

Online implementation of a soft actor-critic agent to enhance indoor temperature control and energy efficiency in buildings

Original

Online implementation of a soft actor-critic agent to enhance indoor temperature control and energy efficiency in buildings / Coraci, D.; Brandi, S.; Piscitelli, M. S.; Capozzoli, A.. - In: ENERGIES. - ISSN 1996-1073. - 14:4(2021), p. 997. [10.3390/en14040997]

Availability:

This version is available at: 11583/2876731 since: 2021-03-25T11:55:15Z

Publisher:

MDPI AG

Published

DOI:10.3390/en14040997

Terms of use:

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Article

Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings

Davide Coraci, Silvio Brandi, Marco Savino Piscitelli  and Alfonso Capozzoli * 

TEBE Research Group, BAEDA Lab, Department of Energy “Galileo Ferraris”, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy; davide.coraci@polito.it (D.C.); silvio.brandi@polito.it (S.B.); marco.piscitelli@polito.it (M.S.P.)

* Correspondence: alfonso.capozzoli@polito.it

Abstract: Recently, a growing interest has been observed in HVAC control systems based on Artificial Intelligence, to improve comfort conditions while avoiding unnecessary energy consumption. In this work, a model-free algorithm belonging to the Deep Reinforcement Learning (DRL) class, Soft Actor-Critic, was implemented to control the supply water temperature to radiant terminal units of a heating system serving an office building. The controller was trained online, and a preliminary sensitivity analysis on hyperparameters was performed to assess their influence on the agent performance. The DRL agent with the best performance was compared to a rule-based controller assumed as a baseline during a three-month heating season. The DRL controller outperformed the baseline after two weeks of deployment, with an overall performance improvement related to control of indoor temperature conditions. Moreover, the adaptability of the DRL agent was tested for various control scenarios, simulating changes of external weather conditions, indoor temperature setpoint, building envelope features and occupancy patterns. The agent dynamically deployed, despite a slight increase in energy consumption, led to an improvement of indoor temperature control, reducing the cumulative sum of temperature violations on average for all scenarios by 75% and 48% compared to the baseline and statically deployed agent respectively.

Keywords: automated system optimisation; building adaptive control; deep reinforcement learning; soft actor-critic; heating system



Citation: Coraci, D.; Brandi, S.; Piscitelli, M.S.; Capozzoli, A. Online Implementation of a Soft Actor-Critic Agent to Enhance Indoor Temperature Control and Energy Efficiency in Buildings. *Energies* **2021**, *14*, 997. <https://doi.org/10.3390/en14040997>

Academic Editor: Giovanni Pernigotto

Received: 2 January 2021
Accepted: 9 February 2021
Published: 14 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Buildings are rated among the most energy-intensive uses, consuming approximately 40% of the worldwide energy demand, with CO₂ emissions of up to 36% [1]. In this context, buildings have seen the progressive evolution of energy systems due to a consistent introduction of renewable energy sources and storage systems, also thanks to incentive programmes (such as “20-20-20”) [2] conceived of to support the reduction of energy waste through the implementation of optimal control and energy management strategies. Building energy management has become a complex process, considering that modern energy systems have to respond to on-site intermittent renewable sources, energy storage, grid requirements and electric vehicle charging, in addition to traditional services such as lighting, ventilation, and air conditioning [3].

HVAC often represents the most energy-intensive building use [4], especially in the non-residential sector, although significant improvements have been implemented in recent years to enhance their energy efficiency.

One of the main challenges related to the optimal management of HVAC systems consists in handling the nonlinearity of the control problem given the effect of several stochastic endogenous (e.g., occupant behaviour) and exogenous (e.g., weather conditions) factors. Therefore, researchers have focused their attention on innovative control strategies for HVAC systems, capable of maintaining indoor thermal comfort conditions and, at the same time, reducing their energy consumption. Eventually, these strategies need to be also

able to handle signals from the electrical grid in order to meet power requirements and improve the grid reliability and stability [3,5].

These opportunities were provided by the introduction of Automated System Optimisation (ASO) tools based on artificial intelligence (AI), which can contribute to significantly enhancing the energy flexibility of a building. The energy flexibility represents a fundamental characteristic of smart buildings that defines their ability to adapt and respond to grid requirements, climatic conditions and user needs [6,7].

Traditional controllers, such as ON/OFF or Proportional-Integrative-Derivative Control (PID), could be inadequate due to their inability in foreseeing the dynamic changes that impact the operation of energy systems [8].

The ON/OFF control regulates a process operating within boundaries, while the PID requires the tuning of the control parameters prior to its deployment [6]. Although the latter has better performance than ON/OFF controllers, it could still be inadequate because the performance of the PID decreases significantly when the operating conditions are different from the tuning conditions in which the constants that regulate the control are calibrated [9]. Today, PIDs are the most common bottom-level control systems, while rule-based control systems (RBC) are considered the standard for optimising the management and top-level control of HVAC systems, but the resulting energy savings are limited [10,11]. In fact, RBC strategies are fixed and not scalable to different climatic conditions or building features as well as unable to predict changes in HVAC systems [3].

To overcome such limitations, the application of model-based control strategies has been explored in the last few years. In particular, Model Predicted Control (MPC), has become a dominant control strategy in research on intelligent building operation [12]. However, despite the implementation of MPC has shown its excellent ability of improving thermal comfort and reducing energy consumption between 15 and 50% [13–15], in both simulative [16] and real test environments [17], its model-based nature represents in some cases a critical aspect.

In fact, a significant problem in MPC is related to the performance dependency on the model definition for the control strategy optimisation, which is often challenging to design with good accuracy. Moreover, it is challenging to generalise its use on different types of plants and buildings. As a result, MPC controllers have not been adopted as expected in the building industry despite the promising results [18].

As a consequence, research has also focused on the development of model-free controllers based on Machine Learning (ML), which has shown great potential for improving the performance of buildings [19], particularly when Reinforcement Learning (RL)-based algorithms [20] are implemented. The interest in control strategies based on RL has increased since it does not require prior knowledge of the process or environment to be controlled.

A control agent based on Reinforcement Learning algorithms learns an optimal control policy by the direct interaction with the environment through a reward mechanism which depends on the control action performed in a specific state of the controlled environment [21]. Due to its simplicity, Q-Learning is the most commonly used technique in the Reinforcement Learning field [22,23]. However, in some circumstances, this algorithm proves to be not optimal when implemented in building control systems that require the definition of a high number of states and actions and greater complexity in the exploration of nonlinear relationships for the controlled environment. To overcome this obstacle, deep neural networks have been employed to approximate the policy function, as in Deep Reinforcement Learning (DRL).

In the next section, reference studies about the use of the DRL control strategies in HVAC systems control are reviewed, along with the motivations and the novelty of the present contribution.

Related Works, Motivations and Novelty of the Paper

The interest in Reinforcement Learning for the control of energy systems has grown in recent years, even if its applications still remain limited.

The first RL application in the energy and building research field dates back to 1998, in which [24] RL was used on an HVAC system serving the DOE (Department of Energy) at Idaho State University, to control the water temperature leaving boiler and the temperature in the two thermal zones served by the system. The applications of DRL for the control of HVAC systems have increased considerably, with the aim of regulating different parameters such as: supply water temperature setpoint [5,25,26], storage tank temperature setpoint [27–29] supply air-flow rate [30], supply air temperature [31], indoor temperature setpoint [29,32–34], frequency of pumps and fans [35,36]. Zhang et al. [5] applied a DRL control type Asynchronous Advantage Actor-Critic (A3C) in a water-based Radiant Heating System, in which the hot water pipes were integrated into window mullions with the objective of reducing the energy consumption of the system while ensuring the internal comfort of the occupants. The adaptive control system operated on the supply water temperature setpoint of the radiant heating system and was able to reduce energy consumption by 16.7% with a slight increase in the Percentage of Person Dissatisfied (PPD).

A further application of DRL control strategies on radiant heating systems was recently proposed by Brandi et al. [26], in which an algorithm based on Deep Q-Network was used to control the supply water temperature of the boiler in an office building. In that case, both static and dynamic deployment of the DRL controller were tested, and the heating energy saving in relation to a climatic rule-based control logic ranged between 5 and 12%, with enhanced indoor temperature control with both types of deployment. Moreover, that study highlighted the significance of an initial application of this control system into a simulated environment: the direct real-implementation would cause a reduction in the control performance since a DRL agent takes a long time to converge towards an acceptable control policy [26,28,37]. Therefore, in most cases reported in the literature, an initial simulation phase is performed in which various tools are combined, such as EnergyPlus [38] or CitySim with deep learning libraries such as Tensorflow [39].

In other studies, Vázquez-Canteli et al. [27,28] applied an algorithm based on Deep Q-Learning to control a heat pump coupled with a cold water storage to minimise its electrical energy consumption achieving 10% of energy saving while maintaining adequate indoor temperature conditions compared to a Rule-Based Control (RBC). Yoon et al. [30] proposed a performance-based thermal comfort control using Double Deep Q-Network to study the existing trade-off between building energy efficiency and improved comfort conditions for the occupants. Keeping the PMV value within the comfort range [40–42], the proposed control strategy made it possible to reduce the VRF energy consumption by 32% and the humidifier system energy consumption by 12% when compared to a fixed RBC.

Some published works have been related to the application of DRL on occupant centric control problems in the built environment. In this context, Park and Nagy [32] presented HVACLearn, a Reinforcement Learning (RL)-based Occupant-Centric Controller (OCC) for dynamically setting the thermostat setpoint, in which the indoor temperature, occupant behaviour and thermal vote were monitored to satisfy occupant comfort needs and enhance energy efficiency. The authors assumed that occupants could express their thermal sensation in real time by pressing two buttons of a sensing unit: too cold or too hot; an increase in the number of buttons pressed indicated a worsening of the indoor temperature-related comfort conditions. The authors simulated HVAC Learn control in a single occupant office with occupant behaviour models and, compared to an RBC, obtained a significant reduction of hot sensation feedbacks from the occupants while consuming the same or less cooling energy.

This literature review proves that Deep Reinforcement Learning represents a great opportunity for the future control systems of HVAC systems, being in most cases more efficient than current widespread control systems in terms of improving energy efficiency and comfort conditions in buildings.

However, the majority of papers in the literature consider the deployment of the DRL agent after its offline pre-training, making this control strategy similar to that based on a model, since the effort required in the pre-training phase indirectly led to the definition of an offline model. The online training of the controller, instead, which would preserve the free-model nature of the RL simulating its direct implementation in a real context, was poorly explored in literature. For this purpose, in this paper, the online training of the DRL controller is performed on a simulation environment combining EnergyPlus and Python for an office building. In this way, it is possible to simulate the initialisation of the controller as it was implemented in a real building. In this case, the learning process of the agent consists of control strategy optimisation on the basis of the information retrieved in real time from the controlled environment.

Moreover, some issues related to the application of off-policy algorithms and the possibility of implementing a model-free algorithm in real world context arise from the literature.

In the case of on-policy algorithms, the agent tries to evaluate and improve the same policy used for action-selection. However, the controller does not explore other control policies than the optimal one learned during exploration, which may result in reaching a local optimum. Thus, an off-policy algorithm would allow the agent to continuously explore other policies for extracting action sets useful for optimising the current policy. This parallel learning process speeds up the learning process, reducing the time required for the control policy to converge to the optimum.

