-depth choice of hyperparameters.

In Fig. 9 we also observe how DRL algorithms have problems in adapting to changing setpoints in certain periods of the simulation, such as the warm months. If we also consider that the higher the temperature, the more difficult it is to guarantee the comfort-consumption balance (see rewards for hot weather environments in Fig. 5), the loss of performance is self-evident. This loss of performance can be caused by external factors that affect the agent's ability to act, such building characteristics or an increased power demand from the HVAC system for cooling.

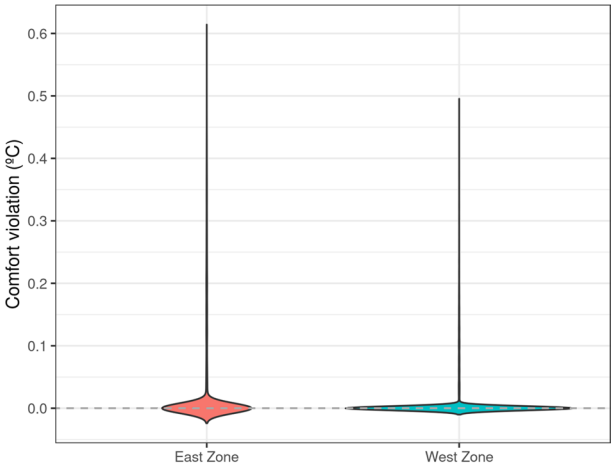
On the contrary, looking at the results in 2ZoneDataCenterHVAC, TD3 manages to outperform RBC in terms of average reward. The fact that a DRL agent is able to outperform an RBC in 2ZoneDataCenterHVAC, but not in 5ZoneAutoDXVAV could be justified by the greater complexity of the control problem, as the latter may require a more sophisticated strategy, involving a larger number of control variables.

As we anticipated earlier, the main reason why TD3 gets a better reward is because of the large reduction in consumption compared to RBC. However, this seems to imply a higher comfort violation, which is ultimately profitable for the agent.

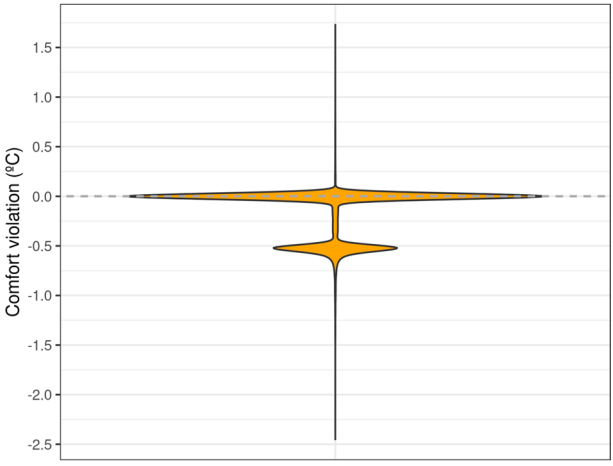
An important aspect to question at this point is under what conditions we consider a comfort violation to occur. If we look at Fig. 7, we can see that although many comfort violations occur for TD3 throughout the year, their value is quite low, mainly due to temperature inertia combined with the fact that power savings imply approaching comfort limits. Therefore, if we only look at Fig. 6, agnosticism towards real temperatures and deviations could lead us to believe that TD3 significantly sacrifices comfort in exchange for reducing consumption, thus reaching critical temperatures, which is not the case. As a proof, Fig. 14a shows the temperature violations achieved by TD3 with respect to the comfort ranges in 2ZoneDataCenterHVAC, proving that the temperature deviations exceed the upper comfort limit by less than 0.7 °C, and never the lower limit.

Something similar occurs in the case of the SAC in 5ZoneAutoDXVAV. Looking at Fig. 14b, we observe that the comfort violations above the upper limit reach a maximum of about 1.6 °C, while the comfort violations below the lower limit reach about 2.4 °C. In the latter case, the comfort violations are higher because the agent adapts with difficulty to the change of the comfort range during summer months, usually staying below the desired minimum temperature.

Finally, we should not overlook a common factor related to the reward function used for both buildings: the value of the scaling factors λ_P and λ_T (see Eq. 5), which were chosen empirically. This may lead to slight inaccuracies in the reward calculation.



(a) TD3 temperature deviations in 2ZoneDataCenterHVAC (hot weather)



(b) SAC temperature deviations in 5ZoneAutoDXVAV (mixed weather)

Fig. 14 Comfort violations by TD3 and SAC in different scenarios

Nevertheless, assuming the same values for all algorithms allows comparing the results under similar conditions.

6.2 Robustness and generalization

The results of the robustness and generalisation experiments yield some interesting insights. On the one hand, we confirm that the agents that performed best in a given environment were those that had been trained in the same environment. That is to say, an agent executed in an environment different from the one in which it was trained reported a loss of performance in terms of average reward.

On the other hand, the rewards obtained in each test environment (see Fig. 10) show that the climate associated with the highest reward was cool, followed by mixed and hot. However, these values should be interpreted with caution: the fact that the rewards obtained in the hot climate were lower for each agent than in the cool climate does not necessarily mean that the agents are unable to perform properly in hot environments, but may be due to external factors. For example, a reward of around −0.4 in hot weather could be comparable to one of around −0.2 in cool weather simply because the HVAC system consumes less when heating than when cooling, which is beyond the agents' ability to act. This leads us to suggest that a column-by-column study of Fig. 10 may not be as informative as a row-by-row comparison of the rewards.

These results also lead us to question whether a specific and fixed climate is the best approach when training DRL algorithms in this domain. Future work should try to clarify whether the differences in the control exerted by agents trained in different climates are really significant—we are currently dealing with centesimal variations in the value of the mean rewards, as well as to address alternatives to training based on single climates. For example, an alternative may involve using a dataset which pools records from several different climates, and from which one climate observation is randomly sampled at each time step. This is an idea already addressed in other works, such as Du et al. (2021), where a DRL agent was trained by sampling observations of heating and cooling scenarios, thus leading to more adaptive action strategies. Consequently, this solution may greatly enhance agents' generalization during training.

6.3 Effectiveness of sequential learning

The search for an improvement in training by applying sequential learning also raises important issues in its application and results. As shown in Fig. 11, this approach resulted in a performance degradation with respect to standard SAC in each of the climates. Thus, the proposed method did not lead to improvements in expected performance.

