Building Technologies & Urban Systems Division
Energy Technologies Area
Lawrence Berkeley National Laboratory

Online transfer learning strategy for enhancing the scalability and deployment of deep reinforcement learning control in smart buildings

Davide Coraci[1], Silvio Brandi[1], Tianzhen Hong[2], Alfonso Capozzoli[1]

[1]Politecnico di Torino, Department of Energy, TEBE research group, BAEDA Lab
[2]Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory

Energy Technologies Area
January 2023

DOI: 10.1016/j.apenergy.2022.120598

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the US Department of Energy under Contract No. DE-AC02-05CH11231.

Highlights

Online transfer learning strategy for enhancing the scalability and deployment of deep reinforcement learning control in smart buildings

Davide Coraci, Silvio Brandi, Tianzhen Hong, Alfonso Capozzoli

- Transfer learning enhances the scalability of DRL controllers in buildings
- The online transfer learning outperforms RBC and online DRL controller
- Online transfer learning does not require modeling effort compared to offline DRL

Online transfer learning strategy for enhancing the scalability and deployment of deep reinforcement learning control in smart buildings

Davide Coraci[a], Silvio Brandi[a], Tianzhen Hong[b], Alfonso Capozzoli[a,*]

[a] *Politecnico di Torino, Department of Energy, TEBE research group, BAEDA Lab, Corso Duca degli Abruzzi 24 Torino, 10129, Italy*
[b] *Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory, One Cyclotron Road Berkeley, CA 94720, USA*

Abstract

In recent years, advanced control strategies based on Deep Reinforcement Learning (DRL) proved to be effective in optimizing the management of integrated energy systems in buildings, reducing energy costs and improving indoor comfort conditions when compared to traditional reactive controllers. However, the scalability and implementation of DRL controllers are still limited since they require a considerable amount of time before converging to a near-optimal solution. This issue is currently addressed in literature through the offline pre-training of the DRL agent. However this solution results in two main critical issues: (1) the need to develop a building surrogate model to perform the training task, and (2) the need to perform a fine-tuning process over several training episodes to obtain a near-optimal control policy.

In this context, this paper introduces an Online Transfer Learning (OTL) strategy that exploits two knowledge-sharing techniques, weight-initialization and imitation learning, to transfer a DRL control policy from a source office building to various target buildings in a simulation environment coupling EnergyPlus and Python.

A DRL controller based on discrete Soft Actor-Critic (SAC) is trained on the source building to manage the operation of a cooling system consisting of a chiller and a thermal storage. Several target buildings are defined to benchmark the performance of the OTL strategy with that of a Rule-Based

*Corresponding author
Email address: alfonso.capozzoli@polito.it (Alfonso Capozzoli)

Controller (RBC) and two DRL-based control strategies, deployed in offline and online fashion. The strategy adopted for OTL emulates the real world implementation with a simulation process by implementing the transferred DRL agent for a single episode in the target buildings. Target buildings have the same geometrical features and are served by the same energy system as the source building, but differ in terms of weather conditions, electricity price schedules, occupancy patterns, and building envelope efficiency levels. The results show that the OTL strategy can reduce the cumulated sum of temperature violations on average by 50 % and 80 % respectively when compared to RBC and online DRL while enhancing the energy system operation with electricity cost savings ranging between 20 % and 40 %. The OTL agent performs slightly worse than the offline DRL controller but it does not require any modeling effort and can be implemented directly on target buildings emulating a real-world implementation.

Keywords: online transfer learning, homogeneous transfer learning, intra-agent transfer learning, building adaptive control, deep reinforcement learning, energy efficiency

---

Nomenclature

α	Boltzmann temperature coefficient
β	Temperature term weight of reward function
χ_i	Internal heat capacity [kJ/m^{2}K]
δ	Electricity cost term weight of reward function
γ	Discount factor
μ	Learning rate
A_t	Control action at control time step t
c_E	Electricity buying price [€/kWh]
D_S	Source domain
D_T	Target domain

$E_{CHILLER}$ Chiller energy consumption [kWh]

E_{cost} Electricity cost [€]

E_{PUMP} Circulation pumps energy consumption [kWh]

$f(\cdot)$ Objective predictive function

g Solar heat gain coefficient

Q_{cap} Capacity of chiller [kW]

R_E Electricity cost term of reward function

R_T Temperature term of reward function

r_t Reward at control time step t

RBC_{CF} Rule-based controller part choosing whether to supply cooling energy to the building

RBC_{OM} Rule-based controller part choosing the operation mode of the energy system

S_{t+1} Environment state at control time step t+1

S_t Environment state at control time step t

SOC_{TES} State-Of-Charge of the water storage

SP_{INT} Indoor air temperature setpoint [°C]

T_{ch} Chiller supply temperature [°C]

T_{INT} Indoor air temperature [°C]

T_{LOW} Lower threshold limit of temperature comfort range [°C]

$T_{s,max}$ Storage temperature upper boundary [°C]

$T_{s,min}$ Storage temperature lower boundary [°C]

T_S Source task

T_T Target task

T_{UPP} Upper threshold limit of temperature comfort range [°C]

T_{viol} Temperature violation [°C]

T_{wxyz} Target building configuration code

U_{OP} Thermal transmittance of the opaque envelope [W/m^{2}K]

U_{TR} Thermal transmittance of the transparent envelope [W/m^{2}K]

Acronyms

AHUs Air Handling Units

BESS Battery Energy Storage System

BCVTB Building Control Virtual Test Bed

CF Cooling Fraction

COP Coefficient of Performance

DDPG Deep Deterministic Policy Gradient

DNNs Deep Neural Networks

DRL Deep Reinforcement Learning

HVAC Heating, Ventilation and Air Conditioning

IES Integrated Energy Systems

IL Imitation Learning

KPIs Key Performance Indicators

LfD Learning from Demonstration

MILP Mixed-Integer Linear Programming

ML Machine Learning

MPC Model Predictive Control

OM Operation Mode

OTL Online Transfer Learning

PID Proportional-Integrative-Derivative

PV Photovoltaic

RBC Rule-Based Controller

RES Renewable Energy Sources

RL Reinforcement Learning

SAC Soft-Actor-Critic

SC Self-Consumption

SOC State-Of-Charge

TES Thermal Energy Storage

TL Transfer Learning

TOU Time-Of-Use

TPE Tree-structured Parzen Estimator

VAV Variable Air Volume

1. Introduction

Building energy consumption currently amounts to approximately 40 % of global primary energy, of which more than 50 % is related to the use of Heating, Ventilation and Air Conditioning (HVAC) systems [1]. In this context, the introduction of advanced energy management strategies is required to support the widespread adoption of Integrated Energy Systems (IES), consisting of Renewable Energy Sources (RES), such as Photovoltaic (PV) system [2], and Battery Energy Storage System (BESS) that aims to improve the Self-Consumption (SC) of the energy produced on-site from PV [3]. Moreover, buildings can exploit other flexibility sources on the thermal side, such as Thermal Energy Storage (TES) and building thermal inertia, that allows to shift or curtail energy demands for HVAC. However, such IES requires appropriate management [4], due to the need of adapting their operation to exogenous factors continuously evolving, as weather conditions, occupancy patterns or electricity tariffs. Although the commonly implemented ON/OFF and Proportional-Integrative-Derivative (PID) control strategies in buildings can be easily implemented and exhibit robust operation, they are reactive and not capable to adapt to changes in the environment to be

controlled [5, 6]. To overcome such limitations, researchers have explored the application of advanced control strategies that enable the management of energy systems by optimizing multi-objective functions [7]. Among advanced controllers, Model Predictive Control (MPC) is the most widely investigated in recent applications, as it can automatically adapt to changes in boundary conditions thanks to its predictive capabilities [8, 9]. MPC has gained considerable attention in the building industry [10] since it has demonstrated a remarkable ability in managing energy systems to enhance indoor comfort conditions while reducing energy consumption [11, 12]. However, the generalized implementation of MPC in buildings fails to emerge as its operation relies on the definition of a model for the optimization of the control problem [13], which is a time-intensive process. As a result, MPC deployment is limited in the building industry [14]. In that context, Reinforcement Learning (RL) emerges as a promising technique due to its model-free and data-driven nature, as the agent directly learns the optimal control policy by interacting with the system through a trial-and-error approach [15]. One particular family of algorithms, named Deep Reinforcement Learning (DRL), couples RL with Deep Neural Networks (DNNs) [16]. In this framework, DNNs are employed to approximate RL policy functions and enable the resolution of real-world problems, which are complex and require the definition of a large number of states and actions to properly represent the control problem [17]. In the context of building energy management, RL-based controllers have been implemented in HVAC systems to regulate the fan speed [18, 19] or to manage the indoor temperature set-point [20, 21] and the supply water temperature at generation level [13, 22, 23]. Furthermore, RL exhibited excellent capabilities in managing thermal storage by controlling their temperature set-point [24] or charge/discharge process at single [16, 21] and multiple building scale [25, 26]. Certain applications in literature evaluate the use of an online training technique to emulate the direct implementation of RL controllers without offline pre-training. The online RL control strategy requires that the optimal control policy is learned while the system is actively controlled [7]. However, this strategy is inefficient since the initial performance of the controller is usually very poor and, as a consequence, a significant training time is required to interact with the environment and achieve a near-optimal control policy. Conversely, the strategy mostly explored in literature foresees an offline pre-training setup of the RL controller before its deployment. Such approach involves the definition of a building surrogate model that can be either data-driven [19] (i.e., building dynamics approximated by means

of neural networks) or physics-based [23], developed with modeling software such as Modelica [27] or EnergyPlus [28]. However, the offline pre-training of DRL controllers can not be performed in buildings with no available (e.g., new buildings) or limited amount of data [29], since a considerable amount of data is required to build the surrogate model of the building. It results that these control strategies are less scalable and generalizable, performing properly only for specific building configurations. In addition, the definition of a model is needed for each building to be controlled, as in the case of the MPC. To address this gap, Transfer Learning (TL) emerges as a promising technique for increasing the scalability of advanced controllers in buildings. TL is a Machine Learning (ML) method that allows the sharing of pre-acquired knowledge for a particular task (i.e., source task) in a different but related problem (i.e., target task) with similar or different domains [30]. The implementation of TL results in a dramatic reduction of the training time required by machine learning models to converge towards a near-optimal solution and is usually applied at the beginning of the training process in the target domain, mainly in the context of supervised machine learning applications [31]. Since ML algorithms suffer from some issues (e.g., the lack of data to adequately train the models) [32], training machine learning-based models to address different tasks (e.g., load forecasting) is challenging [33]. Initial applications of TL are related to image recognition [34, 35], game playing [36] and natural language processing [37, 38]. However, reusing previous knowledge from various sources could be beneficial in the context of smart buildings. Therefore, in recent years TL was implemented in smart buildings in the context of load prediction [39, 40, 41], occupancy detection and activity recognition [42, 43], building dynamics [44, 45, 46] and system control [47, 48].