Therefore, in this paper, an off-policy algorithm named Soft Actor-Critic is implemented [43]. This algorithm ensures the continuous agent exploration of new policies, but at the same time prevents the possibility that it transposes lousy trajectories. Another benefit provided by the Soft Actor-Critic algorithm is represented by the opportunity to choose actions in a continuous range, which is advantageous in the control problem of energy systems.

In this paper, an online implementation of a DRL controller is designed to enhance the indoor temperature conditions while reducing the energy consumption related to the heating system. This agent is designed to ensure the maximisation of the objective function through the control of the supply water temperature to the heating terminal units of a thermal office zone.

A sensitivity analysis is carried out on the hyperparameters of the model to assess their impact on the performance achieved by the control agent during its online training. Among all the DRL controller configurations, the one that ensured the best performance against a climatic rule-based control baseline is selected to perform a scenario analysis. The best configuration is adopted for testing the adaptive skills of the control agent by changing some forcing variables, consisting of eight different control scenarios. The adaptive performance of the agent is investigated following two deployment approaches, dynamic and static, in which the control agent is able or not, respectively, to update the control policy when changes are introduced in the environment.

Based on the literature review on DRL control logic in HVAC systems, the novelties that are introduced in this study can be summarised as follows:

- A DRL agent is developed using the SAC algorithm for controlling the supply water temperature in a heating system operating in a real building. To the best of the authors' knowledge, the SAC algorithm has been poorly explored, although it is recommended for controlling energy systems prone to dynamic changes in environmental conditions, such as HVAC systems. The SAC controller performance is compared with a rule-based climatic logic, to enhance the indoor temperature conditions and the energy efficiency of the system.
- The control agent, different from most of the papers reported in the literature, is trained online using a calibrated energy model of a real building, using an integrated environment coupling EnergyPlus and Python. In this way, the performances of

the agent can be analysed, as it is implemented in a real building without offline pre-training.

- A sensitivity analysis on the hyperparameters is performed during the online training, in order to choose the agent employing the combination of hyperparameters that provides the best performance compared to the RBC baseline.
- The adaptability of the DRL agent deployed online was tested for several different scenarios, including changes of outdoor weather conditions, indoor temperature requirements, occupancy schedules and internal heat capacity, for two different deployment approach, dynamic and static.

The rest of the paper is organised as follows. In Section 2, the theoretical aspects of RL and the algorithms are provided. In Section 3, the case study is introduced, while the implemented methodological framework is described in Section 4. Section 5 provides information about the simulation environment, the definition of the baseline controller and the design of the proposed DRL control logic. Moreover, Section 5 also provides details on the deployment and scenario analysis. The results are presented in Section 6 and discussed in Section 7. Section 8 includes the conclusions and includes future research directions.

2. Reinforcement Learning Overview

In the Reinforcement Learning framework, a control agent learns the optimal control policy through a direct interaction with the environment through a trial-and-error approach. RL can be formalised as a Markov Decision Process (MDP) that is defined by a four-values tuple, including: state, action, transition probabilities and reward. The state is a set of variables whose values provide a representation of the controlled environment. The action is the control signal that the agent performs on the control environment in order to maximise its goals encoded in the reward function. The transition probabilities define the probability of the environment to move from a state s to a state s' when an action a is performed. According to MDP theory, these probabilities depend only on the value of state s and not on the previous states of the environment [44].

The RL framework applied to a control problem in the energy management domain can be simplified as follows: a control agent (e.g., a control module linked to the building management system) interact with an environment (e.g., a thermal building zone). At each time step, the agent performs an *action* (e.g., setting of the supply water heating temperature) when the environment is in a certain *state* (e.g., the building is occupied, but the temperature is above the upper comfort band limit). The agent receives, along with the observation of the state, a *reward* that measures the performance of the agent with respect to its control objectives. Starting from a certain state of the environment the RL algorithm tries to identify the optimal control policy π that maximises the cumulative sum of future rewards. The optimal policy is determined by the RL agent by evaluating two functions, namely state-value function, and action-value function.

The state-value function represents the goodness of being in a certain state S_t with respect to the control objectives [45]. This function provides the expected value of the cumulative sum of future rewards that the control agent would obtain by starting from state S_t and following the policy π and it is defined as follows:

$$v_{\pi}(s) = E[r_{t+1} + \gamma v_{\pi}(s') | S_t = s, S_{t+1} = s'] \quad (1)$$

where γ [0,1] is the discount factor for future rewards. For $\gamma = 0$ the agent will give greater importance to immediate rewards. Conversely, for $\gamma = 1$ the agent will seek to maximise only future rewards.

Similarly, the action-value function represents the goodness of taking a certain action A_t in a certain state S_t following a specific control policy π [46]. The action value function can be expressed as follows:

$$q_{\pi}(s, a) = E[r_{t+1} + \gamma q_{\pi}(s', a') | S_t = s, A_t = a] \quad (2)$$

In the context of RL algorithms the most widely applied approach is the Q-Learning. Q-Learning exploits a tabular approach to map the relationships between states and action pairs [25]. These relationships are formalised as state-action values or Q-values, which are gradually updated by the control agent through the *Bellman equation* [47]:

$$Q(s, a) = Q(s, a) + \mu[r_t + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (3)$$

where μ [0,1] is the learning rate which determines how rapidly new knowledge overrides old knowledge. With $\mu = 1$ new knowledge completely overrides what was previously learned by the control agent.

2.1. Deep Reinforcement Learning

Despite its effectiveness, Q-Learning may have inadequate results when the dimensions of the state and action spaces grows [24]. To overcome this limitation, Artificial Neural Networks (ANNs) [45,48] were recently introduced to substitute lookup tables in mapping the relationships between states and actions in order to determine the optimal control policy. One of the main advantages of ANNs is their capability to handle continuous inputs and outputs making them suitable for a variety of control problems [26]. Moreover, they have proven to be effective methods to capture hidden and nonlinear pattern among data [49,50]. RL algorithms which employ ANNs, and in particular Deep Neural Networks (DNN), are identified as Deep Reinforcement Learning (DRL) [51].

2.2. Soft Actor-Critic

In this work a particular DRL control algorithm, namely Soft Actor-Critic (SAC), was implemented. SAC is an off-policy algorithm based on the maximum entropy RL framework, introduced by Haarnoja et al. [43]. Different from Q-learning, SAC is capable of handling continuous action spaces, enhancing its applicability to a variety of control problems. Soft Actor-Critic uses a particular Actor-Critic architecture that employs two different deep neural networks for approximating action-value function and state-value function. The *Actor* maps the current state based on the action it estimates to be optimal, while the *Critic* evaluates the action by calculating the value function.

The main feature of SAC is the entropy regularisation: this algorithm is based on the maximum entropy reinforcement learning framework, in which the objective is to maximise both expected reward and entropy [52] as follows:

$$\pi^* = \operatorname{argmax}_{\pi} E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha H_t^{\pi}) \right] \quad (4)$$

where H is the Shannon entropy term, which expresses the attitude of the agent in taking random actions, and α is a regularisation coefficient that indicates the importance of the entropy term over the reward. Generally, α is zero when considering conventional reinforcement learning algorithms.

The maximisation of this target function has a close connection with the exploration-exploitation trade-off, ensuring that the agent is explicitly pushed towards the exploration of new policies and at the same time avoids that it gets stuck in sub-optimal behaviour.

3. Case Study

The building investigated is an office building located in Turin, Italy, and consists of five heated floors and a basement, with a net heated surface of about 9300 m² (Figure 1). The five heated floors are organised into two heating zones: the ground floor, and the remaining four floors.

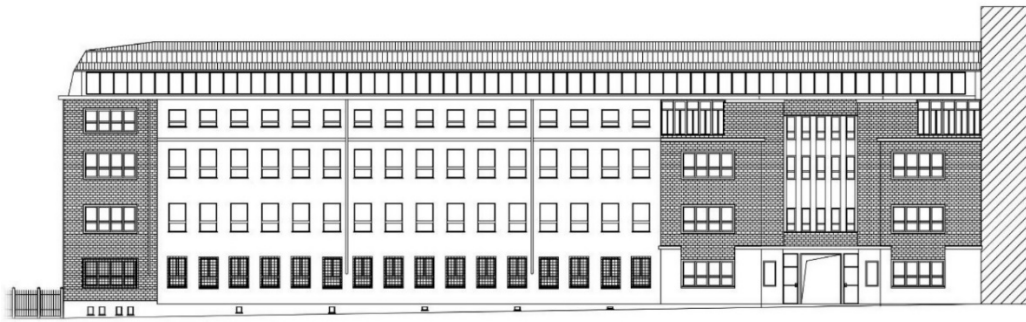


Figure 1. Front view of the external façade for the building considered in the case study.

The average transmittance values of the opaque and transparent envelope components are 1.08 and $2.7 \text{ W/m}^2\text{K}$, respectively. The occupancy schedules were defined on the basis of the actual office opening and closing times. Every weekday, with the exception of Sundays and holidays, the office is occupied from 7 a.m. to 7 p.m.

The two thermal zones are served by a hot water circuit consisting of two loops connected by a heat exchanger. Four gas-fired boilers (with a total nominal capacity of 1300 kW) are installed in the primary loop, serving a collector from which different pumps withdraw water to serve the radiator terminals in the secondary loop.

The supply water temperature is regulated in the installed control strategy through a constant speed pumping system and a three-way valve. The heating system implemented in the EnergyPlus simulation model consists of two different circuits serving each thermal zone with a gas-fired boiler that supplies hot water to the radiators for each circuit (Figure 2).

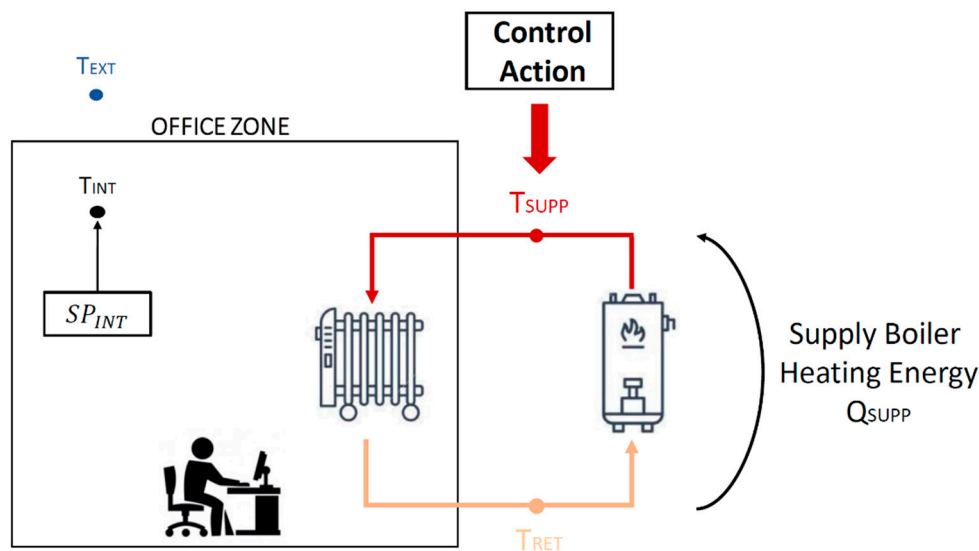


Figure 2. Scheme of the heating system considered in the case study.