A noticeable finding is that the results obtained report a lower loss of performance by the agent trained with sequential learning in the hot climate. This leads us to question whether this is caused by the fact that it was the last climate used for training. If so, we would be facing the already mentioned problem of catastrophic forgetting, which implies a bias in the agent's learning towards the most recent training data, discarding the learning of previous climates.

In fact, if we compare SAC trained with sequential learning with the rest of SAC agents trained in single climates (Fig. 15), we notice that the rewards are always worse, with the exception of hot, where results are close to those of SAC trained in mixed, and superior to those trained in cool. Again, this improved control in hot climate could be a direct consequence of catastrophic forgetting, as it concerns the last climate used in the agent's training, thus partially forgetting what the agent learned for cool and mixed environments.

To evaluate whether this phenomenon occurs, we tested the performance of an agent trained with sequential learning by following the reverse process. To do so, a SAC agent was firstly trained in hot weather, then in mixed, and finally in cool. Its performance is shown and compared with the rest of the SAC agents in Fig. 16, where we can observe a significant improvement in the reward for cool weather, at the expense of worsening performance for the rest of climates.

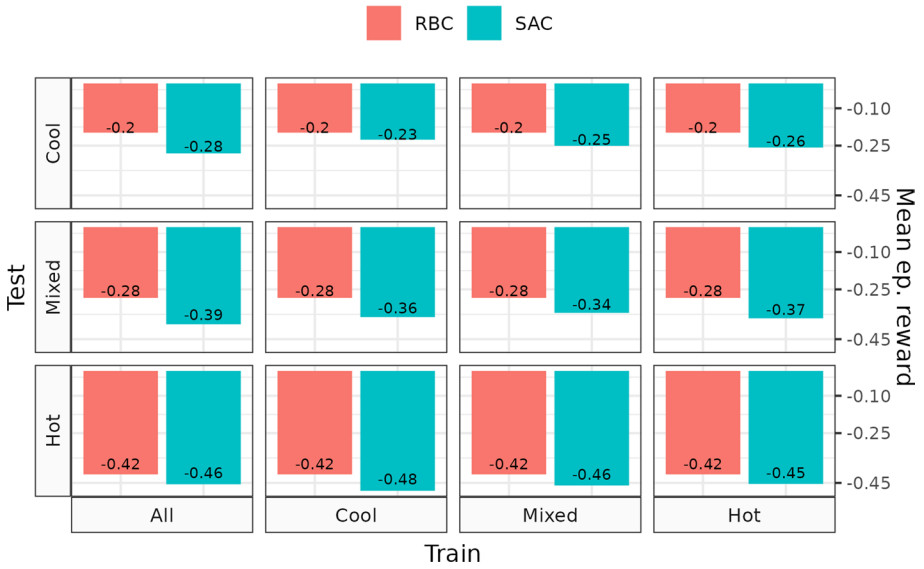


Fig. 15 Evaluation rewards for SAC trained with sequential learning in all climates, and SAC trained in single climates. Compared with RBC mean rewards in 5ZoneAutoDXVAV

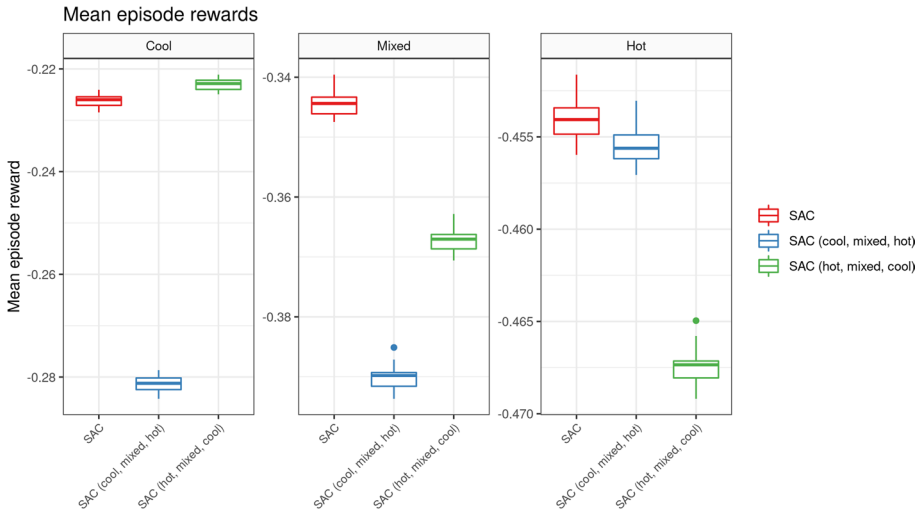


Fig. 16 Comparison between standard SAC rewards, and SAC agents trained under sequential learning in different orders

Looking at the results reinforces the idea that there is a bias towards the latter training climate—in this case, cool. This leads us to believe that catastrophic forgetting does indeed occur, preventing diversified training.

It is also noteworthy how the new agent trained in all three climates manages to outperform a SAC agent trained only in cold, so the application of sequential learning may

not be worthwhile in these circumstances. However, as a consequence of the existence of catastrophic forgetting, future work should address the influence of the architecture of the neural networks used by the agents, as well as the way in which the training data is handled. Again, we reiterate the idea that sampling from a set of observations from different climates may be a suitable option for this type of problems.

6.4 Comfort-consumption trade-off

From the results shown in Figs. 12 and 13, we can see how the weight assigned to comfort and, therefore, to consumption—directly influences the importance of this value in the training and final performance of the algorithms.

As we already anticipated, a higher comfort assurance implies, in most cases, a consequent increase in power demand. However, there are interesting cases such as SAC in 5ZoneAutoDXVAV and cold weather, where it is possible to reduce the comfort violation by increasing its importance in the reward function without significant drawbacks in terms of consumption.

Furthermore, the choice of weighting for comfort and consumption depends entirely on the problem and the environment we are dealing with. However, there are cases, such as the one mentioned above, where we can achieve significant improvements in comfort with hardly any penalty in consumption. It is therefore a problem that needs to be approached cautiously and, if possible, several options should be evaluated in a simulated way.