In the next section, reference studies about the use of the TL for the sharing of the control policy in buildings are reviewed. Furthermore, the motivations and the novelty of the present contribution are provided.

1.1. Related works on TL applications for advanced controllers in buildings

Applications of TL to system control are limited if compared to the others investigated in the context of smart buildings and are mainly dated back to the last three years. Moreover, the implementation of TL in the control field mainly refers to agents based RL, as in the case of robotics [49] or automotive [50]. In the context of building system control, the implementation of TL provides multiple advantages, since it allows the transfer of information

between advanced controllers, easing the deployment of such algorithms that are commonly tailored for specific control problems and scaling up the application of such algorithms in buildings with a limited amount of historical data. In addition, TL techniques could lead to the online implementation of RL-based controllers, ensuring acceptable performance from the early stages of deployment. However, the current state of the art concerning TL applications for DRL controllers evaluates the performance of the controller by applying a fine-tuning process performed over several episodes on the target building, as highlighted in the few published papers related to the application of TL for sharing control policy in buildings. Lissa et al. [51] proposed a methodology based on TL to enable the sharing of a RL control policy operating on a HVAC system between different rooms in the same building. In particular, a series of experiments were carried out to evaluate the variance in RL performance compared to the case without TL, as a function of the geometrical and geographical differences of the various rooms, as well as the different sizes of the HVAC system. This approach improved the indoor comfort conditions by reducing the discomfort time of the occupants. A similar methodology was employed by Fang et al. [52] to investigate the cross temporal-spatial transferability of a DRL controller in a HVAC system consisting of a chiller and three Air Handling Units (AHUs) to enhance indoor temperature conditions while reducing energy consumption. In detail, the authors develop a TL methodology to assess the effect of the climate and the number of neural network layers on the knowledge sharing process. As a result, the transfer process is effective when the DRL agent is transferred between buildings located in similar climatic conditions, exhibiting better performance by sharing two of the five layers of the neural network that approximates the control policy in the source building. Furthermore, Xu et al. [53] evaluate TL performances when a RL control policy was transferred from source to target buildings in different climates and with different envelope features and HVAC configurations. This study is the only one in which the use of heterogeneous TL was evaluated since it assessed the possibility of transferring a control policy between buildings with different numbers of thermal zones. Zhang et al. [54] implemented a strategy to transfer a library of RL multi-agent control policies from a multi-zone source building to a target building. Before transferring the control policy, the authors designed a strategy to choose the best pre-trained RL policy among those obtained on the source building for the management of zone temperature setpoints in a Variable Air Volume (VAV) system. As a result, 40 % of the energy con-

sumed by the HVAC system was saved on the target building compared to the baseline controller and 50 % compared to the RL controller trained from scratch over 5000 episodes. Tsang et al. [55] developed a framework to share the DRL optimal control policy between agents managing the electrical devices in an autonomous household. In detail, dependent devices are grouped to avoid scalability issues and the source controller knowledge is shared with the devices in the same group to advise the control action choice. This approach ensured a reduction of the training time of target device controllers by around 25 %. Similarly, the transfer of a RL agent controlling appliances was investigated by Zhang et al. [56]. In this case, the use of TL resulted in a reduction of the RL controller training time on the target buildings, improving its performance from the early stages of implementation compared to the case without transfer. The RL control agent was trained on a benchmark home with the same number and type of appliances and then fine-tuned on the target buildings. TL was evaluated for transferring battery management control policies between similar buildings with an integrated energy system in [57]. In particular, the building similarity was assessed by considering K-shape clustering to group buildings according to their energy consumption patterns. The operation of batteries was planned using a RL controller and transferred to target buildings in the same cluster. This approach allowed to achieve performances in target buildings similar to those of a Mixed-Integer Linear Programming (MILP) controller in 10 days. To conclude, the policy transfer for RL controllers was evaluated at microgrid scale in [47, 48]. Fan et al. [47] have developed a methodology to transfer between microgrids a Deep Deterministic Policy Gradient (DDPG) controller, reducing the training time in target building to achieve a near-optimal control policy. The optimal control policy is learned during a training phase developed in the source building to reduce operating costs by learning an optimal scheduling microgrid strategy. Lissa et al. [48] proposed an intra-transfer learning method, named parallel transfer learning, which allowed knowledge to be shared between five different agents during their training process without waiting until the end. This transfer approach was implemented in a microgrid with five homes, each with its energy system consisting of a PV system and a heat pump, and with its DRL controller managing the heat pump for minimizing energy costs. As a result, training time was reduced by a factor of 5 and energy savings of 10 % were achieved compared to the case without transfer.

1.2. Novelty and contributions of the paper

A fundamental gap emerged from the analysis of the current scientific literature about the application of TL to control policies in buildings. Analyzed applications evaluate the performance of a transferred control policy in a target building by applying a fine-tuning process over multiple episodes. In typical energy and building applications one episode usually represents an entire season (either heating or cooling) if not even a whole year. Thus, following this approach in a real-world context, a transferred control agent could still require multiple episodes before converging to acceptable solution which could translate in several years of implementation. In order to effectively enhance the scalability of DRL controllers in buildings it is desirable that a transferred agent is capable to achieve acceptable performance shortly after its implementation in the target building.

Therefore, it is required to develop an Online Transfer Learning (OTL) approach capable to rapidly perform the fine-tuning of the control policy pre-trained in the source building while its already implemented in the target building. A similar approach was previously exploited only to transfer supervised learning models. In [58], the authors shared the knowledge of a pre-trained offline classifier on a source scenario to different targets. In [59], the authors transferred a classification and regression model to predict the building dynamics. To the best of our knowledge, an online transfer learning strategy was not yet explored in the context of building control policy transfer.

Following these considerations, the present paper proposes an online transfer learning methodology to share the control policy of a DRL agent, based on a formulation of Soft-Actor-Critic (SAC) introduced by Christodoulou [60] capable of handling discrete action spaces. The SAC agent was firstly pre-trained on a source building to minimize electricity cost and enhance indoor temperature conditions.

Since the performance of DRL controllers is strongly influenced by hyperparameters, their implementation requires the definition of a method to obtain the optimal set of hyperparameters. Therefore, during the training phase of the DRL agent on source building an automated procedure was carried out to optimize the set of hyperparameters using Optuna [61]. Then, the best source DRL controller was transferred to several target buildings, derived from the source building by varying the weather conditions, electricity price schedules, occupancy schedules and building thermophysical properties.

The pre-trained control agent was implemented and fine-tuned in each target building through the proposed methodology. The proposed controller was benchmarked in terms of electricity cost and temperature violations against an online DRL controller and an offline pre-trained DRL controller. Moreover, an additional RBC strategy was introduced as a benchmark to provide a comparison with a traditional control strategy.

The experiments were carried out by means of a simulation environment combining EnergyPlus and Python, as in [13, 23]. According to the TL classifications discussed in [29, 30], the knowledge-sharing methodology developed in this paper is classified as homogeneous transductive transfer learning, as the transfer process is implemented in buildings where DRL controllers operate in a similar domain (i.e., same geometry and energy systems) and with an identical objective function. The knowledge sharing was performed exploiting a model parameter-based TL technique, named weight-initialization, since the target model weights are initialized using the pre-trained model weights. Furthermore, according to [31], our TL method is labelled as intra-agent transfer learning, since the target agents do not know the possible future implications of a new training process for the source agent after the knowledge sharing process. Based on the literature review on transfer learning of DRL controllers in buildings, the main innovative contributions of this paper can be summarised as follows:

- An online transfer learning strategy, based on homogeneous transductive TL, was developed to transfer a DRL controller pre-trained on a source office building to minimize electricity cost while enhancing indoor temperature conditions. The review of the literature shows that transfer learning approaches have been poorly explored for DRL control systems in buildings and the knowledge sharing approaches adopted have been rarely identified with respect to theoretical statements of transfer learning. Moreover, to the best of the authors' knowledge, an online transfer learning approach has not been explored in the framework of building control systems.

- A pre-trained DRL agent transferred on several target buildings and online deployed was benchmarked with other two DRL control strategies, offline and online deployed without any prior knowledge of the environment to be controlled. Moreover an additional RBC strategy was introduced to benchmark OTL performances. To the best of our knowledge, the implementation of transfer learning in the context of

control systems in buildings has not been compared yet with an online DRL strategy. The comparison among OTL and online DRL was the fairest to highlight the benefits of applying transfer learning. In fact the online DRL controller was developed to emulate the direct implementation of a controller that does not require the development of a simulation model of the controlled environment to perform pre-training.

- Several target building configurations were designed, where the best controller pre-trained on the source building was transferred to speed up the training process in target controllers. The source and target buildings differ in terms of weather conditions, electricity price schedules, occupancy schedules and building thermophysical properties, but have the same geometry and energy system. As a result, the effect of each variable on the performance of transfer learning can be independently quantified.