This study was focused only on the indoor air temperature control, since the building is served by water terminal units, able to operate only on the sensible part of thermal load.

The energy model was calibrated through an iterative approach, using hourly indoor temperature and monthly boiler gas consumption data available for the real building. Thermo-physical properties were extracted from an energy diagnosis report that was available for the building under analysis. A manual calibration based on an iterative approach was performed, varying the infiltration airflow rate, internal gains and equipment power according to [53]. However, considering the available information, the model was

not calibrated from an indoor comfort point of view. This could lead to discrepancies regarding the real values of the calibrated variables [54].

In this context, two statistical indices (MBE and Cv(RMSE)) were calculated, as suggested by the ASHRAE guidelines [55], in order to prove the accuracy of the calibration process. The results obtained after the calibration process are reported in Table 1.

Table 1. Results of the calibration process.

Statistical Indices	Building Energy Model		Limits from ASHRAE Guideline [55]	
	Temperature (Hourly)	Consumption (Monthly)	Hourly	Monthly
MBE[%]	3.63	−1.06	±10	±5
Cv (RMSE)[%]	8.57	5.62	30	15

The objective of the developed DRL controller is to maintain the desired temperature conditions during occupancy while consuming the minimum heating energy as possible, through the regulation of the supply water temperature to heating terminal units of the office zone. The controller is penalised if the indoor air temperature falls outside the temperature acceptability range, defined between $[-1, 1]$ °C from the desired indoor temperature value of 21 °C. The same acceptability range 21 ± 1 °C was considered for the baseline controller.

4. Methodological Framework

The methodological framework adopted in this paper unfolds over two different stages, as shown in Figure 3:

1. Design of the DRL control problem
2. Deployment phase

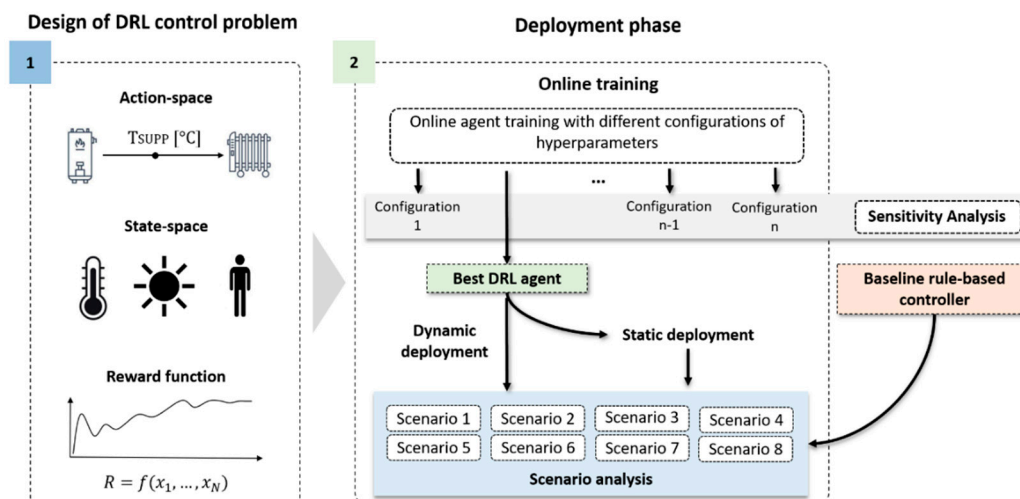


Figure 3. Methodological framework of the analysis.

4.1. Design of the DRL Control Problem

Initially, the main components of adaptive control logic were defined, i.e., action-space, state-space and reward function. The control actions to be selected by the controller were included in the action-space. The state-space includes observations concerning the state of the controlled environment for each control time step. This information is provided to the agent to learn the optimal control policy that maximise the objective function of the problem. The design stage was concluded through the definition of the reward function. This feature was defined accordingly with the target established for the agent.

4.2. Deployment Phase

After defining the action-space, the state-space and the reward function, the implementation of the DRL agent was investigated. The deployment framework consisted of two phases: the online training of the DRL agent and a scenario analysis. In both cases each simulation involved a three-month heating season, from 1st December to 28th February, repeated only for one episode to simulate the direct implementation of the DRL controller in a real-world context. Firstly, during the online training phase, the influence of hyperparameters in the performance of the algorithm was analysed through a sensitivity analysis. The performances of the controller associated with the different hyperparameters configurations were also compared with the Rule-Based Control (RBC).

Successively, the DRL controller characterised by the best configuration of hyperparameters was exploited for testing its adaptability to the change of specific boundary conditions during the scenario analysis. All scenarios were deployed both dynamically and statically, which means that the control agent was able (as in the online training) or not to update the optimal control policy according to the change of boundary conditions. These changes were grouped in eight different scenarios, in which the weather conditions (i.e., considering a new heating season by changing the weather file), the indoor temperature conditions, the occupancy patterns, the characteristics of the transparent and opaque envelope (such as internal heat capacity) were modified. The agent performance in the dynamic deployment phase was compared to the static deployment and the rule-based baseline controller. The same acceptability range for the indoor air temperature was considered for a robust comparison between baseline and DRL controllers. However, the selection of the acceptability range can affect the results.

5. Implementation

This section discusses the developed simulation environment and the formulation of the rule-based baseline and DRL controllers. Subsequently, the setting up of online training and scenario analysis are reported.

5.1. Simulation Environment

The interaction between the control agent and the building was simulated within a surrogate environment that integrates Energyplus v9.2.0 and Python interface based on OpenAI Gym [56]. The Tensorflow [39] library was used to allow the control agent to learn optimal control policy. In particular, a Building Control Virtual Test Bed (BCVTB) and the ExternalInterface from EnergyPlus were used to connect the two software programmes. The information exchange between the two software programmes was the same as proposed in [26]. Time steps differ between control and simulation, and in this work, the control time step was set to 30 minutes, while the simulation time step was set to 5 minutes. It was verified that a simulation time step of 5 minutes ensures an appropriate trade-off between an optimal convergence of numerical results and an acceptable computational cost. The control time step was set equal to six times the simulation time step in order to properly take into account the system time constant for the control problem addressed in this paper. A similar approach was adopted in [5,26,31].

Moreover, the simulation environment was set in order to:

- allow the system to turn-ON/OFF at any time during the day;
- keep the system switched-OFF on Sundays.

5.2. Rule-Based Baseline Control Logic

The performance of the DRL controller was evaluated against a baseline controller consisting of a combination of rule-based and climatic-based logics for the control of the supply water temperature of the considered heating system. The supply water temperature varies linearly from 40 °C to 70 °C when outdoor temperature ranges between 12 °C and −5 °C. These values were selected according to those implemented in the real Building Energy Management System (BEMS). This control strategy is in operation for up to one

hour before the occupants leave the building (indicated as $t_{end,occ}$). The RBC controller manages the boiler switch-ON according to the difference between the lower bound of the temperature comfort range and the actual indoor temperature, and the time before the arrival of occupants. Specifically, the boiler is switched-ON:

- between four and three hours before the arrival of the occupants if the temperature difference is greater than 3 °C;
- between three and two hours before the arrival of the occupants if the temperature difference is greater than 2 °C;
- two hours before the arrival of the occupants if the temperature difference is greater than 0.

After the switch-ON of the heating system, the control logic follows the sequence illustrated in Figure 4. The boiler is turned-OFF at any time when the indoor thermal zone temperature T_{INT} is higher than the upper bound of the temperature comfort range T_{UPP} , equal to 22 °C.

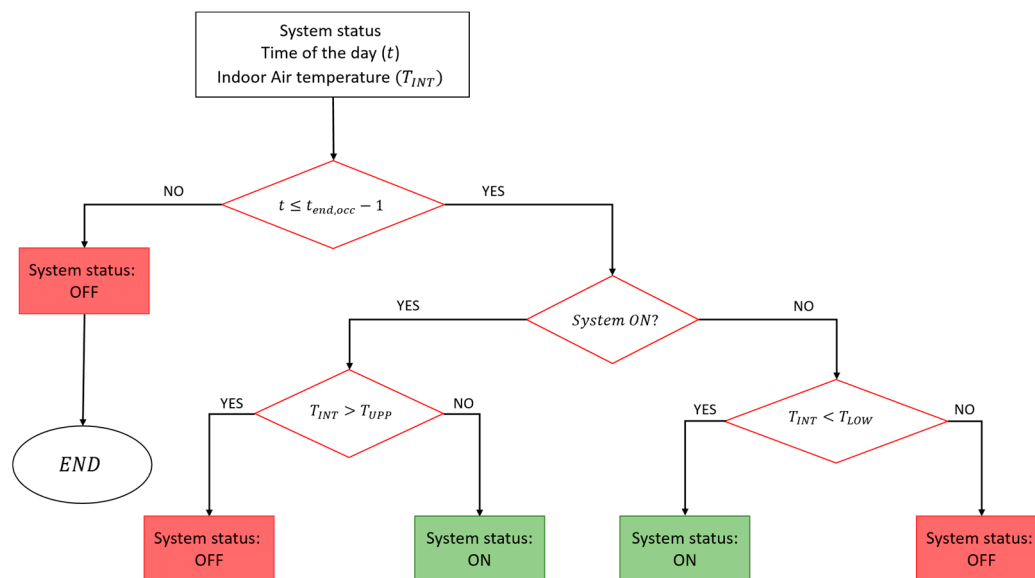


Figure 4. Baseline control logic after the first boiler switch-ON.

In addition, if the zone is occupied and the temperature drops below the lower threshold limit temperature T_{LOW} , equal to 20 °C, the system is turned-ON.

The boiler is set in the OFF condition on Sundays due to the office being closed.

5.3. Design of the DRL-SAC Control Logic

This section is focused on the design of DRL control algorithm, in order to define the action-space, the state-space and the reward function.

5.3.1. Design of the Action-Space

The action supplied by the controller is the supply water temperature value to the radiators. The action-space includes values between 20 °C and 70 °C and was expressed as a continuous space, since SAC was selected as the control algorithm. Hence, the action chosen is:

$$A_t: 20 \leq T_{SUPPt} \leq 70 \quad (5)$$

Moreover, the circulation pump is shut down if the chosen action is equal to or lower than 25 °C.

5.3.2. Design of the State-Space

The state-space is composed of a series of observations provided as inputs to the agent. The values assumed by the space-state variables influence the control action. The state-space was made of 52 features, reported in Table 2, together with their lower and upper bounds. The variables chosen are feasible to be collected in a real-world implementation and provide to the agent the information necessary to predict the immediate future rewards. The *Indoor Air Temperature* information during each control step was described as difference between the indoor temperature setpoint SPINT and the actual indoor air temperature TINT as directly linked to the reward formulation, as shown in Section 5.3.3. This quantity is memorised in the state-space at the current control time step $\Delta t_{control}$ and for three previous control steps, $\Delta t_{control,-1}$ (30 min before), $\Delta t_{control,-2}$ (1h before) and $\Delta t_{control,-4}$ (2h before). *Outdoor Air Temperature* was included because it is the exogenous factor with the most significant influence on the heating energy consumption. *Occupants' Presence status* indicates if, in a certain control time step, the zone is occupied or not (based on the occupancy schedules) through a (0,1) binary variable. The two last features are saved in the state-space at the current control time step $\Delta t_{control}$ and for all control time steps within the following 12 h. Observations were scaled within a range of (0, 1) in order to feed the neural network with min-max normalisation.