If we look at the rewards obtained by the different SAC agents in 5ZoneAutoDXVAV, we observe that a greater emphasis on comfort generally leads to better overall performance. As we have observed, this is because improvements in comfort do not have a large impact on power consumption, which offsets the results.

Nevertheless, in the case of TD3 in 2ZoneDataCenterHVAC, the comfort and consumption weighting changes do pose more of a challenge in terms of trade-off, with 50–50% being the best weighting choice in all climates.

Let us now study what happens with the two exceptions mentioned in Sect. 5.6: SAC in 5ZoneAutoDXVAV and hot weather, and TD3 in 2ZoneDataCenterHVAC with cool weather, where a 75% weight for comfort in the reward function gives better

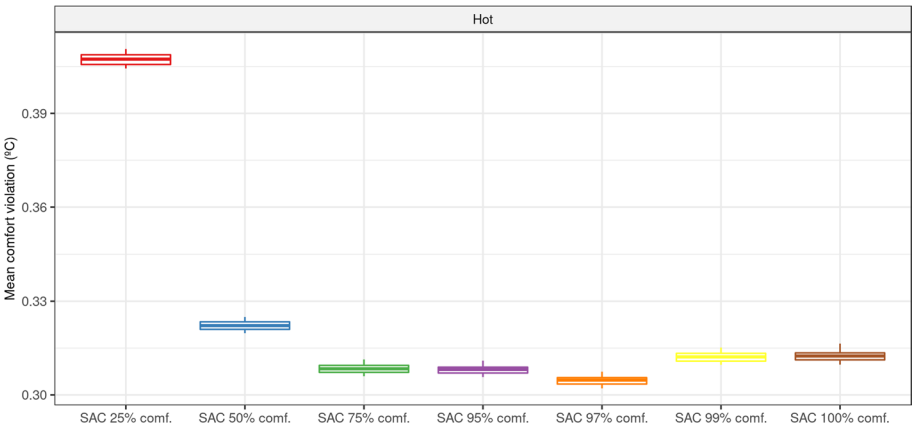


Fig. 17 Mean episode comfort violations for SAC with additional comfort weights in 5ZoneAutoDXVAV

results in terms of comfort violations than assigning a 100% percent weight to this value.

Taking the SAC case as an example, if we look at Fig. 17, we observe that adding intermediate weights (95%, 97% and 99%) to comfort between those previously considered (75% and 100%), leads to a range of similar results with centesimal variations. Therefore, the average comfort violations per episode reveal that there is some convergence in agents' performance, and that, despite increasing the importance of comfort in the reward function, their capacity is limited and it is not possible for them to improve comfort assurance. This is also the case for TD3 in 2ZoneDataCenterHVAC with cool weather, for which similar experiments were conducted.

Finally, we conclude that these two exceptions are indeed attainable cases which do not denote a noticeable loss of performance, but rather an oscillation around a limit that the agent is physically unable to overcome.

6.5 Summary and implications

We conclude with an overview of the results obtained and their potential implications for researchers and practitioners.

The results presented in Sects. 5.3 and 6.1 show how DRL-based agents match the performance of RBCs in simple scenarios, such as 5ZoneAutoDXVAV—where there is little room for improvement and how they perform better in environments of increasing complexity, such as 2ZoneDataCenterHVAC. We therefore recognise that the margin for improvement of a DRL over a reactive controller should be examined in the light of the complexity of the building being controlled. This complexity can be given by the system to be controlled, including its number of zones, as well as the observed and control variables. Other aspects may include climate variability or external factors such as occupancy or scheduling, which greatly affect the scalability and complexity of RBCs.

DRL agents were able to level reactive controllers in simple environments and surpass them in complex environments. Therefore, we consider that the complexity of the building being controlled is a major factor to be taken into account in the selection of controllers.

The robustness of agents deployed in environments with a climate different from that in which they were trained is another relevant issue (see Sects. 5.4 and 6.2). Although there were yield losses, we believe that it will depend on the case to consider whether or not these losses are negligible.

On the other hand, we observed in Sects. 5.5 and 6.3 that a sequential learning approach did not show improvements, but biases towards the last environment used to train the agent. While it may be due to overfitting, the usefulness of this type of training needs to be further studied.

Deploying a controller in an environment other than the one used in its training resulted in performance losses that might be acceptable depending on the case study. These losses were higher, and possibly impractical, in agents trained progressively through different climates, showing biases towards the last training environment employed.

One of the main advantages that DRL controllers offer compared to reactive control techniques is their flexibility in customizing the comfort-consumption trade-off by simply altering their reward function. In Sects. 5.6 and 6.4, we observed where are the limits of the comfort/consumption strategy that an agent can learn.

Moreover, while only these two factors have been considered in this paper, there are interesting studies, such as (Brandi et al. 2022), where the estimation of power consumption and prices are taken into account, considering a DRL agent adapted to sporadic peak consumption and stationary events. This reveals the customization capabilities of DRL agents in building environments without a complex model of each controller or an extensive rule base.

DRL agents are flexible to balance comfort and consumption preferences, and can incorporate other criteria by just adding extra terms in their reward.

Facing the use of DRL controllers in real environments, we must consider that the variables available in the simulation are also available in the deployment environment. Most of the variables used in this study could be easily monitored in a real environment, or replaced by proxies if not available (e.g. comfort metrics).

Purely online deployment and training of these DRL agents in real environments is not recommended since it requires long periods of time to achieve a minimally efficient behavior policy. On the contrary, the approach usually followed is to train the agents in simulated environments, avoiding cold starts, and then deploy these agents in real environments and continue to receive feedback from the information collected online.

In order to enable the deployment of a DRL agent trained in a simulated environment, it is necessary for the agent to have access, either directly or indirectly, to similar variables to those used in its training. The DRL setup facilitates continuous learning and adaptation after deployment.

7 Conclusions and future work

In this paper, we have addressed the performance comparison of multiple DRL algorithms in building HVAC control. Using the Sinergym framework, these algorithms were tested in various buildings and climates, considering both the search for comfort-consumption balance and scenarios in which one objective prevailed over the other. The robustness of the agents in different climates was also evaluated, as well as the potential benefits of progressive training to improve agents' performance.