The rest of this paper is organized as follows. In Section 2 the theoretical foundations of DRL controllers and TL are described. Section 3 introduces the formulation of the control problem while Section 4 describes the methodological framework. Implementation details concerning source and target buildings, the online transfer learning and the controllers developed in this paper are provided in Section 5. Section 6 outlines the results obtained while Section 7 discusses them before providing conclusions and future directions in Section 8.

2. Methods

This section describes the methods adopted in this paper. However, only theoretical foundations regarding transfer learning are described in detail. Theoretical aspects concerning DRL and the discrete SAC algorithm applied in this paper can be found in [15, 60, 62, 63].

2.1. Transfer Learning

Transfer Learning is a machine learning method, emerging as a promising technique to reuse the knowledge acquired in a particular task for improving performances in a different but related problem [29]. Knowledge sharing happens at the beginning of the learning process, to accelerate the convergence process of machine learning models over the situation in which learning is performed from the beginning without prior knowledge. The mathematical

definition of TL requires the description of the concepts of domain and task, as described by [30]. Specifically, the domain consists of a feature space X and a marginal distribution probability $P(X)$, while the task consists of the label space Y and an objective predictive function $f(\cdot)$. This function is learned from the training data (represented by a pair (x_i,y_i)) and used to approximate the conditional probability $P(y|x)$ as well as to predict the label of new instances. The transfer process can occur between multiple domains, however research has focused on the case where knowledge sharing occurs between a source domain $D_S = (x_{S_1}, y_{S_1})$, ..., $(x_{S_{n_S}}, y_{S_{n_S}})$ and a target domain $D_T = (x_{T_1}, y_{T_1})$, ..., $(x_{T_{n_T}}, y_{T_{n_T}})$. Thus, according to [29, 30], TL is defined as the process that improves the learning of the predictive function in the target domain D_T with learning task T_T, using the acquired knowledge in the source domain D_S with task T_S. In general, domains and tasks can be the same or different. TL foresees that knowledge can be shared where source and target have different or similar domains, tasks and solutions. In this regard, according to [29], it is possible to identify some classifications concerning the similarity of tasks (i.e., label classification), features and labels (i.e., space classification), and knowledge sharing modalities (i.e., solution classification). The label classification splits into three categories TL depending on task similarity and label availability:

- inductive transfer learning, considering the availability of labelled data in source and target domains and that source and target tasks are different, without any interest in domain differences;
- transductive transfer learning, considering that source and target domains are different but with the same tasks. In this case labelled data are available only for source domain;
- unsupervised transfer learning, considering the unavailability of labelled data in source and target domains (that could be different or not) and different tasks between source and target.

The differences in source and target features (i.e., spaces) and labels are accounted within the space classification. In this case, TL is classified as homogeneous when source and target spaces and labels are identical. Otherwise, TL is classified as heterogeneous when spaces and/or labels differ between source and target. Moreover, TL is classified according to the knowledge sharing method adopted in solution classification: instance-based TL,

feature representation-based TL, relation knowledge-based TL and model parameter-based TL. Details about the first three categories are provided in Pinto et al. [29], being outside the scope of this work since it was implemented the model parameter-based TL. This kind of knowledge sharing is widely used for neural networks and involves the sharing of certain parameters or their distributions between source and target tasks, such as model weights. The model parameter-based TL can be divided into 3 sub-categories according to model parameter sharing modes:

- feature-extraction, where weights from pre-trained model are used for some layers that are not domain dependent and do not require further fine-tuning;
- weight-initialization, where the target model weights are initialized using the pre-trained model weights. In this case, an additional fine-tuning process could be performed;
- relational knowledge-based, considering the sharing of data relationship in case of similarity among the source and target datasets.

2.1.1. Transfer Learning for RL applications

In the context of RL, [31, 64, 65] give some indications about possible applications of TL for this algorithm. However, to classify TL applications for RL according to the previous categories, it is necessary to establish the correspondence between domain, label space and task defined for generic machine learning problems and state-space, action-space and reward function in the case of RL. In this case, the input feature space (i.e., domain) corresponds to the state-space in the RL framework, while the label space is equivalent to the RL action space. As indicated in [29], in RL the task corresponds to the combination of action space, reward function and transition function. Using this definition, RL applications can be categorized using labels and space classification for general machine learning problems. In detail, this paper explores the use of homogeneous transductive transfer learning, as the state and action spaces are the same between the source and target buildings, while the transition function changes due to differences between the domains in climatic conditions, electricity price schedules, occupancy schedules and building thermophysical properties. Furthermore, the knowledge sharing between RL agents is carried out considering model parameter-based TL: a

comparison between feature-extraction and weight-initialization is reported in Figure 1.

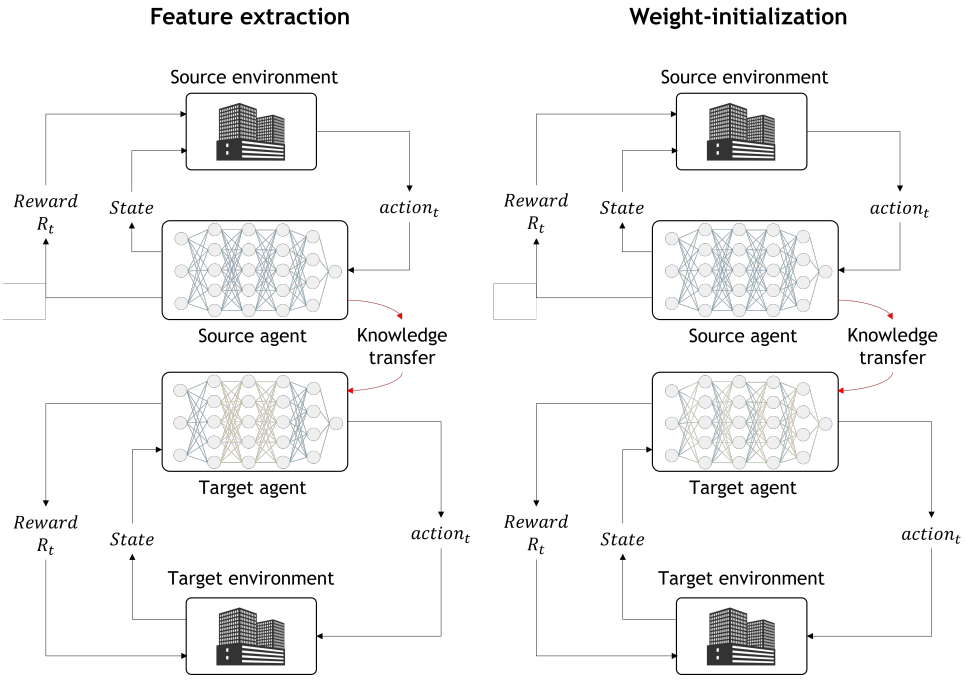

Figure 1: Model parameter-based TL for RL agents: comparison between feature-extraction and weight-initialization

In this paper, the weight-initialization TL method is employed as knowledge-sharing strategy between source and target agents. The differences associated with the source of knowledge, availability and required domain knowledge enable the identification of an additional classification for transfer learning. According to Da Silva and Costa [31], it is possible to define:

- Intra-Agent TL, representing transfer methods that do not require explicit communication for accessing to internal knowledge of the agents. In this case, it transferred the knowledge acquired by the source agent up to that moment (whether or not the training process has been completed) without the target agent knowing the possible future implications of new training process for the source agent.

- Inter-Agent TL, describing transfer methods to reuse the knowledge from communication with other agents. In this case, it transferred the knowledge already available from another agent at any time during the training process of the source agent, and the knowledge may be transferred bidirectionally, from target to source and vice-versa.

According to this classification, our work is categorized as Intra-Agent transfer learning, with the adoption of knowledge transfer named Mentor/Observer [31]: the target agent (i.e., observer) learns an optimal policy by observing the successful optimization of the control problem performed by the source controller (i.e., mentor). To conclude, [31, 65] indicate different settings of knowledge reuse in addition to transfer learning, such as:

- Imitation Learning (IL), where the agent in the target domain learns an optimal strategy on a particular task by observing an expert (e.g., RBC) that optimizes the same task. In this case, the target agent is aware of the transitions from the expert (and could store them in a buffer) but is not informed about the chosen set of actions and reward signals.

- Learning from Demonstration (LfD), similar to imitation learning, but in this case the expert controller could inform the target agent about the chosen set of actions and the target policy could be improved by accessing reward signals.

In this work, the source agent knowledge is reused by combining Transfer Learning and Imitation Learning, since the target agent policy is initialized using weights from pre-trained source policy, and the target agent buffer is initialized with the transition from RBC warm-up.

3. Control problem formulation

In this paper, homogeneous transfer learning is implemented to share the control policy between a source office building in Turin, Italy, and different target buildings. Source and target buildings are characterized by the same geometry and the same energy system. However, targets derived from the source building are located in different weather conditions and have different electricity price schedules, occupancy patterns and envelope features: further details are provided in Section 5.1. The proposed application is carried out

for a cooling season lasting 3 months (i.e., from 1 June to 29 August). The case study is conceived to benchmark the performance of transfer learning for a DRL-based controller system with that of RBC and two particular advanced control strategies, offline DRL and online DRL. The building is served by a cooling system consisting of an air-to-water chiller and a cold thermal storage acting as a buffer between the building and chiller. The energy system was modeled using EnergyPlus with available features for chiller and TES. The energy system provides cooling energy to the building through the electric chiller, operating at constant cold water temperature setpoint T_{ch}, or the TES, operating at constant cold water flow rate. The thermal storage operates between a minimum temperature $T_{s,min}$ and a maximum temperature $T_{s,max}$, which match respectively the maximum and minimum TES State-Of-Charge (SOC). The cooling energy is delivered to the environment using zone terminals, connected to the carrier fluid circuit in which the cold water flows by means of circulation pumps. Moreover, thermostatic control is considered in this case study, since the supply of energy to the building depends on indoor temperature conditions. A simplified scheme of the analyzed energy system is shown in Figure 2.