Table 2. Variables included in the state-space.

Variable	Min Value	Max Value	Unit
ΔT Indoor Setpoint—Mean Indoor Air	−3	8	°C
ΔT Indoor Setpoint—Mean Indoor Air, 30 min before	−3	8	°C
ΔT Indoor Setpoint—Mean Indoor Air, 1 h before	−3	8	°C
ΔT Indoor Setpoint—Mean Indoor Air, 2 h before	−3	8	°C
Outdoor Air Temperature (and future 12 h predictions)	−8	32	°C
Occupants' Presence status (and future 12 h predictions)	0	1	-

5.3.3. Design of the Reward Function

The reward function was formulated as a linear combination of two different parts, an energy-related term and the temperature-related term. They were combined employing two weights (δ and β , respectively) that were made to vary in order to change their importance.

Equation (6) describes the reward function r implemented in this work that the SAC algorithm exploits to learn the optimal control policy as introduced in equations form (1) to (4).

The energy term (r_E) refers to the heating energy consumption, calculated in kWh. The temperature term (r_T) was evaluated as the temperature difference between the upper/lower threshold temperature and the indoor air temperature evaluated in °C. The general expression of the reward is as follows:

$$r = r_E + r_T \quad (6)$$

The objective of the agent is to maximise the reward, with a maximum (ideal) value of zero. On Sunday, due to the absence of occupants, the reward value was set to zero.

The energy term of the reward is expressed in the following way:

$$R_E = -\delta * E_{HEAT} \quad (7)$$

where E_{HEAT} is the energy heating consumption for the current control time step.

The temperature term, on the other hand, has different expressions depending on the presence of occupants.

If no occupants are present, the temperature term is:

$$r_T = 0 \quad (8)$$

If occupants are present, the temperature term could have three different expressions:

- if $T_{INT} < T_{LOW}$:

$$r_{T,OCC=1} = -\beta * (SP_{INT} - T_{INT})^2 \quad (9)$$

- if $T_{INT} > T_{UPP}$:

$$r_{T,OCC=1} = -\beta * (T_{INT} - SP_{INT})^3 \quad (10)$$

- if $T_{LOW} \leq T_{INT} \leq T_{UPP}$:

$$r_{T,OCC=1} = 0 \quad (11)$$

This formulation of the temperature term was chosen to try to speed up the learning process and to avoid the exploration of unacceptable states from the beginning of the process.

5.4. Online Training Setup

The use of a DRL algorithm implies the selection of several hyperparameters that influence the behaviour and performance of the agent. To assess their influence, a sensitivity analysis was performed.

The hyperparameters involved in sensitivity analysis were:

- temperature—term weight reward factor β ;
- energy—term weight reward factor δ ;
- learning rate μ ;
- gradient steps, which indicate the number of gradient updates after each step.

The weight factors help to define the relative importance of the two reward terms and therefore to determine the agent choices in investing greater attention to comfort instead of energy consumption.

Three piecewise functions were defined for both gradient steps (GS1, GS2, GS3) and learning rate (LR1, LR2, LR3), as detailed in Figure 5. The value of these two features decreased according to the percentage progress (between 0% and 100%) of the simulation. The choice of designing these two parameters as time-varying functions allows the optimisation of the online training.

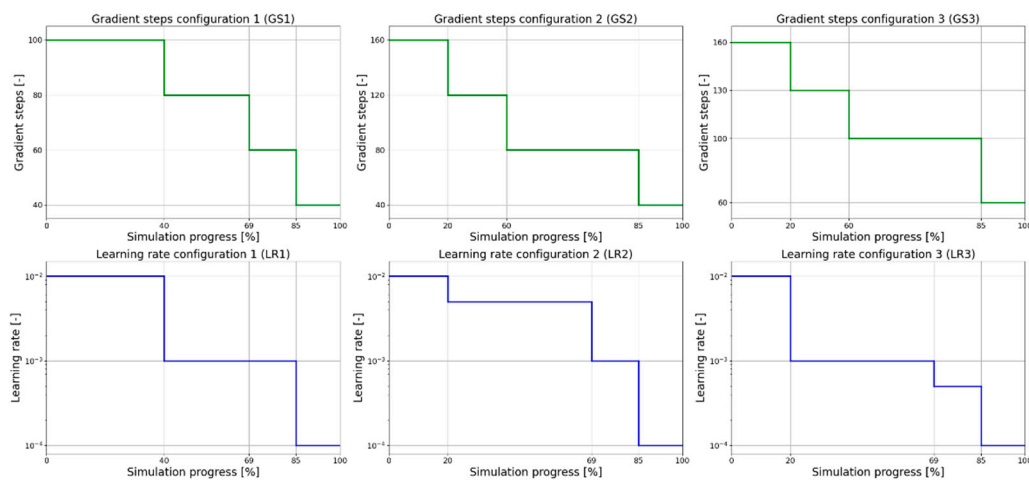


Figure 5. Piecewise functions assumed during simulation for gradient steps (GS1, GS2, GS3) and learning rates (LR1, LR2, LR3).

At the beginning of the simulation, the DRL controller has no adequate background concerning the action to be chosen. For this reason, gradient steps and learning rate values were set higher, in order to facilitate the exploration process. As the simulation

proceeded, both terms were decreased in order to allow the control agent to learn from the environment, but at the same time to not over-explore the space of the actions, as excessive exploration could cause a deviation from the optimal control policy that the agent might be able to learn at the beginning of the online training phase.

Within the set of hyperparameters, it was decided to keep those indicated in Table 3 fixed. A four-layer feed-forward neural network was used, with 64 neurons per hidden layer. The deployment episode, repeated only once, as required when performing an online training, had a duration of three months, from 1 December to 28 February, for a total of 4320 control steps (90 days) and 25,920 simulation steps (every 5 min). On average, the episode took 60 minutes to be simulated on a machine with an 8th Generation Intel® Core™ i7-8550U @ 4.0 GHz processor and 16.0 GB RAM. The weather file used in this phase refers to the heating season 2018/2019.

Table 3. Hyperparameters of DRL controller kept constant during the online training.

Variable	Value
DNN Architecture	4 Layers
Neurons per hidden layer	64
Batch size	32 Control Steps
Buffer size	1488 Control Steps (31 days)
Episode Length	4320 Control Steps (90 days)
Target Model Update	1 Control Step
Deployment Episodes	1
Learning Start	336 Control Steps (7 days)
Discount factor γ	0.9

The different configurations of hyperparameters tested are shown in Table 4.

Table 4. Hyperparameter configurations for DRL controller.

Run	B	δ	Gradient Steps	Learning Rate μ
1	10	0.05	GS1	LR1
2	20	0.05	GS1	LR1
3	10	0.05	GS2	LR1
4	10	0.05	GS2	LR3
5	5	0.01	GS2	LR1
6	10	0.05	GS2	LR2
7	20	0.05	GS2	LR1
8	20	0.05	GS2	LR3
9	20	0.1	GS1	LR1
10	20	0.1	GS2	LR2
11	20	0.1	GS2	LR3
12	20	0.1	GS3	LR1
13	20	0.1	GS3	LR2
14	20	0.1	GS3	LR3
15	10	0.05	GS3	LR2
16	10	0.05	GS3	LR3

5.5. Scenario Analysis

The configuration considered to be the best among those explored during the online training phase was chosen for testing the adaptability of the agent in eight different scenarios, for the same simulation period as the online training phase but in a different climatic heating season. The scenarios considered include variation of outdoor conditions, building features (thermal transmittance U , internal heat capacity χ_i , solar heat gain coefficient g), temperature setpoint and occupancy schedule. In detail, the weather file used in the online training phase was replaced with the reference weather file (ITA-

TORINO-CASELLE-IGDG.epw) available in EnergyPlus for Turin (Italy) to test a different heating season.

The agent was deployed dynamically and statically:

- in dynamic deployment, the RL agent is characterised by continuous learning as during the online training. For each control time step, the agent collects information about the state of the system, computes a control action, observes the reward value and the next state and then updates the control policy. In this case, the agent exhibits great flexibility to the continuous changing of system characteristics, indicating a good attitude in adjusting the control policy. On the other hand, this type of deployment requires higher computational cost, and the control policy may as a result be more unstable than the static case [5];
- in static deployment, the agent trained online was used as a static function. This process requires less computational time than the dynamic one, but the control agent needs to be re-trained when building features are changed in order to exhibit adaptive capabilities.

The DRL was deployed both statically and dynamically for each scenario and the performance achieved in these two cases was compared. The scenarios considered are summarised in Figure 6.

	Weather Conditions	Indoor Temperature Setpoint	Occupancy Schedule	Opaque Envelope	Transparent Envelope
S1	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 7AM–7PM	$U = 1.084 \text{ W/m}^2\text{K}$ $\chi_i = 62.7 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$
S2	EnergyPlus weather file for Turin (Italy)	22 °C	Mon–Sat 7AM–7PM	$U = 1.084 \text{ W/m}^2\text{K}$ $\chi_i = 62.7 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$
S3	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 7AM–7PM	$U = 0.211 \text{ W/m}^2\text{K}$ $\chi_i = 35.0 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$
S4	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 7AM–7PM	$U = 1.084 \text{ W/m}^2\text{K}$ $\chi_i = 62.7 \text{ kJ/m}^2\text{K}$	$U = 1.1 \text{ W/m}^2\text{K}$ $g = 0.33$
S5	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 8AM–6PM	$U = 1.084 \text{ W/m}^2\text{K}$ $\chi_i = 62.7 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$
S6	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 7AM–7PM	$U = 1.020 \text{ W/m}^2\text{K}$ $\chi_i = 28.2 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$
S7	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 7AM–7PM	$U = 0.434 \text{ W/m}^2\text{K}$ $\chi_i = 56.8 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$
S8	EnergyPlus weather file for Turin (Italy)	21 °C	Mon–Sat 7AM–7PM	$U = 0.431 \text{ W/m}^2\text{K}$ $\chi_i = 23.8 \text{ kJ/m}^2\text{K}$	$U = 2.7 \text{ W/m}^2\text{K}$ $g = 0.75$

Change not applied Change applied

Figure 6. Changes of the boundary conditions assumed in the eight analysed scenarios.

6. Results

The results obtained by implementing the SAC control logic are presented in this section. Initially, the performance of the DRL agent in the online training phase was compared with the baseline. Afterwards, during the scenario analysis, the results obtained for the agents deployed statically and dynamically are highlighted.

6.1. Online Training

As reported in Section 5.4, during the online training, a sensitivity analysis was preliminarily performed on some hyperparameters. Specifically, 16 configurations were evaluated, as shown in Table 4. To assess the goodness of each configuration, suitable metrics were introduced and compared.

In detail, two metrics were selected during the period of operation:

- the amount of energy needed for heating the supplied water to the terminals;

- the cumulative sum of temperature violations during the occupancy hours, measured in °C.