In the search for proper comfort-consumption balance, SAC and TD3 were able to perform similar control to reactive controllers in most scenarios, even improving power savings at the cost of low comfort penalties. Since the agents were trained with completely raw observations—neither exhaustive variable selection nor preprocessing beyond normalization—and without extensive hyperparametrization, an immediate future work may involve the detailed study of which variables should compose agents' observations, as well as a deeper analysis to define which hyperparameters best fit each building-climate-algorithm combination. In any case, a relevant highlight derived from these results is the fact that positive results can be obtained for arbitrary environments without the need to significantly fine-tune observations and hyperparameters.

Regarding the robustness and generalisability of DRL agents, it was found that modifying the evaluation climate of an agent with respect to the one used in its training negatively influences its performance. Interestingly enough, these are minor losses that may be generally acceptable, avoiding the need to use a wide variety of climates in the training of the agents. Thus, a proposal for future work will involve training by sampling a pool of observations from a wide variety of climates, thus increasing the generalization capabilities of the agents and reducing training time and efforts.

Moreover, the use of sequential learning for training was not successful in most cases, mainly due to the bias of the agents to specialize in the control of the most recently learned environments. This phenomenon of catastrophic forgetting motivates future research from different perspectives, such as exploring the influence of the network architecture of DRL agents on the occurrence of this phenomenon, or studying possible improvements along this progressive learning process (i.e. experience replay and regularization).

Finally, the study of the comfort-consumption trade-off, and its translation into the agents' reward function, offered some significant highlights. On the one hand, it was observed that there are cases where improvements in comfort have hardly any penalties in power consumption, which leads us to suggest that it cannot be assumed that increasing the importance of comfort in the reward function will always lead to a detriment in power demand. This is where reward engineering comes into play, so future research should be conducted to compare different rewards and their influence on the comfort-consumption trade-off. On the other hand, it was observed that there are comfort weights beyond which agents are physically incapable of improving temperature assurance in the desired ranges. Generally, as the comfort weight tended towards 100% in the reward function, we began to find some convergence in the comfort penalties without considerable detriments or improvements.

As a conclusion of this work, and in view of the state of the art, we encourage future contributions in the area to be tested under multiple environments and configurations. Frameworks like Sinergym can be very useful in this regard.

Addressing the problem of building energy optimization from the perspective of multi-agent systems is another interesting approach for which standardized comparisons are also necessary. These solutions offer several advantages in large buildings where coordination between multiple independent HVAC controllers is required (Nagarathinam et al. 2020; Yu et al. 2020).

From a wider perspective, it would also be interesting to extend these studies with configurations where the use of green energies is considered as a variable to be optimized, given the rise of smart grids and renewable energies. For this purpose, the continued use of common evaluation frameworks and methodologies will enable the comparison of results, the standardization of the field, and the joint progress towards the improvement of building energy optimization.

Appendix 1: hyperparameters

Tables 6, 7 and 8 show the hyperparameter combinations tested for each DRL algorithm. While for PPO and TD3 those marked in bold are the ones that provided the best performance, in the case of SAC all the combinations presented similar performance, so the default values of Stable Baselines3 were selected. We are aware of the complexity underlying the search and selection of hyperparameters in DRL, as there are many possible heuristics in the search for optimal values. However, this task is beyond the scope of this paper, and we will intend to address it in future research.

All of the combinations were run under the same random seed for a total of 20 training episodes, followed by 20 evaluation episodes, whose results were used to compare.

Regarding the architecture of the agents' networks, the default architectures provided by StableBaselines3 were used. These networks consist of two fully connected layers with:

- 64 units per layer for PPO.
- 256 units for SAC.
- 400 (first layer) and 300 (second layer) units for TD3.

Table 6 Hyperparameters tested for PPO. The combination in bold was the one used in the experiments

	Learning_rate	n_steps	Batch_size	n_epochs	Gamma	Gae_lambda
1	0.0003	2048	64	10	0.99	0.95
2	0.0003	2048	128	10	0.99	0.95
3	0.0003	4096	128	10	0.99	0.95
4	0.001	2048	64	10	0.99	0.95
5	0.01	2048	64	10	0.99	0.95
6	**0.001**	**4096**	**128**	**15**	**0.9**	**0.9**
7	0.0003	4096	64	15	0.99	0.95

Table 7 Hyperparameters tested for SAC. The combination in bold was the one used in the experiments

	Learning_rate	Buffer_size	Batch_size	Tau	Gamma
1	**0.0003**	**1,000,000**	**256**	**0.005**	**0.99**
2	0.003	1,000,000	256	0.005	0.99
3	0.0003	10,000	256	0.005	0.99
4	0.0003	1,000,000	128	0.005	0.99
5	0.0003	1,000,000	256	0.005	0.9
6	0.0003	1,000,000	256	0.05	0.99
7	0.0001	100,000	256	0.003	0.9

Table 8 Hyperparameters tested for TD3. The combination in bold was the one used in the experiments

	Learning_rate	Buffer_size	Batch_size	Tau	Gamma
1	**0.003**	**1,000,000**	**100**	**0.005**	**0.9**
2	0.003	1,000,000	128	0.005	0.9
3	0.003	100,000,000	100	0.005	0.99
4	0.001	1,000,000	100	0.004	0.9
5	0.003	10,000,000	100	0.005	0.8
6	0.001	100,000	128	0.005	0.8
7	0.003	100,000	100	0.005	0.8
8	0.003	1,000,000	256	0.005	0.8
9	0.003	1,000,000	512	0.005	0.99
10	0.003	1,000,000	128	0.005	0.7
11	**0.003**	**1,000,000**	**64**	**0.005**	**0.9**

Combination 11 performed best for all environments, with the exception of `datacenter-mixed`, where combination 1 provided better results

Appendix 2: evaluation results

Tables 9 and 10 contain the numerical results in terms of reward, power consumption, and comfort for the evaluated agents. The row names refer to:

Table 9 Mean values and standard deviations for evaluation metrics in 5ZoneAutoDXVAV