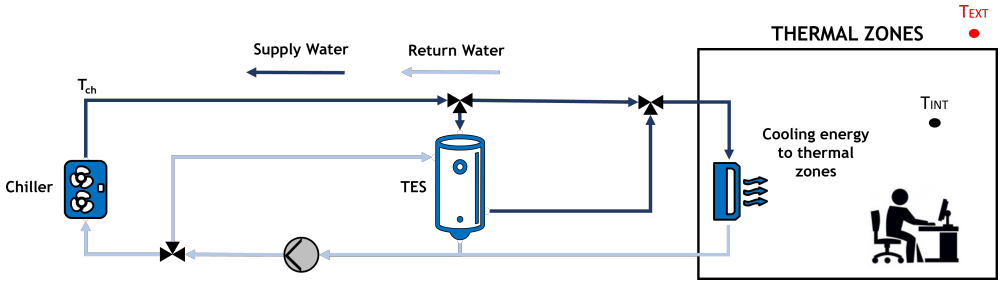

Figure 2: Simplified scheme of the cooling energy system

The DRL controller is developed to reduce electricity cost associated with the operation of the chiller and circulation auxiliary while maintaining the indoor temperature inside an acceptability range defined between [25, 27] °C, which corresponds to [-1, +1] from the desired indoor temperature setpoint of 26 °C by:

- optimal managing the cooling system by choosing between different operation modes as detailed below;

- choosing whether or not to supply cooling energy to the thermal zones.

The controller can manage the energy system according to three cooling operation modes, as shown in Figure 3:

1. **Discharging mode** (operation mode = -1), where the cooling energy required by the building is delivered by discharging the thermal storage. In this setting, the energy system operates at variable supply water temperature.
2. **Chiller mode** (operation mode = 0), where cooling energy is provided to the building exclusively by the chiller. In this operation mode, the energy system operates at constant supply water temperature.
3. **Charging mode** (operation mode = 1), where cooling energy is provided simultaneously to the storage and the building (if needed). In this setting, the energy system operates at constant supply water temperature.

The system operation modes are conceived to avoid that cooling energy is supplied to the building by both the chiller and the cold thermal energy storage simultaneously.

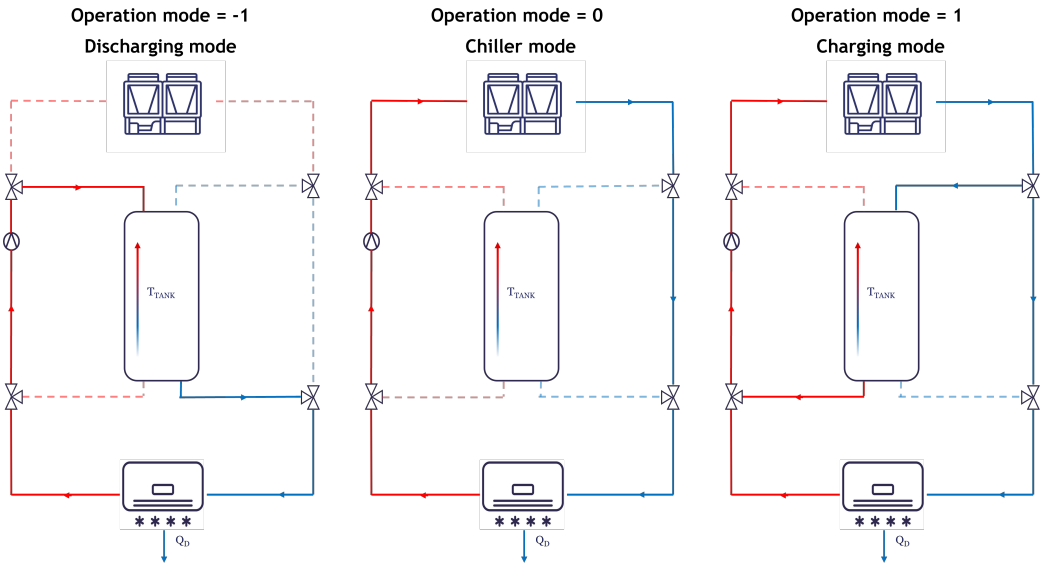

Figure 3: Operation modes of the cooling energy system

4. Methodology

This section outlines the methodological framework employed in this paper, offering insights concerning the TL process and the advanced controllers adopted to benchmark TL performance. The methodological process is organized into four stages, as shown in Figure 4.

4.1. Design of control problem

The first stage of the methodological process involves the development of the RBC and the DRL controller implemented in the source building.

4.1.1. Design of rule-based controller

The RBC is made up of two parts, one that chooses whether to supply cooling energy to the building (i.e., RBC_{CF}) and the other which decides the operating mode of the energy system (i.e., RBC_{OM}). These two agents are not independent, since the mode of operation of the energy system depends on whether cooling energy is delivered to the zone. Further details on the RBC design are provided in Section 5.3.

4.1.2. Design of DRL controller

DRL controller is developed to reduce electrical cost and maintain adequate indoor temperature conditions during occupancy hours. The development of the DRL controller involves the definition of its main components, i.e. the action space (which includes all possible actions to be selected by the control agent), the state space (containing all the observations required by the agent to optimize the control policy) and the reward function, intended to be representative of the control problem objective.

4.2. DRL training phase on source building

During the second methodological step, the DRL controller is trained on the source building in an offline manner. Details about this training method are provided in Subsection 4.4.1. During the DRL agent training process, an automated procedure is performed via Optuna [61] to find the optimal configuration of control algorithm hyperparameters since the performance of DRL controllers is considerably influenced by the choice of such variables. As a result, it is identified the best control agent among the analyzed controllers: this is employed during the next stage involving the transfer of the control policy to the target buildings. Further details on the DRL training phase are provided in Section 5.5.

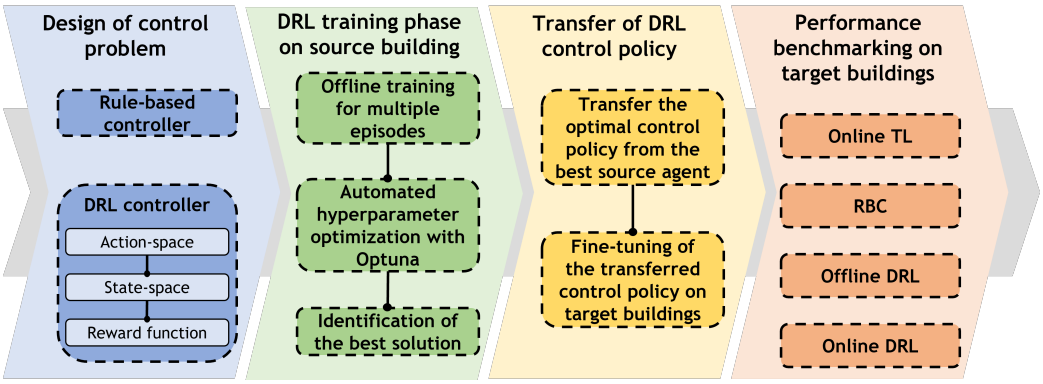

Figure 4: Methodological framework adopted in this work

4.3. Transfer of DRL control policy

The third step of the framework involves the development of a methodology for the implementation of TL for control policy sharing. In this way, it is shared the optimal control policy of the best agent trained on source building to the controllers to be implemented in the target buildings. The TL strategy employed in this paper is categorized as homogeneous transductive TL according to Section 2.1, since controllers implemented on the source and target buildings address the same task and the same state and action spaces but different probability distribution for state space (i.e., different domains). Furthermore, the transfer strategy employed is the weight-initialization as the knowledge transfer is linked to the sharing of neural network parameters between source and target controllers. In detail, the weights of Actor and Critic networks of the target controllers are initialized using the weights of the pre-trained source agent. Then, a fine-tuning process is performed to enable the updating of the neural network weights allowing the agent to adapt the control policy to the new conditions existing in the target buildings. However, TL approaches available in literature evaluate a fine-tuning process carried out in an offline manner, i.e., repeating this procedure several episodes consecutively. As a result, the scalability of the TL process is limited since it is required the development of a model for each target building to which source agent control policy will be transferred. Therefore, an online transfer learning methodology, described in Subsection 4.4.3, is developed to transfer the source optimal control policy.

4.4. Performance benchmarking on target buildings

In the last methodological step, a robust benchmark with two advanced strategies is provided for OTL with the offline and online (not transferred) DRL controllers which belong to the same family of advanced controllers. Moreover, the RBC was introduced to provide a benchmark with a traditional control strategy that is commonly implemented in real buildings.

The offline DRL strategy corresponds to that used during the DRL agent training phase on the source building (i.e., offline deep reinforcement learning), while the online strategy is designed to emulate the direct real-world DRL controller implementation without having any knowledge of the environment to be controlled. Further details about these DRL training strategies are provided in Subsections 4.4.1 and 4.4.2. The performances of these controllers are assessed during their implementation on different target buildings, derived from the source building by changing boundary conditions. In detail, nineteen target buildings are evaluated, accounting for different weather conditions, electricity price schedules, occupancy patterns, and envelope efficiencies (i.e., changing the thermophysical characteristics of the opaque and transparent envelope). Thus, a sensitivity analysis is performed to evaluate the effectiveness of TL as a function of the differences between the source and the target buildings. Further details on target building configurations are provided in Section 5.1.