A temperature violation occurred when, during the occupancy period, the temperature was not within the acceptability range $(-1, 1)$ of the 21 °C setpoint. It was calculated as the absolute difference between the indoor temperature value and the lower or upper limit of the acceptability range, when the internal temperature was lower or higher than these limits.

The two metrics were evaluated for each simulation step and summed up at the end of the online training period. Moreover, the metrics were also calculated for the baseline to compare its performances with the DRL agent. The results obtained for the baseline and for the various configurations of hyperparameters are shown in Table 5. The results indicate that some configurations (i.e., Run 12-13-14-16) accomplish the objective of reducing the heating energy demand and improving the control of indoor temperature during the occupancy period.

Table 5. Online training results for the different configurations of hyperparameters.

Run	Energy Consumption E_{heat} [MWh]		Temperature Violations T_{viol} [°C]	
	Baseline	DRL Logic	Baseline	DRL Logic
1		240.3		547.5
2		244.2		331.6
3		233.2		840.4
4		243.0		853.1
5		258.2		394.4
6		235.9		676.2
7		261.8		510.0
8	234.4	253.0	328.3	206.4
9		240.3		433.3
10		237.3		362.5
11		240.5		292.8
12		227.8		263.7
13		231.3		121.9
14		226.7		306.7
15		225.9		438.6
16		225.3		287.1

The most significant benefit was represented by the improvement of maintaining the indoor temperature within the predefined band. In that sense, the best result was provided by configuration 13, in which the cumulative temperature violations, for the entire deployment period, was equal to 121.9 °C. This value was reduced by more than 60% compared to the baseline case. At the same time, for configuration 13, a slight reduction in heating energy consumption was obtained in comparison to RBC baseline.

Therefore, the agent considered configuration 13 to be the best solution after the sensitivity analysis.

Figure 7 shows the cumulative reward evolution for this solution. The reward does not have a direct physical meaning, as it gives indications on the control policy convergence of the proposed agent. A non-convergent trend could cause instability of the optimal control policy. Figure 7 is composed of three subplots. The top panel shows the cumulative total reward trend, while the two bottom panels report the evolution of the energy and temperature terms of the reward function. For the sake of clarity, the evolution of the temperature-term is reported in logarithmic scale.

The agent starts the exploration with relatively high values of the two terms of the reward. During the online training period, a recurring behaviour for both energy and temperature term can be observed. On Sunday, the reward was not computed, given the shut-down of the heating system due to the absence of occupancy; therefore, its value was set to zero, as highlighted in Figure 7. On Monday, the two reward terms show a

decrease due to the high amount of energy required to bring the environment to the desired temperature conditions and due to inability of the agent to guarantee these temperature conditions before the beginning of the occupancy period. During the last days of the heating season, the agent is able to better enhance the reward having gained experiences during the online training period.

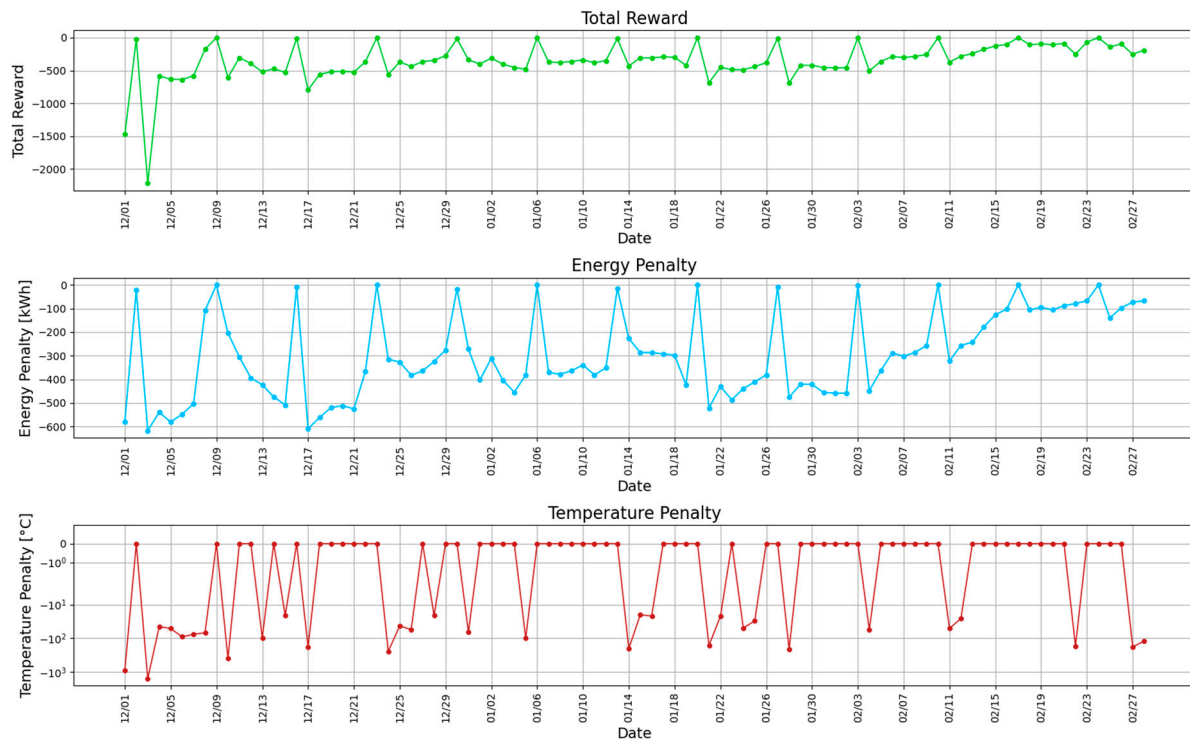


Figure 7. Reward evolution for the best DRL online trained agent.

Training a DRL control agent online without any offline pretraining may require a large amount of time before the performance obtained can be considered acceptable. The moving average over a 4-day time window was calculated for the cumulative sum of temperature violations as shown in Figure 8 to assess the time required for the best trained online DRL controller to perform similarly to the baseline. It was not necessary to calculate the moving average for energy consumption since it is similar as the baseline. Figure 8 compares the moving average for the DRL agent and the baseline, starting from the first day of its availability, i.e., the fourth day of the training period. During the first stages of the training period, the baseline performs better than the model-free agent: for the latter, the moving average trend suggests that the learning of the optimal control policy starts from the second week of training. This choice implies that the action-selection during the first week is randomised; thus, it is useful for the preliminary exploration of the action space and not for the optimisation of the control strategy.

During the third week, the DRL agent outperforms the baseline in establishing the desired indoor temperature conditions. Hence, the timeframe required to achieve performance at least similar to the baseline resulted reasonable.

Figure 9 compares the indoor temperature distributions for the baseline and DRL control logic during the occupancy period for each day of the week. The baseline and the DRL control logic were compared considering as objective the same acceptability range for the indoor temperature.

The DRL controller is capable of better maintaining the indoor temperature within the acceptability range (with a lower value of temperature violations). In addition, the indoor temperature is mostly maintained with the DRL controller in the lowest part of the acceptability range.

The temperature distribution comparison for the two control logics shows that the indoor temperature has a smaller range around the setpoint in the case of the adaptive control strategy for every weekday.

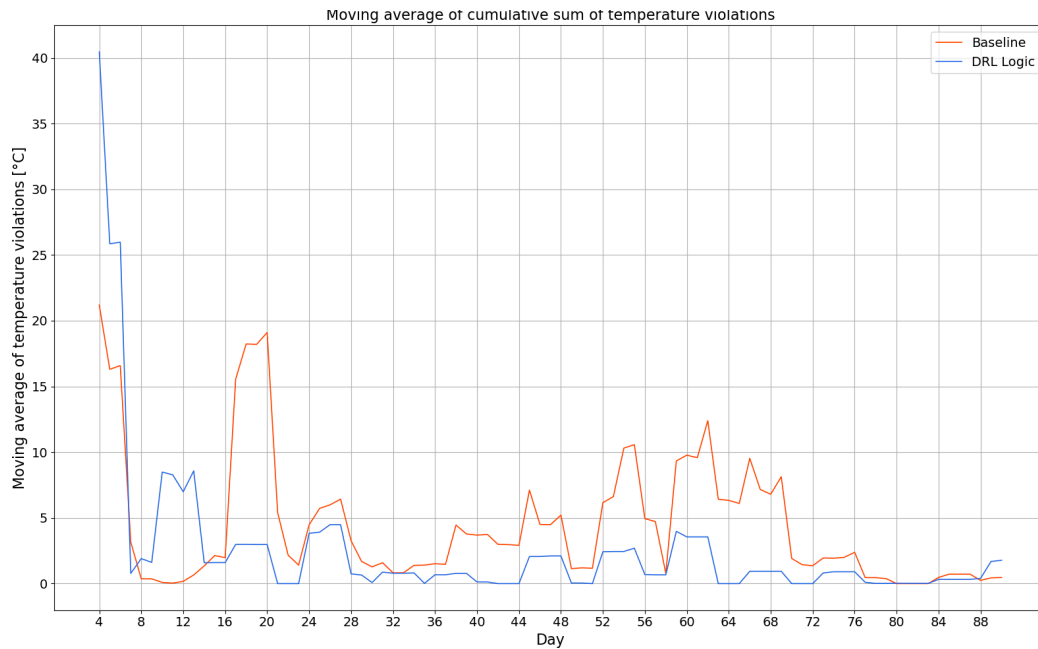


Figure 8. Moving average of cumulative sum of temperature violation for baseline and DRL controller (time window $W = 4$).

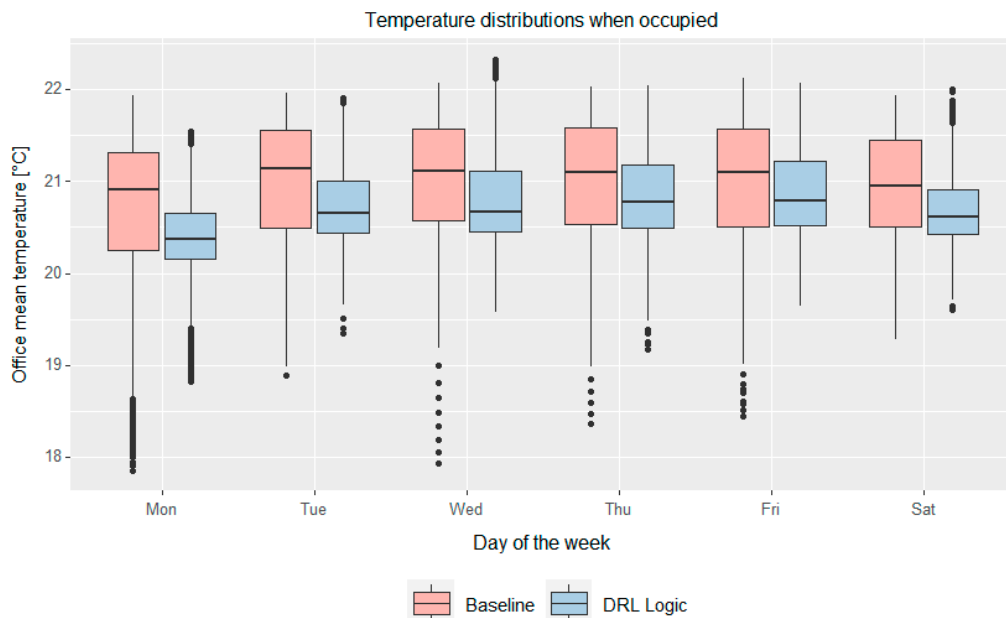


Figure 9. Box plots of indoor office temperature distribution per each working day for baseline and DRL controller.