	Cool	Mixed	Hot
Rew.	PPO (mean = −0.250, SD = 0.001)	PPO (mean = −0.367, SD = 0.002)	PPO (mean = −0.470, SD = 0.003)
	SAC (mean = −0.226, SD = 0.001)	SAC (mean = −0.344, SD = 0.002)	SAC (mean = −0.454, SD = 0.001)
	TD3 (mean = −0.258, SD = 0.001)	TD3 (mean = −0.368, SD = 0.002)	TD3 (mean = −0.477, SD = 0.001)
	RBC (mean = −0.198, SD = 0.003)	RBC (mean = −0.284, SD = 0.001)	RBC (mean = −0.416, SD = 0.001)
	RAND (mean = −0.601, SD = 0.002)	RAND (mean = −0.715, SD = 0.002)	RAND (mean = −0.735, SD = 0.002)
Pow.	PPO (mean = 3315.004, SD = 22.956)	PPO (mean = 5506.539, SD = 38.463)	PPO (mean = 5949.604, SD = 15.845)
	SAC (mean = 3161.168, SD = 22.741)	SAC (mean = 5248.838, SD = 35.758)	SAC (mean = 5860.091, SD = 16.135)
	TD3 (mean = 3439.188, SD = 23.281)	TD3 (mean = 5681.867, SD = 38.989)	TD3 (mean = 6260.862, SD = 16.208)
	RBC (mean = 3317.391, SD = 24.349)	RBC (mean = 5642.405, SD = 32.619)	RBC (mean = 5893.126, SD = 13.720)
	RAND (mean = 2476.047, SD = 17.468)	RAND (mean = 4207.539, SD = 27.431)	RAND (mean = 4900.652, SD = 13.134)
Comf.	PPO (mean = 32.997, SD = 0.137)	PPO (mean = 33.661, SD = 0.109)	PPO (mean = 33.805, SD = 0.113)
	SAC (mean = 28.254, SD = 0.168)	SAC (mean = 33.101, SD = 0.138)	SAC (mean = 32.853, SD = 0.144)
	TD3 (mean = 33.972, SD = 0.066)	TD3 (mean = 33.429, SD = 0.005)	TD3 (mean = 34.716, SD = 0.085)
	RBC (mean = 26.416, SD = 0.379)	RBC (mean = 27.791, SD = 0.369)	RBC (mean = 30.441, SD = 0.256)
	RAND (mean = 63.044, SD = 0.295)	RAND (mean = 61.312, SD = 0.256)	RAND (mean = 58.166, SD = 0.195)

Table 10 Mean values and standard deviations for evaluation metrics in 2ZoneDataCenterHVAC

	Cool	Mixed	Hot
Rew.	PPO (mean = −0.406, SD = 0.004)	PPO (mean = −0.948, SD = 0.004)	PPO (mean = −1.363, SD = 0.003)
	SAC (mean = −0.352, SD = 0.004)	SAC (mean = −0.934, SD = 0.003)	SAC (mean = −1.299, SD = 0.003)
	TD3 (mean = −0.326, SD = 0.004)	TD3 (mean = −0.856, SD = 0.004)	TD3 (mean = −1.216, SD = 0.003)
	RBC (mean = −0.331, SD = 0.002)	RBC (mean = −0.877, SD = 0.005)	RBC (mean = −1.282, SD = 0.004)
	RAND (mean = −0.879, SD = 0.006)	RAND (mean = −1.424, SD = 0.005)	RAND (mean = −1.830, SD = 0.006)
Pow.	PPO (mean = 8036.334, SD = 78.246)	PPO (mean = 18893.892, SD = 74.454)	PPO (mean = 26861.237, SD = 63.086)
	SAC (mean = 6933.573, SD = 74.947)	SAC (mean = 18382.084, SD = 62.678)	SAC (mean = 25453.090, SD = 60.553)
	TD3 (mean = 6254.018, SD = 72.473)	TD3 (mean = 16854.495, SD = 79.335)	TD3 (mean = 24022.427, SD = 65.902)
	RBC (mean = 6620.088, SD = 38.331)	RBC (mean = 17547.968, SD = 97.998)	RBC (mean = 25638.081, SD = 73.229)
	RAND (mean = 7271.774, SD = 88.468)	RAND (mean = 18212.941, SD = 91.845)	RAND (mean = 25992.284, SD = 74.121)
Comf.	PPO (mean = 2.009, SD = 0.063)	PPO (mean = 1.819, SD = 0.100)	PPO (mean = 6.697, SD = 0.142)
	SAC (mean = 4.823, SD = 0.157)	SAC (mean = 7.303, SD = 0.143)	SAC (mean = 13.151, SD = 0.225)
	TD3 (mean = 29.883, SD = 0.267)	TD3 (mean = 17.344, SD = 0.235)	TD3 (mean = 14.910, SD = 0.248)
	RBC (mean = 0.003, SD = 0)	RBC (mean = 0.003, SD = 0)	RBC (mean = 0.003, SD = 0)
	RAND (mean = 64.582, SD = 0.213)	RAND (mean = 64.288, SD = 0.350)	RAND (mean = 65.026, SD = 0.234)

- Rew.: mean episode reward.
- Pow.: mean episode power consumption (W).
- Comf.: mean episode comfort violation time (%).

Author contributions Conceptualization, methodology, formal analysis and investigation: all authors; Data curation: A. Manjavacas; Software: A. Campoy-Nieves, J. Jiménez-Raboso, A. Manjavacas; Resources: M. Molina-Solana, J. Gómez-Romero. Writing—original draft preparation: A. Manjavacas; Writing—review and editing: Miguel Molina-Solana, Juan Gómez-Romero; Visualization: A. Manjavacas; Funding acquisition, project administration and supervision: M. Molina-Solana, J. Gómez-Romero.

Funding Funding for open access publishing: Universidad de Granada/CBUA. Acknowledgements to grants SINERGY (PID2021.125537NA.I00) funded by MICIU/AEI/10.13039/501100011033 and "ERDF—A way of making Europe", IA4TES (MIA.2021.M04.0008) funded by the Spanish Ministry of Economic Affairs and Digital Transformation (NextGenerationEU), and D3S (P21.00247), "SE2021 UGR IFMIF-DONES, A-TIC-244-UGR20" funded by ERDF/Junta de Andalucía.

Code and data availability The code and datasets used in this work are available in the following repository: https://github.com/ugr-sail/paper-drl_building.