4.4.1. Offline deep reinforcement learning

The offline training strategy for a DRL agent, shown in Figure 5 (c), foresees that the training period, named training episode, is repeated multiple times to ensure a stable control policy for the agent. However, this process exhibits a remarkable weakness although it guarantees a stable control policy: in case of changes in the environment to be controlled, controller retraining is required. This recursive training process is difficult to implement in practice, as it would require several episodes (e.g., corresponding to several cooling seasons in this study) before the agent would be able to upgrade the control policy, as well as a significant modeling effort to obtain a model of the building to be controlled.

4.4.2. Online deep reinforcement learning

The online DRL training strategy requires that the control agent converges to the optimal policy while actively controlling the system [7].

To imitate a direct real-time implementation, the training of the DRL agent is carried out on a single episode and not for several episodes as in the case of offline DRL. The advantage of this strategy relies on its model-free nature, as it is not necessary to generate a model of the building to be controlled. However, during the early stages of the training period the agent does not have any knowledge of the control problem and the risk that the actions chosen by the controller result in poor performance is significant. The memory buffer of the online DRL agent is initialized with the transitions obtained from the RBC operation (i.e., imitation learning). This procedure is detailed in the Subsection 4.4.3. Moreover, a number of gradient steps higher compared to the offline DRL agent is adopted to ease the exploration process and speed up the learning process after the first week of online DRL implementation [7].

However, a large number of gradient steps could involve the risk that the control agent converges to an optimal but deterministic control policy as the training process proceeds. To mitigate this issue, the value of the time step in which the learning process takes place is increased compared to the offline DRL strategy. A graphical representation of the online DRL strategy is shown in Figure 5 (b) and further details are given in Section 5.6.

4.4.3. Online transfer learning strategy

The strategy adopted for OTL emulates a real-time implementation as in the online DRL, but in this case the agent was pre-trained on the source building. Then, the agent was further fine-tuned on the new environmental conditions. The whole process is developed over a single episode. A representation of the implemented OTL strategy is shown in Figure 5 (a). In detail, the knowledge reuse approach is organized into two phases: imitation learning and transfer learning with weight-initialization. The agent pre-trained on the source building is transferred to initialize the target controller, but it does not operate during the imitation learning phase, performed during the first week of the analyzed period (i.e., from 1 June to 7 June), as the RBC is implemented. During this phase, the memory buffer of the OTL agent is initialized with transitions from RBC logical strategy described in Section 5.3. Transitions are stored in the memory buffer during each control time step. This process was found effective for enhancing the OTL agent in learning during the first days of deployment the relation between the chosen action, states (i.e., the evolution of the environment to be controlled), and reward function (i.e., the electricity cost associated with the operation of

energy system and the temperature violations).

As defined in Section 2.1, this knowledge reuse process is known as imitation learning. After this first phase, weights of the neural networks that approximate the DRL control policy are initialized with those of the source agent. Therefore, the controller is fine-tuned over the cooling season with a strategy guaranteeing that the target agent updates its control policy without completely overriding the pre-acquired knowledge in the source building that could be useful to the target controller. Thus, the value of the learning rate is decreased by half and the learning procedure is modified compared to the training phase of the source DRL controller. As shown in Figure 5 (a), the training period is alternated by steps in which the learning of the agent occurs or not. Concretely, a learning step is defined every n days, starting from the end of the warm-up period. Moreover, to avoid performance degradation for the controller, it is adopted a number of gradient steps higher compared to the offline DRL control strategy. The number of gradient steps denotes the number of batches extracted randomly from the buffer memory on which the gradient is updated at each control time step [7]. Detailed information about the values chosen for the typical parameters adopted for the OTL strategy is given in Section 5.6.

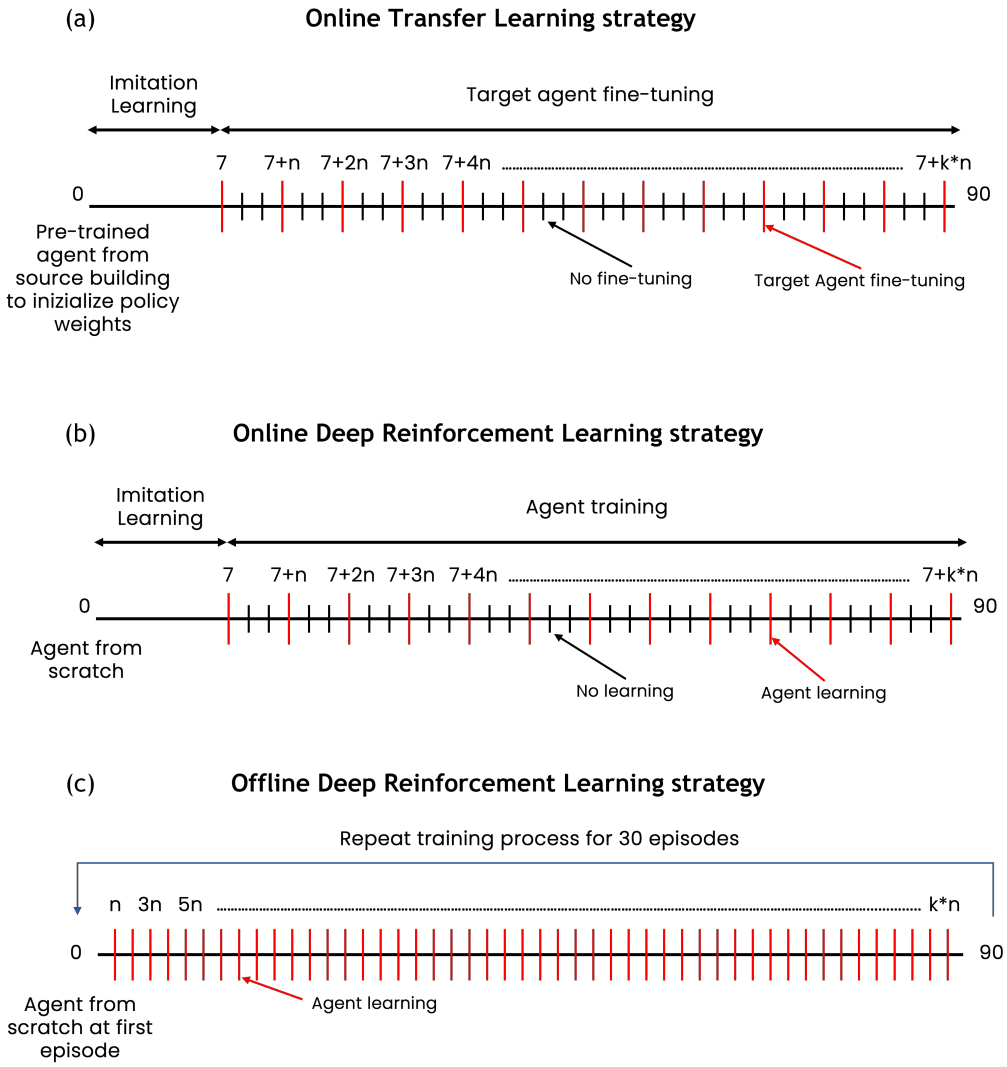

Figure 5: Learning strategies deployed in target buildings: (a) online transfer learning, (b) online deep reinforcement learning and (c) offline deep reinforcement learning

5. Implementation

This section discusses the implementation details for the source building and configurations of target buildings. Moreover, it provides a detailed description of the developed simulation environment and the control strategies

implemented on source and target buildings.

5.1. Implementation details on source and target building configurations

This section provides details about the source and target building configurations in which RBC, offline DRL, online DRL and OTL controllers are implemented. As specified in Section 3, source and target buildings are characterized by the same geometrical features and cooling system.

In particular, these buildings consist of three spaces, two 10-person office rooms and one 3-person control room, plus a technical room not served by the air-conditioning system, with a total floor area of 196 m^2 and a net conditioned area of 97 m^2. The office rooms and control room are occupied at maximum capacity during the whole occupancy period. The average transmittance values of the opaque and transparent envelope components for the source buildings are 0.16 and 0.55 W/m^2K respectively, with a window-to-wall ratio of 7 %.

The source building is occupied from Monday to Friday between 8:00 and 18:00. The price of electricity supplied from the grid to enable the operation of the energy system in the source building is defined according to a time-based tariff structure (i.e., Time-Of-Use (TOU)) commonly applied in Italy, derived from the 0.143 €/kWh electricity average price for the period June-September 2021 indicated by Italian grid regulating authority [66] (i.e., ARERA). In detail, three price bands were defined: low price, with a rate of 0.071 €/kWh (i.e., equal to one-half of the medium electricity price defined by ARERA); medium price, with an electricity rate of 0.143 €/kWh (i.e., assuming a value equal to the medium electricity price defined by ARERA); high price, with a rate of 0.214 €/kWh (i.e., equal to 1.5 times the medium electricity price defined by ARERA).

The chiller capacity and the power delivered to each thermal zone are determined from the ideal case where the building demand is considered as an external disturbance of the system. According to the ideal-load EnergyPlus calculation, the design cooling power to maintain an indoor temperature of 26 °C and a relative humidity of 55 % during the occupancy period is 1.8 kW per each office zone and 1 kW for the control room. Moreover, the chiller has a 12 kW design capacity Q_{cap}. These design values are derived for the source building from the sizing process when implementing the reference weather file available in EnergyPlus for Turin, Italy (ITA-TORINO-CASELLE-IGDG.epw). Furthermore, the TES is sized considering 3 times

the maximum ideal hourly cooling demand of the building. Therefore, according to the ideal-load calculation, the TES size for the source building is $3\,m^3$. The same approach was adopted to find these design features for each target building, according to each weather condition.