6.2. Scenario Analysis

In this section, the results of the scenario analysis are reported, where the adaptability of the control agent selected in the online training phase was tested in the eight scenarios as proposed in Figure 6. The deployment of the agent was simulated both dynamically and statically. As previously mentioned, a deployment episode runs for 90 days, including December, January and February. The meteorological file used is the reference one for Turin

available in EnergyPlus, simulating the performance of the DRL control agent on a different heating season from the one considered in the online training phase. Table 6 summarises the results obtained for baseline, static and dynamic deployment, for all scenarios with reference to energy consumption and cumulative sum of temperature violations during the whole heating season.

Table 6. Performance of all the controllers for all the scenarios and deployment strategies.

Scenario	Energy Consumption E_{heat} [MWh]			Temperature Violations T_{viol} [°C]		
	Baseline	Static Deployment	Dynamic Deployment	Baseline	Static Deployment	Dynamic Deployment
1	407.2	429.2	457.6	3409.3	1386.1	651.5
2	426.9	459.3	493.1	7060.9	2822.7	1901.8
3	337.4	347.7	361.9	2058.4	812.1	553.0
4	379.8	399.1	423.1	3104.3	1433.9	887.3
5	417.0	404.2	457.5	2220.5	4288.2	988.0
6	398.2	420.6	456.5	3477.5	1279.9	664.7
7	358.1	370.1	391.8	2104.1	876.5	452.1
8	356.2	370.4	393.3	2462.7	881.7	416.0

The SAC agent provided a great advantage in terms of control of indoor temperature especially in dynamic deployment. On the other hand, the results for energy consumption were opposite to the previous ones, as the dynamically deployed agent required more energy than the statically deployed agent and the baseline for all scenarios.

The dynamically deployed agent led to an increase of the heating energy need to reach the set point temperature conditions during occupancy that results in the reduction of temperature violations. Although the control strategy provided by the dynamically deployed agent led to an energy consumption over the entire period on average about 11% and 7% greater than the baseline and the statically deployed agent, respectively, it ensured an average reduction of 75% and 48% in terms of cumulative temperature violations. In particular, the temperature violations for the baseline controller refer indoor temperature values mostly below the lower bound of the acceptability range. This is the main reason for the lower energy consumption of the RBC controller.

In Figures 10–12, detailed results are reported for some scenarios, with the aim of highlighting the behaviour of the dynamically deployed agent and comparing it with the baseline or with the static counterpart.

Figure 10 reports the comparison between the dynamically deployed and baseline control agent in scenario S1 during a typical week of the analysed period. The figure shows the indoor temperature patterns and the supply water temperature for these two controllers, along with the outdoor air temperature profile.

Monday is the most critical day, since during Sunday the heating system has been switched-OFF. Therefore, the indoor temperature during the first switch-ON phase on Monday is much lower than on the other days of the week: this requires careful management of the pre-heating phase to prevent the indoor temperature to fall outside the temperature range for comfort conditions on Monday, as occurred for the baseline controller highlighted in Figure 10.

The adaptive control agent is able to reduce the temperature violations through an optimal management of the pre-heating phase maintaining the indoor temperature profile during the occupancy phase close to the lower part of the temperature band, to also limit the heating energy demand.

On other weekdays, the SAC agent succeeds in avoiding temperature violations respect to the baseline agent which, instead, reports violations during the first occupation phase. Above all, this phenomenon could be attributed to the flexibility of the agent concerning outdoor temperature conditions, since the agent receives information about the outdoor temperature pattern for the 12 h ahead the control time step considered. Therefore, the agent keeps the heating system ON until the end of the occupancy phase of

the day, considering that the outdoor temperature was lower in the early morning, as can be observed from the supply water temperature values shown in Figure 10. In this way, the dynamic deployment manages the switch-ON phase better than the baseline, avoiding temperature violations during the first hours of occupancy period.

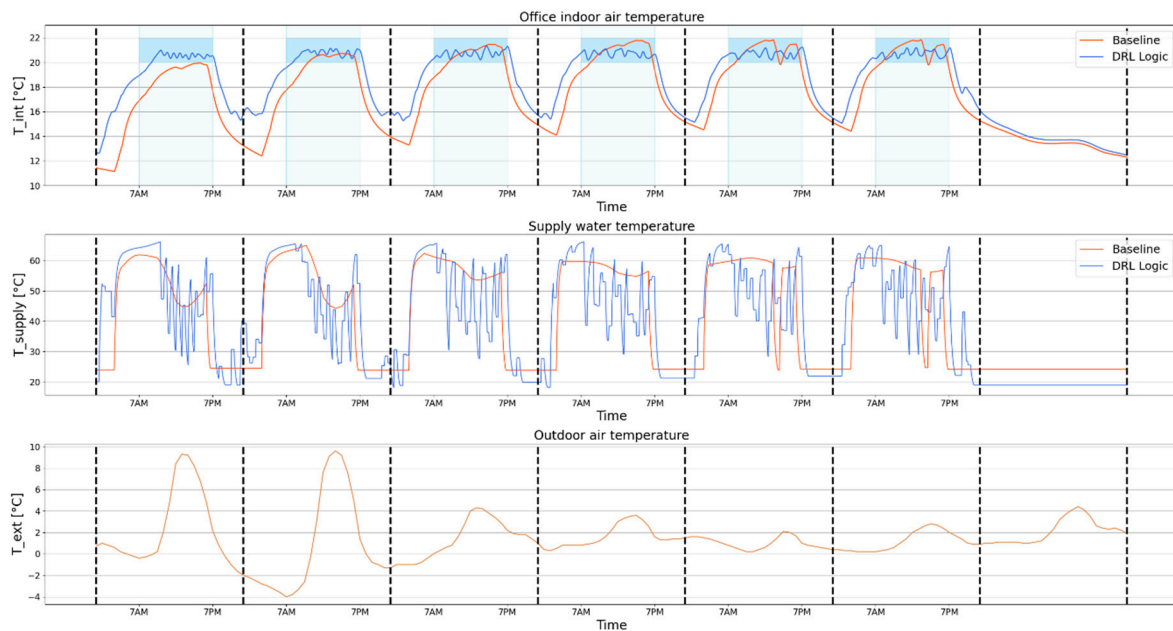


Figure 10. Comparison between dynamically deployed SAC control agent and baseline controller during a week for the Scenario 1.

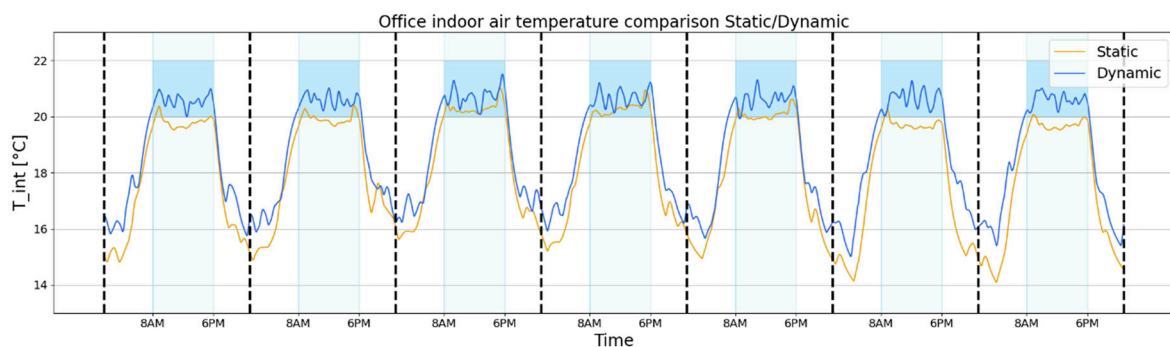


Figure 11. Comparison between statically and dynamically deployed agent in terms of indoor temperature profiles during a week for the Scenario 5.

Figure 11 compares the indoor temperature patterns resulting from the statically and dynamically deployed agents during a week in the scenario S5. In this scenario, the building is also occupied on Sundays, and the occupancy period is restricted to between 8 AM and 6 PM for all weekdays. The static agent is not able to handle the change of occupancy schedules, and therefore to change the control policy learned during the online training phase. As reported in Table 6, the cumulative sum of temperature violation is much higher than that of the agent deployed dynamically.

On the contrary, the dynamically deployed agent is able to learn the change of occupancy patterns and optimises the control policy. Then, the required indoor temperature conditions during the occupancy are satisfied and temperature violations on all weekdays are reduced.

To conclude the scenario analysis, the indoor temperature profile and the supply water temperature in Scenario S7 and S8 are compared for the statically and dynamically

deployed agents, as shown in Figure 12, in order to demonstrate the flexibility of the agent when opaque envelope characteristics are modified.

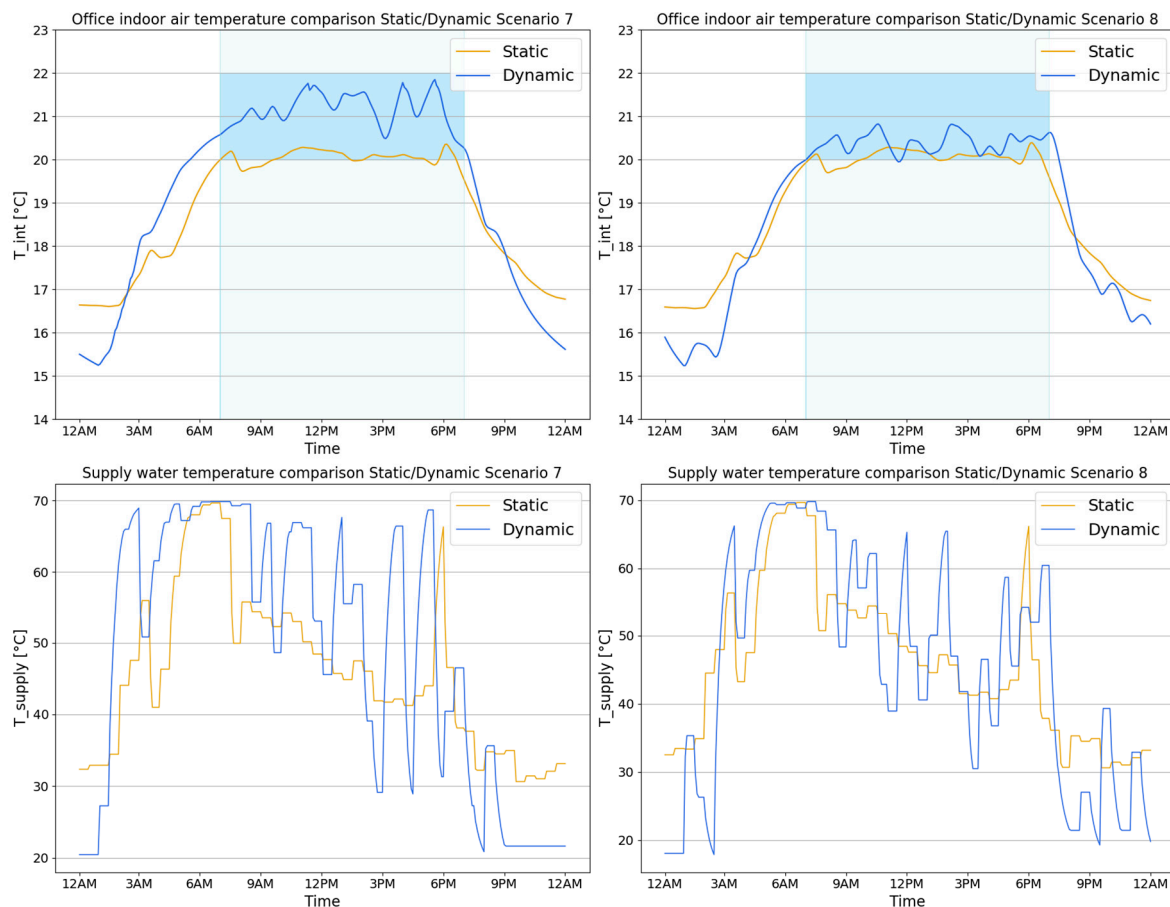


Figure 12. Comparison between statically and dynamically deployed agent in terms of indoor temperature and supply water temperature profiles during a typical day in Scenarios 7 and 8.