Declarations

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Agarwal R, Schwarzer M, Castro PS, Courville A, Bellemare MG (2021) Deep reinforcement learning at the edge of the statistical precipice. In: Advances in neural information processing systems. https://doi.org/10.48550/arXiv.2108.13264

ASHRAE (2004) ASHRAE: ASHRAE 55-2004: thermal environmental conditions for human occupancy. ASHRAE

ASHRAE (2016) ASHRAE: ASHRAE TC9.9: data center power equipment thermal guidelines and best practices systems. ASHRAE

ASHRAE (2021) ASHRAE: guideline 36-2021: high performance sequences of operation for HVAC systems. ASHRAE

Azuatalam D, Lee W-L, de Nijs F, Liebman A (2020) Reinforcement learning for whole-building HVAC control and demand response. Energy AI. https://doi.org/10.1016/j.egyai.2020.100020

Barrett E, Linder S (2015) Autonomous HVAC control, a reinforcement learning approach. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 3–19. https://doi.org/10.1007/978-3-319-23461-8_1

Benth FE, Šaltytė-Benth J (2005) Stochastic modelling of temperature variations with a view towards weather derivatives. Appl Math Financ 12(1):53–85. https://doi.org/10.1080/1350486042000271638

Biemann M, Scheller F, Liu X, Huang L (2021) Experimental evaluation of model-free reinforcement learning algorithms for continuous HVAC control. Appl Energy. https://doi.org/10.1016/j.apenergy.2021.117164

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

Borase RP, Maghade D, Sondkar S, Pawar S (2021) A review of PID control, tuning methods and applications. Int J Dyn Control 9(2):818–827. https://doi.org/10.1007/s40435-020-00665-4

Brandi S, Piscitelli MS, Martellacci M, Capozzoli A (2020) Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings. Energy Build 224:110225. https://doi.org/10.1016/j.enbuild.2020.110225

Brandi S, Fiorentini M, Capozzoli A (2022) Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management. Autom Constr. https://doi.org/10.1016/j.autcon.2022.104128

Chen Y, Norford LK, Samuelson HW, Malkawi A (2018) Optimal control of HVAC and window systems for natural ventilation through reinforcement learning. Energy Build 169:195–205. https://doi.org/10.1016/j.enbuild.2018.03.051

Coraci D, Brandi S, Piscitelli MS, Capozzoli A (2021) Online implementation of a soft actor-critic agent to enhance indoor temperature control and energy efficiency in buildings. Energies 14(4):997. https://doi.org/10.3390/en14040997

Costanzo GT, Iacovella S, Ruelens F, Leurs T, Claessens BJ (2016) Experimental analysis of data-driven control for a building heating system. Sustain Energy Grids Netw 6:81–90. https://doi.org/10.1016/j.segan.2016.02.002

Deng X, Zhang Y, Qi H (2022) Towards optimal HVAC control in non-stationary building environments combining active change detection and deep reinforcement learning. Build Environ. https://doi.org/10.1016/j.buildenv.2021.108680

Ding X, Du W, Cerpa AE (2020) MB2C: model-based deep reinforcement learning for multi-zone building control. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. pp 50–59. https://doi.org/10.1145/3408308.3427986

Du Y, Li F, Munk J, Kurte K, Kotevska O, Amasyali K, Zandi H (2021) Multi-task deep reinforcement learning for intelligent multi-zone residential HVAC control. Electr Power Syst Res 192:106959

Duan Y, Chen X, Houthooft R, Schulman J, Abbeel P (2016) Benchmarking deep reinforcement learning for continuous control. https://doi.org/10.48550/arXiv.1604.06778

Efheij H, Albagul A, Albraiki NA (2019) Comparison of model predictive control and PID controller in real time process control system. In: 2019 19th international conference on sciences and techniques of automatic control and computer engineering (STA). IEEE, pp 64–69. https://doi.org/10.1109/STA.2019.8717271

Fazenda P, Veeramachaneni K, Lima P, O'Reilly U-M (2014) Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems. J Ambient Intell Smart Environ 6(6):675–690. https://doi.org/10.3233/AIS-140288

Findeis A, Kazhamiaka F, Jeen S, Keshav S (2022) Beobench: a toolkit for unified access to building simulations for reinforcement learning. In: Proceedings of the thirteenth ACM international conference on future energy systems. pp 374–382. https://doi.org/10.1145/3538637.3538866

French RM (1999) Catastrophic forgetting in connectionist networks. Trends Cogn Sci 3(4):128–135. https://doi.org/10.1016/S1364-6613(99)01294-2

Fu C, Zhang Y (2021) Research and application of predictive control method based on deep reinforcement learning for HVAC systems. IEEE Access 9:130845–130852. https://doi.org/10.1109/ACCESS.2021.3114161

Fu Q, Han Z, Chen J, Lu Y, Wu H, Wang Y (2022a) Applications of reinforcement learning for building energy efficiency control: a review. J Build Eng. https://doi.org/10.1016/j.jobe.2022.104165

Fu Q, Chen X, Ma S, Fang N, Xing B, Chen J (2022b) Optimal control method of HVAC based on multi-agent deep reinforcement learning. Energy Build 270:112284

Fujimoto S, Hoof H, Meger D (2018) Addressing function approximation error in actor-critic methods. In: International conference on machine learning. pp 1582–1591. https://doi.org/10.48550/arXiv.1802.09477

Gao G, Li J, Wen Y (2020) Deepcomfort: energy-efficient thermal comfort control in buildings via reinforcement learning. IEEE Internet Things J 7(9):8472–8484. https://doi.org/10.1109/JIOT.2020.2992117

Geng G, Geary G (1993) On performance and tuning of PID controllers in HVAC systems. In: Proceedings of IEEE international conference on control and applications. IEEE, pp 819–824 . https://doi.org/10.1109/CCA.1993.348229

Ghahramani A, Zhang K, Dutta K, Yang Z, Becerik-Gerber B (2016) Energy savings from temperature setpoints and deadband: quantifying the influence of building and system properties on savings. Appl Energy 165:930–942. https://doi.org/10.1016/j.apenergy.2015.12.115

Gholamzadehmir M, Del Pero C, Buffa S, Fedrizzi R et al (2020) Adaptive-predictive control strategy for HVAC systems in smart buildings: a review. Sustain Cities Soc 63:102480. https://doi.org/10.1016/j.scs.2020.102480

Gibney E et al (2016) Google AI algorithm masters ancient game of Go. Nature 529(7587):445–446. https://doi.org/10.1038/529445a