Other specifications for the TES and chiller are the same for source and target buildings. In detail, the thermal energy storage operates between a minimum temperature $T_{s,min}$ of 10 °C and a maximum temperature $T_{s,max}$ of 18 °C, which match respectively the maximum ($SOC_{TES} = 1$) and minimum ($SOC_{TES} = 0$) state of charge. The design water mass flow rate during the charging phase is 0.2 kg/s while for discharging phase corresponds to the sum of the design mass flow rates of the office rooms and control room, equal to 0.35 kg/s. The chiller supply water temperature at the outlet is 7 °C, while reference leaving and entering fluid temperatures are respectively 6.7 °C and 35 °C. These two features are employed by the EnergyPlus chiller model to provide the reference Coefficient of Performance (COP) value, equal to 2.7.

Source and target buildings differ in terms of weather conditions, electricity price schedules, occupancy schedules and building thermophysical properties, but have the same geometry and energy system. As a result, the effect of each variable on the performance of TL can be independently quantified. The target buildings were evaluated in four different cities (i.e., weather conditions) and employed different electricity price schedules and occupancy schedules to explore the capabilities of the transferred agent in adjusting the pre-trained control policy from source building considering these changes. The examined configurations of target buildings are shown in Figure 6.

Each target building configuration is denoted by the code T_{wxyz}, where w [0, 3] refers to the climatic conditions investigated, x [0, 1] and y [0, 1] to the price and occupancy schedules and z [0, 4] to the building envelope features considered. The impact of climate on the control policy transfer process was evaluated considering the same (i.e., Turin) or similar (i.e., Paris) climatic conditions as those of the source building, as well as very different conditions, in warmer (i.e., Palermo) or colder (i.e., Helsinki) locations. These localities were chosen according to the classification established by the European Commission based on Cooling and Heating Degree Days. Each city represents a particular climate type (e.g., mediterranean climate for Palermo) according to the weather classification described in [67, 68].

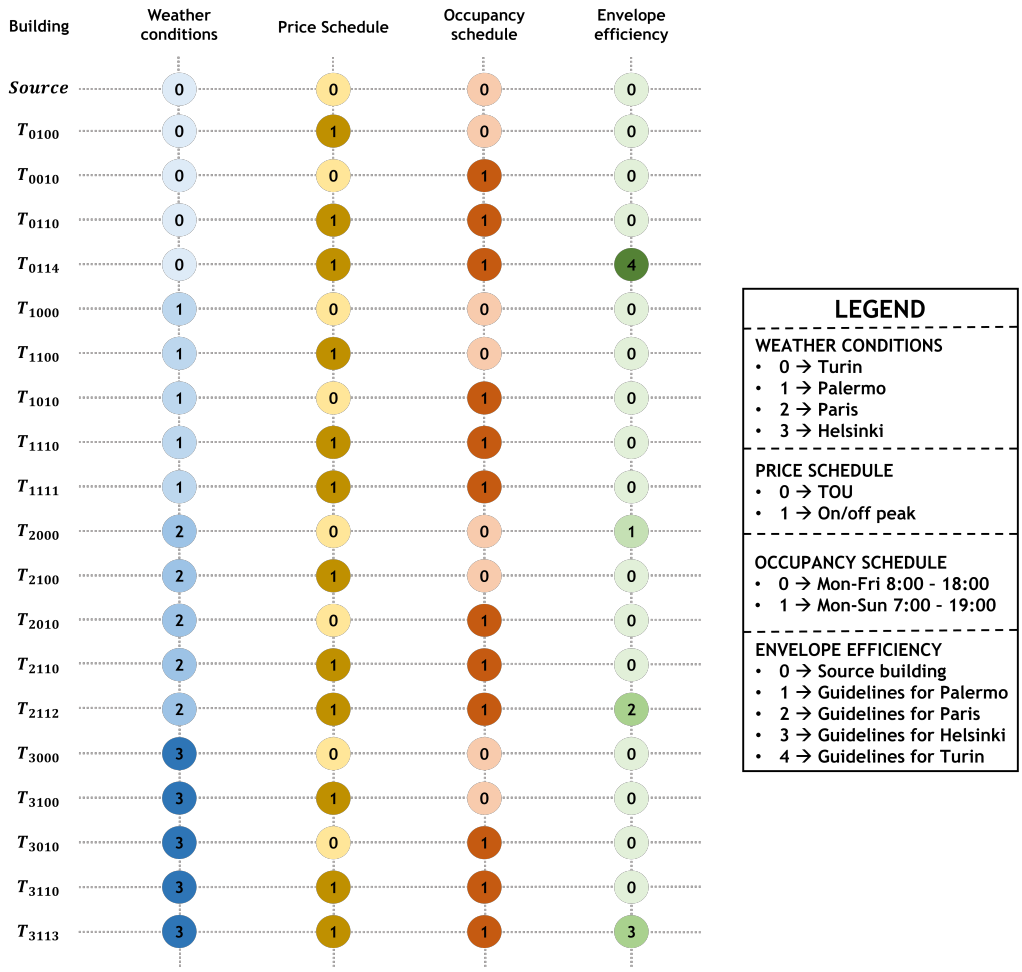

Figure 6: Source building and target building configurations

Furthermore, the target building configurations are distinguished by the electricity price and occupancy schedules employed. In detail, two price schedules were considered, as shown in Figure 7: the first schedule (i.e., 0) is based on TOU as in the source building, while the other (i.e., 1) is an on-off peak price scheme based on the Austin (Texas) electricity tariffs [69]. Specifically, an off-peak rate (0.029 €/kWh) during the period 20:00 - 7:00 and an on-peak rate (0.063 €/kWh) during the daytime period 7:00 - 20:00 were assumed. Two occupancy schedules were implemented in the target buildings and these differ in terms of weekdays and occupancy hours.

The first occupancy schedule (i.e., 0) assumes that the building is occupied during the period Monday-Friday 8:00-18:00, while the other schedule (i.e., 1) evaluates the presence of occupants during the period Monday-Sunday 7:00-19:00.

Eventually, different combinations of envelope efficiency were assessed for each target building, matching the thermophysical properties of the opaque envelope (i.e., opaque thermal transmittance U_{OP} and internal heat capacity χ_i) and the transparent envelope (i.e., transparent thermal transmittance U_{TR} and solar heat gain coefficient g). The five envelope efficiency combinations employed for target buildings are shown in Table 1. The envelope efficiency configuration 0 refers to the source reference building, while the others are defined according to the building standards of each locality in which the requirements for the thermophysical features are specified. The thermophysical properties values of the reference buildings for Palermo (i.e., efficiency configuration 1), Paris (i.e., efficiency configuration 2), Helsinki (i.e., efficiency configuration 3) and Turin (i.e., efficiency configuration 4) were chosen according to [70, 71, 72, 73].

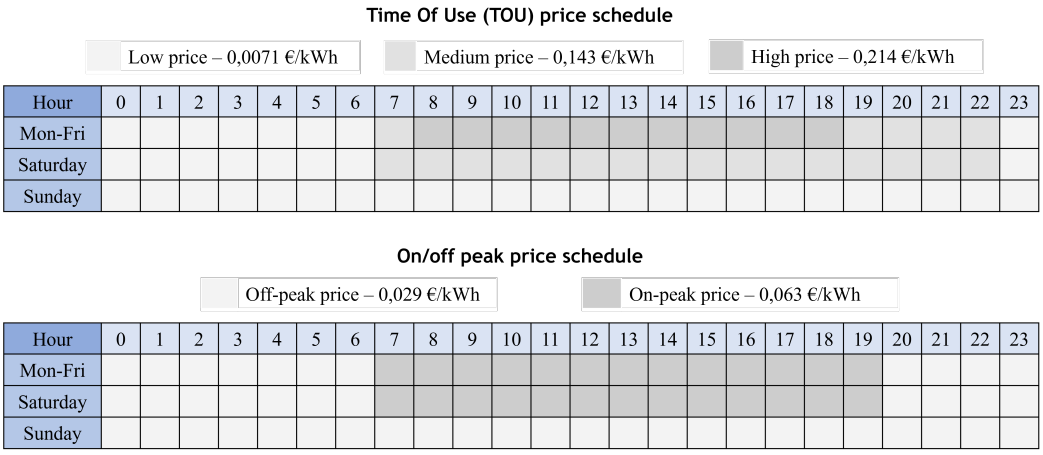

Figure 7: Electricity price schedules employed for target buildings

Table 1: Opaque and transparent envelope features for target buildings

Efficiency configuration	$U_{OP}[W/(m^2 * K)]$	χ_i $[kJ/(m^2 * K)]$	$U_{TR}[W/(m^2 * K)]$	Solar factor g
0	0.16	38.9	0.5	0.49
1	0.45	48.0	2.5	0.35
2	0.34	44.2	1.9	0.65
3	0.18	40.0	1	0.5
4	0.3	43.5	1.3	0.35

5.2. Simulation environment

The experiments were conducted via a co-simulation environment integrating EnergyPlus [28] and a Python interface based on OpenAI Gym [74]. Figure 8 shows the architecture of the developed co-simulation environment, modified from [16]. Building dynamics and energy system are modeled in EnergyPlus. At each simulation time step, the EnergyPlus building model receives in input the control actions from the Python side and information about weather conditions from the EnergyPlus reference weather file. The outputs of this model consist of information on the energy system (i.e., TES SOC), indoor conditions (i.e., indoor air temperature, occupancy status) and additional information (i.e., outdoor air temperature, weekday and hour of the day) included on the state-space of the controller. Control systems are developed in Python. In detail, the outputs from the EnergyPlus side and information about the electricity price are provided as inputs to the Python side, while the outputs are the control actions (i.e., cooling system operation mode and cooling fraction to thermal zones). The RBC and DRL agent select the control action at each time step according to the state space information and the reward function. The two software are interfaced through the Building Control Virtual Test Bed (BCVTB), operating as a middleware [23], and the *ExternalInterface* function of EnergyPlus. The interaction between Python and EnergyPlus is dynamic and occurs during each simulation time step. However, it can occur that the agent does not perform a control action in each simulation time step. In this case, time steps are distinguished between control and simulation. The simulation time step was set to 15 minutes since it ensures an optimal convergence of numerical results during the EnergyPlus simulation. However, in this work the control time step was not set equal to the simulation time step since it is not optimal to perform an action every 15 minutes in an energy system including a TES. As a result,

the control time step was set to 30 minutes to adequately take into account for the thermal inertia of the TES. In this framework, each control action (performed every 30 minutes) was applied to every two simulation time steps. A similar approach was adopted in [13, 23].