In scenarios S7 and S8, the stratigraphy of the vertical walls was modified introducing respectively a massive and a light curtain wall. These changes aim to highlight the differences related to the internal heat capacity value considering that the thermal transmittance value were kept almost equal for the two cases. The adoption of a massive wall for vertical walls provides a significant internal thermal capacity than the light curtain wall, as can be seen in Figure 6.

As stated previously, the static agent is not able to handle the change in boundary conditions, as demonstrated by the analysis of indoor temperature and supply water temperature profiles, which are identical in Scenario 7 and 8.

On the other hand, the dynamically deployed agent shows excellent flexibility with respect to changing the opaque envelope features. The agent was able to exploit the charging/discharging of thermal mass in both scenarios, anticipating the switch-ON time in the massive wall case. At the same time, the presence of a massive wall involves an advantage in terms of the heat accumulated. It makes it possible to avoid the system working until the end of the occupancy period, ensuring at the same time the required indoor temperature conditions.

The heating system is switched-ON about two hours in advance in the massive wall to allow an adequate charge of the mass and switched-OFF two hours in advance in order to better exploit the discharged heat. For both scenarios, the dynamically deployed agent allows the indoor temperature conditions required for the environment to be met, thus improving the performance compared to the static counterpart.

7. Discussion

This paper focused on the development of an adaptive model-free control strategy for a heating system that serves a real office building, whose current implemented control strategy consists in a climatic rule based one. This case study has particular importance, since the majority of non-residential buildings in Italy are equipped with this type of heating system.

The goal of this work was to train a DRL control agent online to maintain adequate temperature conditions during the occupied period and reduce the heating energy demand. For the developed controller, it was used a DRL algorithm based on a continuous action-space, named Soft Actor-Critic.

The DRL controller was deployed, for a calibrated energy model, in a simulation environment coupling EnergyPlus and Python.

It was chosen to perform an online training to emphasise the model-free nature of reinforcement learning. The online training of the DRL agent was able to guarantee performance improvements compared to the baseline, mainly in terms of indoor temperature control.

A sensitivity analysis was carried out during the online training phase, due to the strong dependence of the DRL controller performance on hyperparameters. The absence of guidelines providing support for the immediate choice of the hyperparameters of the algorithm implied the necessity of conducting this kind of analysis. As a consequence, despite the model-free nature of reinforcement learning, a modelling effort was required to analyse the impact of the hyperparameters in the performance of the controller.

The online training of the DRL controller may imply the occurrence of a discomfort period in the early learning stages, depending on the quickness at which the control agent learns an optimal control strategy. To obtain optimal performance faster, two hyperparameters of the algorithm, (i.e., gradient steps and learning rate) were assumed to be decreasing piecewise functions, in order to ensure higher values at the beginning of the simulation and guaranteeing the speeding up of the learning process. This strategy led to performances similar to those obtained with the RBC baseline after two weeks in the simulation period. As the simulation proceeded, these parameters were reduced in order to allow the agent to learn by interacting with the environment, avoiding over-exploration that could have caused deviation from the optimal control policy.

The best online training of DRL agent was then used for a subsequent scenario analysis, useful for testing the flexibility of the SAC control logic for the same period (including December, January and February), changing some boundary conditions, such as external weather conditions, occupancy schedules, structural conditions and indoor comfort conditions. These changes were grouped into eight different deployment scenarios.

During the scenario analysis, the higher heating energy demand associated with the dynamically deployed agent was justified by the significant improvement obtained in the indoor temperature conditions compared to the statically deployed agent and the baseline controller.

Compared to the static case, a dynamic deployment needs more computational time to complete the simulation due to the online update of the control strategy obtained from the training phase. The obtained results suggest that this kind of deployment is better able to handle changes in the controlled environment, highlighting its high potential for providing adaptive control policy. However, dynamic deployed DRL agents are more exposed to control instability issues that can compromise the identification of an optimal policy. For this reason, in this study, a proper analysis for the setting of agent hyperparameters was performed, thus also preventing the occurrence of significant deviations from the optimal control strategy during dynamic deployment.

8. Conclusions and Future Works

In this paper, the online implementation of an agent based on Soft Actor-Critic was evaluated for the control of a water-based heating system serving an office building. It

was necessary to use a simulation environment to test its performance on a calibrated energy model.

After performing a sensitivity analysis on the hyperparameters of the SAC algorithm during the online training phase, the best agent was selected. In particular, it was able to reduce the cumulative sum of temperature violations by more than 60% with respect to the baseline controller.

A scenario analysis was then performed to test the flexibility and adaptability of the controller to the variation of different boundary conditions grouped within eight deployment scenarios. For this purpose, the agent was tested both statically and dynamically.

The best DRL agent was more effective when deployed dynamically in terms of indoor temperature control, with a reduction of the cumulative sum of temperature violations of about 50% and 75% in comparison to static deployment and rule-based control respectively. However, the dynamically deployed agent led to a heating energy consumption about 7% and 11% higher than the statically deployed agent and the baseline respectively. This trade-off is still reasonable, considering that the building's indoor temperature conditions were remarkably improved when considering a dynamic deployment of the agent.

Future works will be focused on the following aspects:

- The developed controller will be implemented in the real world, although switching from simulations to implementation is very complicated and still represents one of the main challenges, especially concerning infrastructure to make the controller available. The building's heating energy requirements were evaluated using building energy simulation software (Energy Plus), assuming that the air was in 'perfect mixing' conditions. However, during the real implementation, the location of the indoor temperature sensors should be carefully defined, considering that the perfect mixing hypothesis could be not verified.
- The SAC control logic will be tested with more modern HVAC systems, including renewable energy sources and energy storage.
- The indoor thermal comfort will be evaluated by means of Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD), including them in the objective function [40,41].
- The aspects of reproducibility and standardisation will be further analysed, as the control agent implemented in this case study could perform differently for different HVAC systems or for buildings located in other climatic zones.

Author Contributions: Conceptualisation, S.B. and A.C.; Data curation, D.C. and S.B.; Formal analysis, D.C. and S.B.; Investigation, M.S.P., S.B. and A.C.; Methodology, D.C., S.B., M.S.P. and A.C.; Project administration, A.C.; Software, D.C. and S.B.; Supervision, A.C.; Validation, M.S.P., S.B. and A.C.; Writing—Original draft, D.C.; Writing—Review & editing, M.S.P., S.B. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The work of Silvio Brandi was carried out in the context of a scholarship at Politecnico di Torino funded by Enerbrain s.r.l. The authors acknowledge Enerbrain s.r.l. for providing information on the case study analysed in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

DRL	Deep Reinforcement Learning
HVAC	Heating, Ventilation and Air Conditioning
ML	Machine Learning
MPC	Model Predictive Control

NN	Neural Network
OCC	Occupant-Centric Controller
PID	Proportional-Integrative-Derivative
PMV	Predicted Mean Vote
PPD	Percentage of Person Dissatisfied
RBC	Rule-Based Control
RL	Reinforcement Learning
SAC	Soft Actor-Critic
α	Entropy regularisation coefficient
β	Temperature term weight of reward function
χ_i	Internal heat capacity
δ	Energy term weight of reward function
γ	Reward discount factor
μ	Learning rate
π	Control policy
A_t	Control action at control time step t
$C_v(\text{RMSE})$	Coefficient of Variation of the Root Mean Square Error
E_{HEAT}	Energy heating consumption at control time step t
g	Solar heat gain coefficient
H	Shannon entropy term
q	Action-value function
MBE	Mean Bias Error
Q_{SUPP}	Supply boiler heating energy
r_t	Reward at control time step t
r_{t+1}	Reward at control time step $t+1$
r_E	Energy term of reward function
r_T	Temperature term of reward function
$r_{T, OCC=1}$	Temperature term of reward function when occupants are present
SP_{INT}	Indoor air temperature setpoint
S_t	Environment state at control time step t
S_{t+1}	Environment state at control time step $t+1$
$t_{end, occ}$	Time at which occupants leave the building
T_{INT}	Indoor air temperature
T_{LOW}	Lower threshold limit of temperature comfort range
T_{OUT}	Outdoor air temperature
T_{RET}	Return water temperature
T_{SUPP}	Supply water temperature
T_{UPP}	Upper threshold limit of temperature comfort range
U	Thermal transmittance
v	State-value function

References

1. Yang, L.; Nagy, Z.K.; Goffin, P.; Schlueter, A. Reinforcement learning for optimal control of low exergy buildings. *Appl. Energy* **2015**, *156*, 577–586. [[CrossRef](#)]
2. Parliament of the European Union. Directive 2018/844/EU of the European Parliament and of the Council of 30 May 2018, amending directives 2010/31/EU on the energy performance of buildings and directive 2012/27/EU on Energy Efficiency. *J. Eur. Commun.* **2018**, *156*, 75–91.
3. Wang, Z.; Hong, T. Reinforcement learning for building controls: The opportunities and challenges. *Appl. Energy* **2020**, *269*, 115036. [[CrossRef](#)]
4. Martinopoulos, G.; Papakostas, K.T.; Papadopoulos, A.M. A comparative review of heating systems in EU countries, based on efficiency and fuel cost. *Renew. Sustain. Energy Rev.* **2018**, *90*, 687–699. [[CrossRef](#)]
5. Zhang, Z.; Chong, A.; Pan, Y.; Zhang, C.; Lam, K.P. Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning. *Energy Build.* **2019**, *199*, 472–490. [[CrossRef](#)]
6. Finck, C.; Beagon, P.; Clauß, J.; Péan, T.; Vogler-Finck, P.; Zhang, K.; Kazmi, H. Review of applied and tested control possibilities for energy flexibility in buildings. *IEA EBC Annex 67* **2018**, 1–59. [[CrossRef](#)]
7. Clauß, J.; Finck, C.; Vogler-Finck, P.; Beagon, P. Control strategies for building energy systems to unlock demand side flexibility—A review. In Proceedings of the 15th International Building Performance Simulation Association, San Francisco, CA, USA, 7–9 August 2017.