GlobalABC (2021) Global alliance for buildings and construction: global status report for buildings and construction: towards a zero-emission, efficient and resilient buildings and construction sector. GlobalABC. https://www.unep.org/resources/report/2021-global-status-report-buildings-and-construction

Gomez-Romero J, Fernandez-Basso CJ, Cambronero MV, Molina-Solana M, Campana JR, Ruiz MD, Martin-Bautista MJ (2019) A probabilistic algorithm for predictive control with full-complexity models in non-residential buildings. IEEE Access 7:38748–38765

Gupta S, Singal G, Garg D (2021) Deep reinforcement learning techniques in diversified domains: a survey. Arch Comput Methods Eng 28(7):4715–4754. https://doi.org/10.1007/s11831-021-09552-3

Haarnoja T, Zhou A, Abbeel P, Levine S (2018) Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. PMLR, pp 1861–1870. https://doi.org/10.48550/arXiv.1801.01290

Han M, May R, Zhang X, Wang X, Pan S, Yan D, Jin Y, Xu L (2019) A review of reinforcement learning methodologies for controlling occupant comfort in buildings. Sustain Cities Soc. https://doi.org/10.1016/j.scs.2019.101748

IEA (2021) International Energy Agency: tracking buildings. IEA. https://www.iea.org/reports/tracking-buildings-2021

Islam R, Henderson P, Gomrokchi M, Precup D (2017) Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. https://doi.org/10.48550/arXiv.1708.04133

Jiménez-Raboso J, Campoy-Nieves A, Manjavacas-Lucas A, Gómez-Romero J, Molina-Solana M (2021) 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. Association for Computing Machinery, New York, USA, pp 319–323. https://doi.org/10.1145/3486611.3488729

Kadamala K, Chambers D, Barrett E (2024) Enhancing HVAC control systems through transfer learning with deep reinforcement learning agents. Smart Energy 13:100131

Kümpel A, Stoffel P, Müller D (2021) Self-adjusting model predictive control for modular subsystems in HVAC systems. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/2042/1/012037

Leitao J, Gil P, Ribeiro B, Cardoso A (2020) A survey on home energy management. IEEE Access 8:5699–5722. https://doi.org/10.1109/ACCESS.2019.2963502

Li Y, Wen Y, Tao D, Guan K (2019) Transforming cooling optimization for green data center via deep reinforcement learning. IEEE Trans Cybern 50(5):2002–2013. https://doi.org/10.48550/arXiv.1709.05077

Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv Preprint. https://doi.org/10.48550/arXiv.1509.02971

Lissa P, Deane C, Schukat M, Seri F, Keane M, Barrett E (2021) Deep reinforcement learning for home energy management system control. Energy AI. https://doi.org/10.1016/j.egyai.2020.100043

Mason K, Grijalva S (2019) A review of reinforcement learning for autonomous building energy management. Comput Electr Eng 78:300–312. https://doi.org/10.1016/j.compeleceng.2019.07.019

Mawson VJ, Hughes BR (2021) Optimisation of HVAC control and manufacturing schedules for the reduction of peak energy demand in the manufacturing sector. Energy. https://doi.org/10.1016/j.energy.2021.120436

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing Atari with deep reinforcement learning. arXiv Preprint. https://doi.org/10.48550/arXiv.1312.5602

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533. https://doi.org/10.1038/nature14236

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. PMLR, pp 1928–1937. https://doi.org/10.48550/arXiv.1602.01783

Mocanu E, Mocanu DC, Nguyen PH, Liotta A, Webber ME, Gibescu M, Slootweg JG (2018) Online building energy optimization using deep reinforcement learning. IEEE Trans Smart Grid 10(4):3698–3708. https://doi.org/10.1109/TSG.2018.2834219

Morari M, Lee JH (1999) Model predictive control: past, present and future. Comput Chem Eng 23(4–5):667–682. https://doi.org/10.1016/S0098-1354(98)00301-9

Morinibu, T, Noda T, Shota T (2019) Application of deep reinforcement learning in residential preconditioning for radiation temperature. In: 2019 8th international congress on advanced applied informatics (IIAI-AAI). IEEE, pp 561–566. https://doi.org/10.1109/IIAI-AAI.2019.00120

Moriyama T, Magistris GD, Tatsubori M, Pham T-H, Munawar A, Tachibana R (2018) Reinforcement learning testbed for power-consumption optimization. In: Asian simulation conference. Springer, pp 45–59. https://doi.org/10.1007/978-981-13-2853-4_4

Mozer MC (1998) The neural network house: an environment hat adapts to its inhabitants. In: Proceedings of the AAAI spring symposium on intelligent environments, vol 58. pp 110–114

Nagarathinam S, Menon V, Vasan A, Sivasubramaniam A (2020) Marco-multi-agent reinforcement learning based control of building HVAC systems. In: Proceedings of the eleventh ACM international conference on future energy systems. pp 57–67

Perera A, Kamalaruban P (2021) Applications of reinforcement learning in energy systems. Renew Sustain Energy Rev. https://doi.org/10.1016/j.rser.2020.110618

Pérez-Lombard L, Ortiz J, Pout C (2008) A review on buildings energy consumption information. Energy Build 40(3):394–398. https://doi.org/10.1016/j.enbuild.2007.03.007

Pinto G, Piscitelli MS, Vázquez-Canteli JR, Nagy Z, Capozzoli A (2021) Coordinated energy management for a cluster of buildings through deep reinforcement learning. Energy 229:120725

Privara S, Cigler J, Váňa Z, Oldewurtel F, Sagerschnig C, Žáčeková E (2013) Building modeling as a crucial part for building predictive control. Energy Build 56:8–22

Rajasekhar B, Tushar W, Lork C, Zhou Y, Yuen C, Pindoriya NM, Wood KL (2020) A survey of computational intelligence techniques for air-conditioners energy management. IEEE Trans Emerg Top Comput Intell 4(4):555–570. https://doi.org/10.1109/TETCI.2020.2991728

Raman NS, Devraj AM, Barooah P, Meyn SP (2020) Reinforcement learning for control of building HVAC systems. In: 2020 American control conference (ACC). IEEE, pp 2326–2332. https://doi.org/10.23919/ACC45564.2020.9147629

Sakuma Y, Nishi H (2020) Airflow direction control of air conditioners using deep reinforcement learning. In: 2020 SICE international symposium on control systems (SICE ISCS). IEEE, pp 61–68. https://doi.org/10.23919/SICEISCS48470.2020.9083565