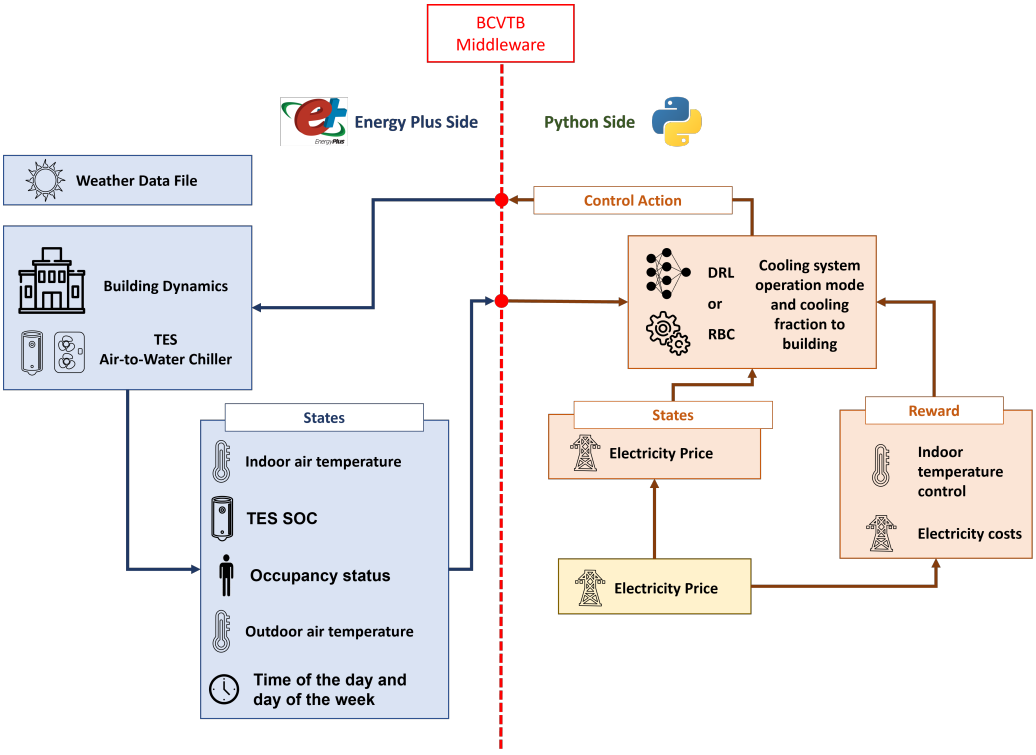

Figure 8: Architecture of the simulation environment (modified from [16])

5.3. Rule-based control strategy

The RBC is made of two agents that decide to provide cooling energy to the environment (RBC_{CF}) and the operation mode of the energy system (RBC_{OM}). The RBC_{CF} control logic consists of two parts, a pre and post first switch ON phase, where the agent starts to supply cooling energy to the building according to specific indoor temperature conditions and the time of the day during working days. The RBC_{CF} starts to supply cooling energy to the thermal zone according to the requirements shown in Table 2. The four combinations listed in Table 2 of start time window and indoor temperature conditions resulted from a sensitivity analysis where different

thresholds were tested to minimize temperature violations during the early stages of the occupancy period.

Table 2: Time period and indoor temperature conditions in RBC_{CF} for the starting phase

Combination	Time period	Indoor temperature
1	$4:00 \leq t < 5:00$	$T_{INT} - T_{UPP} \geq 3°C$
2	$5:00 \leq t < 6:00$	$T_{INT} - T_{UPP} \geq 2°C$
3	$6:00 \leq t < 7:00$	$T_{INT} - T_{UPP} \geq 1°C$
4	$t \geq 7:00$	$T_{INT} - T_{UPP} \geq 0°C$

After the starting phase, cooling energy is supplied to the building until the indoor temperature falls below the lower threshold of the acceptability range T_{LOW} (i.e., 25 °C). Conversely, when the indoor temperature rises over the upper threshold of the acceptability range T_{UPP} (i.e., 27 °C), the RBC_{CF} agent starts again to provide energy to the building. The cooling energy supply is interrupted when occupants leave the building (i.e., 18:00). The second agent (RBC_{OM}) manages the cooling energy system to select its operation mode. In detail, when the electricity price is low and the SOC_{TES} is lower than 0.75 (i.e., corresponding to a TES temperature of 12 °C), the RBC_{OM} operates the cooling system in charging mode, until the electricity price rises above the minimum value or the SOC_{TES} reaches the maximum value (i.e., SOC_{TES} equal to 1 or TES temperature equal to 10 °C). When the RBC_{CF} decides to supply cooling energy to the building and the electricity price is not low, RBC_{OM} operates the cooling system in discharging mode if SOC_{TES} is not zero and in chiller mode if the storage is empty.

5.4. Design of DRL control strategy

In this section, details about the design of the reward function and state-action spaces are provided. Furthermore, the setup of the agent during the training phase on source building is discussed.

5.4.1. Design of state-space

The choice of observations to be included within the state-space is fundamental for the DRL controller as it enables the agent to understand the effect of the selected action on the controlled environment. Furthermore, the state-space should be constituted by variables easily measurable and that speed

up the control problem optimization, increasing the chances of its real-world implementation. The state variables used in this work are shown in Table 3, with a detail relative to the reference time step (i.e., specified as a function of the actual control timestep t). Incorporating the *Outdoor air temperature* in the state space was required to evaluate its impact on building energy consumption. *Indoor temperature* conditions were evaluated employing the difference between the indoor setpoint temperature and the actual indoor air temperature. This variable was observed at the current time step and at the two previous timesteps (i.e., t-1 and t-2), to assess the temperature evolution in the building over time and account for the thermal dynamics effect of the building and energy system [16]. For the same reason, the state of charge of the cooling thermal storage (*TES SOC*) was evaluated during the same timesteps. Introducing the SOC within the state space was required to provide the agent with adequate information for better managing the cooling system. Since it is required that the DRL agent reduces the cost of electricity withdrawn from the grid, the *Electricity price* was included within the state space. In addition, its perfect predictions over the next 12h were accounted for enabling the agent to optimally choose the operation mode of the cooling system. The information regarding the presence of occupants in the building was provided to the agent through the (0, 1) binary variable *Occupants' presence status*, included in the state-space for the current time step and the following 12 h. To conclude, the occupancy schedule could be recognized by the controller by combining this feature with the last two variables contained in the state-space, *Day of the week* and *Time of the day*.

Table 3: Variables included in the state-space

Variable	Unit	Timestep
ΔT Indoor Setpoint - Mean Indoor Air	°C	t, t-1, t-2
SOC_{TES}	-	t, t-1, t-2
Outdoor air temperature	°C	t
Time of the day	h	t
Day of the week	-	t
Electricity price	€/kWh	t, t+1, ...,t+24
Occupants' presence status	-	t, t+1, ...,t+24

5.4.2. Design of action-space

Considering that the discrete version of SAC was used as control algorithm, the action space is of the same type. The action space could be expressed as:

$$A = OMxCF = (om, cf) : om \in OM, cf \in CF \tag{1}$$

In detail, the DRL controller chose an action in the range [0, 4], each one corresponding to a combination of Operation Mode (OM) and Cooling Fraction (CF). Therefore five possible actions could be selected by the agent according to Table 4 to choose:

- the operation mode of the cooling system (i.e, discharging mode [-1], cooling mode [0], charging mode [1]);
- whether [-1] or not [0] to supply cooling energy to the thermal zone.

Moreover, safety constraints were introduced to avoid that the system operated in charging mode when the storage was fully charged (i.e., SOC_{TES} = 1) and in discharging mode when the storage was empty (i.e., SOC_{TES} = 0). In these circumstances, the system operated in chiller mode.

Table 4: Details on action-space

Action	Operation mode	Cooling fraction
0	0	0
1	1	0
2	-1	-1
3	0	-1
4	1	-1

5.4.3. Design of reward function

The reward function must be defined according to the objectives of the control problem. Therefore, in this case study the reward function was made of two terms, an electricity cost-related term and a temperature-related term, since the DRL controller was designed to reduce electricity cost of the cooling system while improving indoor temperature conditions. Furthermore, the two reward terms were weighted by introducing coefficients δ and β to adjust their importance. The general expression of the reward was defined as follows:

$$R = -(\delta * R_E + \beta * R_T) \tag{2}$$

The electricity cost-related term refers to costs associated with the energy withdrawn from the grid to feed chiller and circulation systems, and it was expressed in the following way:

$$R_E = c_E * (E_{CHILLER} + E_{PUMP}) \tag{3}$$

where c_E [€/kWh] is the electricity price for buying defined according to the implemented price schedule, while $E_{CHILLER}$ and E_{PUMP} corresponds to chiller and pumping system energy consumption, evaluated in kWh.

The temperature related-term was defined according to the presence of the occupant and the indoor temperature conditions.

When the building is not occupied, the temperature-term is:

$$R_T = 0 \tag{4}$$

During working hours, the temperature-term could have different expressions:

- if $T_{INT} < T_{LOW} - 2$:
$$R_T = 50 \tag{5}$$
- if $T_{LOW} - 2 \leq T_{INT} < T_{LOW}$:
$$R_T = (SP_{int} - T_{INT})^3 \tag{6}$$
- if $T_{LOW} \leq T_{INT} \leq T_{UPP}$:
$$R_T = 0 \tag{7}$$
- if $T_{UPP} < T_{INT} \leq T_{UPP} + 2$:
$$R_T = (T_{INT} - SP_{int})^3 \tag{8}$$
- if $T_{INT} > T_{UPP} + 2$:
$$R_T = 50 \tag{9}$$

The reward had a fixed value if the temperature was below 23 °C or above 29 °C to avoid convergence problems for the algorithm related to the high magnitude of the reward, as in the SAC the learning process was influenced by the definition of the Boltzmann temperature coefficient α as a function of the reward magnitude.