8. Geng, G.; Geary, G. On performance and tuning of PID controllers in HVAC systems. In Proceedings of the IEEE International Conference on Control and Applications, Vancouver, BC, Canada, 13–16 September 1993; IEEE: New York, NY, USA, 2002; Volume 2, pp. 819–824.
9. Afram, A.; Janabi-Sharifi, F. Theory and applications of HVAC control systems—A review of model predictive control (MPC). *Build. Environ.* **2014**, *72*, 343–355. [[CrossRef](#)]
10. Aghemo, C.; Virgone, J.; Fracastoro, G.; Pellegrino, A.; Blaso, L.; Savoyat, J.; Johannes, K. Management and monitoring of public buildings through ICT based systems: Control rules for energy saving with lighting and HVAC services. *Front. Arch. Res.* **2013**, *2*, 147–161. [[CrossRef](#)]
11. Mechri, H.E.; Capozzoli, A.; Corrado, V. USE of the ANOVA approach for sensitive building energy design. *Appl. Energy* **2010**, *87*, 3073–3083. [[CrossRef](#)]
12. Dragoña, J.; Arroyo, J.; Figueroa, I.C.; Blum, D.; Arendt, K.; Kim, D.; Ollé, E.P.; Oravec, J.; Wetter, M.; Vrabie, D.L.; et al. All you need to know about model predictive control for buildings. *Annu. Rev. Control.* **2020**, *50*, 190–232. [[CrossRef](#)]
13. Ma, Y.; Borrelli, F.; Hancey, B.; Packard, A.K.; Bortoff, S.A. Model Predictive Control of thermal energy storage in building cooling systems. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC), Shanghai, China, 15–18 December 2009; IEEE: New York, NY, USA, 2009; pp. 392–397.
14. Oldewurtel, F.; Parisio, A.; Jones, C.N.; Gyalistras, D.; Gwerder, M.; Stauch, V.; Lehmann, B.; Morari, M. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy Build.* **2012**, *45*, 15–27. [[CrossRef](#)]
15. Sturzenegger, D.; Gyalistras, D.; Morari, M.; Smith, R.S. Model Predictive Climate Control of a Swiss Office Building: Implementation, Results, and Cost–Benefit Analysis. *IEEE Trans. Control. Syst. Technol.* **2016**, *24*, 1–12. [[CrossRef](#)]
16. Paris, B.; Eynard, J.; Grieu, S.; Talbert, T.; Polit, M. Heating control schemes for energy management in buildings. *Energy Build.* **2010**, *42*, 1908–1917. [[CrossRef](#)]
17. Prívarva, S.; Široký, J.; Ferkl, L.; Cigler, J. Model predictive control of a building heating system: The first experience. *Energy Build.* **2011**, *43*, 564–572. [[CrossRef](#)]
18. Kontes, G.D.; Giannakis, G.I.; Sánchez, V.; De Agustin-Camacho, P.; Romero-Amorrortu, A.; Panagiotidou, N.; Rovas, D.V.; Steiger, S.; Mutschler, C.; Gruen, G. Simulation-Based Evaluation and Optimization of Control Strategies in Buildings. *Energies* **2018**, *11*, 3376. [[CrossRef](#)]
19. Hong, T.; Wang, Z.; Luo, X.; Zhang, W. State-of-the-art on research and applications of machine learning in the building life cycle. *Energy Build.* **2020**, *212*, 109831. [[CrossRef](#)]
20. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, UK, 2012.
21. Huys, Q.J.M.; Cruickshank, A.; Serès, P. Reward-Based Learning, Model-Based and Model-Free. In *Encyclopedia of Computational Neuroscience*; Springer International Publishing: Cham, Switzerland, 2014; pp. 1–10.
22. Watkins, C.J.C.H.; Dayan, P. Q-Learning. *Mach. Learn.* **1992**, *8*, 279–292. [[CrossRef](#)]
23. Peng, J.; Williams, R.J. Incremental Multi-Step Q-Learning. *Mach. Learn. Proc.* **1994**, 226–232. [[CrossRef](#)]
24. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [[CrossRef](#)]
25. Ahn, K.U.; Park, C.S. Application of deep Q-networks for model-free optimal control balancing between different HVAC systems. *Sci. Technol. Built Environ.* **2020**, *26*, 61–74. [[CrossRef](#)]
26. Brandi, S.; Piscitelli, M.S.; Martellacci, M.; Capozzoli, A. Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings. *Energy Build.* **2020**, *224*, 110225. [[CrossRef](#)]
27. Vázquez-Canteli, J.; Kämpf, J.; Nagy, Z. Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration. *Energy Procedia* **2017**, *122*, 415–420. [[CrossRef](#)]
28. Vázquez-Canteli, J.R.; Ulyanin, S.; Kämpf, J.; Nagy, Z. Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities. *Sustain. Cities Soc.* **2019**, *45*, 243–257. [[CrossRef](#)]
29. Yu, Z.; Dexter, A. Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning. *Control. Eng. Pr.* **2010**, *18*, 532–539. [[CrossRef](#)]
30. Yoon, Y.R.; Moon, H.J. Performance based thermal comfort control (PTCC) using deep reinforcement learning for space cooling. *Energy Build.* **2019**, *203*, 109420. [[CrossRef](#)]
31. Jia, R.; Jin, M.; Sun, K.; Hong, T.; Spanos, C. Advanced Building Control via Deep Reinforcement Learning. *Energy Procedia* **2019**, *158*, 6158–6163. [[CrossRef](#)]
32. Park, J.Y.; Nagy, Z. HVACLearn: A reinforcement learning based occupant-centric control for thermostat set-points. In Proceedings of the Eleventh ACM International Conference on Future Energy Systems (e-Energy '20), Virtual Event, Melbourne, Australia, 22–26 June 2020; Association for Computing Machinery (ACM): New York, NY, USA, 2020; pp. 434–437.
33. Lu, S.; Wang, W.; Lin, C.; Hameen, E.C. Data-driven simulation of a thermal comfort-based temperature set-point control with ASHRAE RP884. *Build. Environ.* **2019**, *156*, 137–146. [[CrossRef](#)]
34. Wang, Y.; Velswamy, K.; Huang, B. A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems. *Processes* **2017**, *5*, 46. [[CrossRef](#)]
35. Zou, Z.; Yu, X.; Ergun, S. Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. *Build. Environ.* **2020**, *168*, 106535. [[CrossRef](#)]
36. Qiu, S.; Li, Z.; Li, Z.; Li, J.; Long, S.; Li, X. Model-free control method based on reinforcement learning for building cooling water systems: Validation by measured data-based simulation. *Energy Build.* **2020**, *218*, 110055. [[CrossRef](#)]

37. Costanzo, G.; Iacovella, S.; Ruelens, F.; Leurs, T.; Claessens, B. Experimental analysis of data-driven control for a building heating system. *Sustain. Energy Grids Netw.* **2016**, *6*, 81–90. [[CrossRef](#)]
38. Crawley, D.B.; Pedersen, C.O.; Lawrie, L.K.; Winkelmann, F.C. EnergyPlus: Energy Simulation Program. *ASHRAE J.* **2000**, *42*, 49–56.
39. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:1603.04467v2.
40. European Committee for Standardization. *EN Standard 16798-1. Indoor Environmental Input Parameters for Design and Assessment of Energy Performance of Buildings Addressing Indoor Air Quality, Thermal Environment, Lighting and Acoustics—Module M1-6*; European Committee for Standardization: Brussels, Belgium, 2019.
41. American Society of Heating, Ventilating, and Air Conditioning Engineers (ASHRAE). *ANSI/ASHRAE Standard 55. Thermal Environmental Conditions for Human Occupancy*; American Society of Heating, Ventilating, and Air Conditioning Engineers: Atlanta, GA, USA, 2017.
42. Alfano, F.R.D.; Olesen, B.W.; Palella, B.I.; Pepe, D.; Riccio, G. Fifty Years of PMV Model: Reliability, Implementation and Design of Software for Its Calculation. *Atmosphere* **2019**, *11*, 49. [[CrossRef](#)]
43. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft Actor-Critic Algorithms and Applications. *arXiv* **2018**, arXiv:1812.05905.
44. Han, M.; May, R.; Zhang, X.; Wang, X.; Pan, S.; Yan, D.; Jin, Y.; Xu, L. A review of reinforcement learning methodologies for controlling occupant comfort in buildings. *Sustain. Cities Soc.* **2019**, *51*. [[CrossRef](#)]
45. Gullapalli, V. A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural Netw.* **1990**, *3*, 671–692. [[CrossRef](#)]
46. Azuatalam, D.; Lee, W.-L.; De Nijs, F.; Liebman, A. Reinforcement learning for whole-building HVAC control and demand response. *Energy AI* **2020**, *2*, 100020. [[CrossRef](#)]
47. Bellman, R.E. Dynamic Programming. *Science* **1966**, *153*, 34–37. [[CrossRef](#)]
48. Palladino, D.; Nardi, I.; Buratti, C. Artificial Neural Network for the Thermal Comfort Index Prediction: Development of a New Simplified Algorithm. *Energies* **2020**, *13*, 4500. [[CrossRef](#)]
49. Yao, J.; Xu, J. Research on the BPNN in the Prediction of PMV. *Appl. Mech. Mater.* **2010**, *29*, 2804–2808. [[CrossRef](#)]
50. Anand, P.; Cheong, D.; Sekhar, C.; Santamouris, M.; Kondepudi, S. Energy saving estimation for plug and lighting load using occupancy analysis. *Renew. Energy* **2019**, *143*, 1143–1161. [[CrossRef](#)]
51. Antonopoulos, I.; Robu, V.; Couraud, B.; Kirli, D.; Norbu, S.; Kiprakis, A.; Flynn, D.; Elizondo-Gonzalez, S.; Wattam, S. Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review. *Renew. Sustain. Energy Rev.* **2020**, *130*, 109899. [[CrossRef](#)]
52. Pinto, G.; Brandi, S.; Capozzoli, A.; Vázquez-Canteli, J.; Nagy, Z. Towards Coordinated Energy Management in Buildings via Deep Reinforcement Learning. In Proceedings of the 15th SDEWES Conference, Cologne, Germany, 1–5 September 2020.
53. Monetti, V.; Davin, E.; Fabrizio, E.; André, P.; Filippi, M. Calibration of Building Energy Simulation Models Based on Optimization: A Case Study. *Energy Procedia* **2015**, *78*, 2971–2976. [[CrossRef](#)]
54. Martínez, S.; Eguía, P.; Granada, E.; Moazami, A.; Hamdy, M. A Performance Comparison of Multi-Objective Optimization-based approaches for Calibrating White-box Building Energy Models. *Energy Build.* **2020**, *216*, 109942. [[CrossRef](#)]
55. American Society of Heating, Ventilating, and Air Conditioning Engineers (ASHRAE). *Guideline 14-2014, Measurement of Energy and Demand Savings*; Technical Report; American Society of Heating, Ventilating, and Air Conditioning Engineers: Atlanta, GA, USA, 2014.
56. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540.