Salakij S, Yu N, Paolucci S, Antsaklis P (2016) Model-based predictive control for building energy management. I: energy modeling and optimal control. Energy Build 133:345–358. https://doi.org/10.1016/j.enbuild.2016.09.044

Salsbury TI (2005) A survey of control technologies in the building automation industry. IFAC Proc Vol 38(1):90–100. https://doi.org/10.3182/20050703-6-CZ-1902.01397

Scharnhorst P, Schubnel B, Fernández Bandera C, Salom J, Taddeo P, Boegli M, Gorecki T, Stauffer Y, Peppas A, Politi C (2021) Energym: a building model library for controller benchmarking. Appl Sci. https://doi.org/10.3390/app11083518

Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017a) Proximal policy optimization algorithms. arXiv Preprint. https://doi.org/10.48550/arXiv.1707.06347

Schulman J, Levine S, Moritz P, Jordan MI, Abbeel P (2017b) Trust region policy optimization. https://doi.org/10.48550/arXiv.1502.05477

Serale G, Fiorentini M, Capozzoli A, Bernardini D, Bemporad A (2018) Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: problem formulation, applications and opportunities. Energies 11(3):631. https://doi.org/10.3390/en11030631

Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge. https://doi.org/10.1109/tnn.1998.712192

Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. J Mach Learn Res. https://doi.org/10.5555/1577069.1755839

Torrey L, Shavlik J (2010) Transfer learning. Handbook of research on machine learning applications and trends: algorithms. Methods, and techniques. IGI Global, Hershey, pp 242–264

U.S. Department of Energy: Prototype Building Models | Building Energy Codes Program (2021). https://www.energycodes.gov/prototype-building-models#Weather Accessed 29 Mar 2022

Valladares W, Galindo M, Gutiérrez J, Wu W-C, Liao K-K, Liao J-C, Lu K-C, Wang C-C (2019) Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm. Build Environ 155:105–117. https://doi.org/10.1016/j.buildenv.2019.03.038

Vázquez-Canteli JR, Nagy Z (2019) Reinforcement learning for demand response: a review of algorithms and modeling techniques. Appl Energy 235:1072–1089. https://doi.org/10.1016/j.apenergy.2018.11.002

Vázquez-Canteli JR, Ulyanin S, Kämpf J, Nagy Z (2019) Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities. Sustain Cities Soc 45:243–257. https://doi.org/10.1016/j.scs.2018.11.021

Vazquez-Canteli JR, Henze G, Nagy Z (2020) MARLISA: multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. pp 170–179

Wang Z, Hong T (2020) Reinforcement learning for building controls: the opportunities and challenges. Appl Energy. https://doi.org/10.1016/j.apenergy.2020.115036

Wang Y, Velswamy K, Huang B (2017) A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. Processes. https://doi.org/10.3390/pr5030046

Wang C, Pattawi K, Lee H (2020) Energy saving impact of occupancy-driven thermostat for residential buildings. Energy Build. https://doi.org/10.1016/j.enbuild.2020.109791

Watkins CJ, Dayan P (1992) Q-learning. Mach Learn 8(3–4):279–292

Wei T, Wang Y, Zhu Q (2017) Deep reinforcement learning for building HVAC control. In: Proceedings of the 54th annual design automation conference 2017. pp 1–6. https://doi.org/10.1145/3061639.3062224

Wölfle D, Vishwanath A, Schmeck H (2020) A guide for the design of benchmark environments for building energy optimization. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. pp 220–229 . https://doi.org/10.1145/3408308.3427614

Xu S, Wang Y, Wang Y, O'Neill Z, Zhu Q (2020) One for many: transfer learning for building HVAC control. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. pp 230–239. https://doi.org/10.1145/3408308.3427617

Yang T, Zhao L, Li W, Zomaya AY (2020) Reinforcement learning in sustainable energy and electric systems: a survey. Annu Rev Control 49:145–163. https://doi.org/10.1016/j.arcontrol.2020.03.001

Yang Y, Srinivasan S, Hu G, Spanos CJ (2021) Distributed control of multizone HVAC systems considering indoor air quality. IEEE Trans Control Syst Technol 29(6):2586–2597. https://doi.org/10.1109/TCST.2020.3047407

Yao Y, Shekhar DK (2021) State of the art review on model predictive control (MPC) in heating ventilation and air-conditioning (HVAC) field. Build Environ. https://doi.org/10.1016/j.buildenv.2021.107952

Yoon YR, Moon HJ (2019) Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling. Energy Build. https://doi.org/10.1016/j.enbuild.2019.109420

Yu L, Sun Y, Xu Z, Shen C, Yue D, Jiang T, Guan X (2020) Multi-agent deep reinforcement learning for HVAC control in commercial buildings. IEEE Trans Smart Grid 12(1):407–419. https://doi.org/10.1109/TSG.2020.3011739

Yu L, Qin S, Zhang M, Shen C, Jiang T, Guan X (2021) A review of deep reinforcement learning for smart building energy management. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3078462

Zai A, Brown B (2020) Deep reinforcement learning in action. Manning Publications, Shelter Island

Zhang H, Yu T (2020) Taxonomy of reinforcement learning algorithms. In: Deep reinforcement learning: fundamentals, research and applications. Springer, Singapore, pp 125–133 (2020). https://doi.org/10.1007/978-981-15-4095-0_3

Zhang D, Han X, Deng C (2018) Review on the research and practice of deep learning and reinforcement learning in smart grids. CSEE J Power Energy Syst 4(3):362–370. https://doi.org/10.17775/CSEEJPES.2018.00520

Zhang Z, Chong A, Pan Y, Zhang C, Lam KP (2019a) Whole building energy model for HVAC optimal control: a practical framework based on deep reinforcement learning. Energy Build 199:472–490. https://doi.org/10.1016/j.enbuild.2019.07.029

Zhang C, Kuppannagari SR, Kannan R, Prasanna VK (2019b) Building HVAC scheduling using reinforcement learning via neural network based model approximation. In: Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation. pp 287–296. https://doi.org/10.1145/3360322.3360861

Zou Z, Yu X, Ergan S (2020) Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. Build Environ. https://doi.org/10.1016/j.buildenv.2019.106535

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.