5.5. Training setup of DRL agent on source building

The DRL agent learns the optimal control policy on the source building before sharing it with the target buildings in the case of OTL. During the training phase the controller is trained in offline manner, whose implementation details are reported in the Subsection 4.4.1. The performance of DRL algorithms depends significantly on the choice of several hyperparameters which have to be chosen accurately. To this end, an automated procedure was adopted to extract the optimal set of hyperparameters using the open-source Python library Optuna [61]. The hyperparameters optimization was performed only during the training phase of the DRL agent on source building. In particular, Optuna minimizes or maximizes an objective function by performing the optimization of the set of hyperparameters provided as input with the corresponding acceptability range. The hyperparameters and the corresponding range values involved in the optimization process are listed in Table 5. In this work, the Tree-structured Parzen Estimator (TPE) was chosen among Optuna sampling algorithms [75] to optimize a multi-objective function, since the DRL agent should minimize the electricity cost while reducing the indoor temperature violations compared to the RBC strategy in the source building. In this case an optimal Pareto-front solution set exists [76], so it was employed the criterion of the minimum Euclidean distance from the ideal point [77] (i.e., the non-real point whose coordinates have the lowest values when separately considering the objectives in the target function). During the automated hyperparameter optimization procedure, twenty agents trained for 30 episodes were considered, and a coordinate point [E_{cost}, T_{viol}] indicating its performance was retrieved for each agent. E_{cost} represents the total electricity cost, calculated as the product of the electricity cost withdrawn from the grid and the sum of the energy consumption of the chiller and auxiliaries in kWh, and it is expressed as follows:

$$E_{cost} = c_E * (E_{CHILLER} + E_{PUMP}) \tag{10}$$

T_{viol} stands as the cumulated sum of the temperature violations during the whole cooling season, as indicated in the following equation:

$$T_{viol} = \sum_{t=0}^{t_{end}} T_{viol,i} \tag{11}$$

A temperature violation $T_{viol,i}$ is computed as the absolute temperature difference between indoor temperature and the upper or lower limit of the

temperature acceptability range [25, 27] °C, when the indoor temperature falls outside this range during the occupancy period. $T_{viol,i}$ could have different expressions according to the indoor temperature value T_{INT}:

- if $T_{INT} < T_{LOW}$:
$$T_{viol,i} = T_{LOW} - T_{INT} \tag{12}$$
- if $T_{LOW} \leq T_{INT} \leq T_{UPP}$:
$$T_{viol,i} = 0 \tag{13}$$
- if $T_{INT} > T_{UPP}$:
$$T_{viol,i} = T_{INT} - T_{UPP} \tag{14}$$

Therefore, the Euclidean distance between the performance at the end of the training phase of each source DRL controller and the ideal point was calculated. Therefore, the solution with the lowest distance and the best performance compared to the RBC in terms of total electricity cost and cumulated sum of temperature violations was chosen as the best.

Table 5: Ranges of DRL hyperparameter values involved in the optimization

Hyperparameter	Value
# Hidden layers	[2, 4]
# Neurons per layer	[64, 128]
Batch size	[64, 128]
Discount factor γ	[0.9, 0.95, 0.99]
Actor/Critic learning rate μ	[0.00025, 0.0005, 0.00075, 0.001]
Reward electricity cost-term weight factor δ	[2, 4, 6, 8, 10, 12]
Reward temperature-term weight factor β	[0.015, 0.03, 0.045, 0.06, 0.075, 0.09]

A training episode includes 90 days, from 1 June to 29 August. Each episode took on average 35 minutes to be simulated on a machine with an 8th Generation Intel@CoreTMi7-8550U @ 4.0 GHz processor and 16.0 GB RAM. The simulation of an episode (which corresponds to one cooling season) includes both EnergyPlus simulation and Python control process. During the simulation, the exchange of information between EnergyPlus and Python was handled through BCVTB. Thus, 35 minutes per episode refer to the time required to complete the simulation of one episode for the DRL controller on

the source building. However, that time is not relevant in the framework of OTL. In fact, the OTL strategy proposed in the present work was conceived to emulate the real-world implementation of a DRL agent pre-trained on the source building to different target buildings without any further modeling effort.

5.6. Implementation details on online transfer learning and DRL learning strategies

This section provides insights about the implementation of the OTL and DRL offline and online learning strategies.

The automated optimization of hyperparameters, also including reward weights δ and β and Boltzmann temperature coefficient α, was carried out only for the DRL controller trained on the source building. The hyperparameters were not re-optimized in target buildings since their optimization process should be performed over several episodes. However, in building applications one episode usually represents an entire cooling/heating season, hence a re-optimization appears inconsistent with the online strategy implemented in this work. Therefore, the hyperparameters δ, β and α are the same as those optimized in the source DRL controller for all controllers implemented in target buildings. However, the weight factor δ of the reward electricity cost-term was doubled, after a sensitivity analysis, compared to that of the source DRL agent when the price schedule implemented in the target building was of the on-off peak type. This procedure was necessary to balance the two reward function terms since the TOU price tariff implemented in the source DRL agent is represented by a higher average weekly electricity price than in the on-off peak tariff case.

The three advanced control strategies implemented in the target buildings differ in terms of the value of some hyperparameters (i.e., batch size, learning rate, learning step and gradient steps), as indicated in Table 6.

Compared to the controller trained on the source building, the period of analysis is the same (i.e., June-August) and the complete set of hyperparameters remains unaltered only for the offline DRL. In the offline DRL setting, a training episode was repeated 30 times before obtaining an optimal solution, with a control time step of 30 minutes and a batch size of 128. Conversely, the online DRL and OTL strategies were implemented for a single episode aiming to represent the direct application in the real system. Moreover, the batch size value was reduced to 32 since a smaller data volume is available to train the control policy. This results in a faster convergence speed towards

a near-optimal solution [78], a prerequisite for a DRL agent directly implemented on the system without offline pre-training. Furthermore, the online DRL and OTL employ larger values of training steps (i.e., every 3 days) and gradient steps (i.e., 20) when compared to the offline DRL strategy. In the case of the OTL, this prevents that the pre-trained control strategy on the source building is not entirely overwritten (i.e., also using a reduced learning rate value), while guaranteeing that the control policy can be optimized according to the different boundary conditions in the target building to be controlled. This approach avoids the over-exploration of the action space since this might result in a deviation from the optimal control policy that the agent may learn at the beginning of the training phase. In the case of online DRL, the use of a gradient step equal to 20 is effective in accelerating the training process during the first weeks of implementation, since the agent in the online DRL configuration has limited amount of available data for training due to the limited experience stored in the buffer at the beginning of the process. Furthermore, using a training step of three days results in performance degradation in the early stages of training but ensures that the control agent acquires a larger number of transitions before performing the next learning stage. Thus, the performance level of the agent improves moving forward in the training period.

Table 6: Hyperparameters selected for offline DRL, online DRL and OTL

Hyperparameter	Offline DRL	Online DRL	OTL
Batch size	128	32	32
Actor/Critic learning rate μ	0.001	0.001	0.0005
Training Episodes	30	1	1
Learning step	30 min	Every 3 days	Every 3 days
Gradient steps	1	20	10

6. Results

This section outlines the results achieved by implementing the methodological framework described in Section 4. The result of the training on the source building of the proposed DRL controller are presented in the first part of the section. The second part describes the results of the proposed OTL strategy and the relative comparison with RBC and DRL approaches.

6.1. Training of DRL agent on source building

As introduced in Section 5.5, the values of the hyperparameters characterizing the DRL controller were optimized through a procedure implemented in the python library Optuna [61]. Twenty different configurations of hyperparameters were analyzed and listed in Table 7. All configurations were trained on 30 episodes with a Boltzmann temperature coefficient α of 0.1. The best configuration was selected according to the criterion of the minimum distance from the ideal point [77] since the optimization process involves two different objectives (i.e., minimization of total electrical cost and minimization of cumulated sum of temperature violations). According to this criterion the 9th configuration, highlighted in yellow in Table 7, was the best among the twenty analyzed configurations.

Compared to the RBC, the DRL controller achieved an electricity cost saving of 19.6 % ($E_{cost,DRL}$ = 56.6 €vs $E_{cost,RBC}$ = 70.4 €), as well as a significant enhancement in indoor temperature conditions, due to a 69 % reduction in the value of cumulated sum of temperature violations over an entire cooling season ($T_{viol,DRL}$ = 54.7 °C vs $T_{viol,RBC}$ = 176.2 °C). Figures 9 and 10 provide details about the comparison between the DRL and RBC controllers implemented on the source building. Figure 9 shows the indoor temperature profiles obtained with the RBC and DRL agent during 15 days of the analyzed period, while Figure 10 provides insights about the chiller consumption (on the top panel) and SOC evolution (on the bottom panel) for both controllers and during the same period evaluated in Figure 9, with a detail on the electricity price tariff.

Table 7: Configurations of DRL hyperparameters involved in the optimization

Configuration	# Layers	# Neurons	Batch size	γ	μ	δ	β	E_{cost} [€]	T_{viol} [°C]
1	4	64	128	0.95	0.0005	10	0.075	57.5	402.6
2	4	128	128	0.99	0.00075	12	0.06	69.9	256.1
3	2	128	64	0.99	0.0005	12	0.09	59.1	218.9
4	2	64	64	0.9	0.00025	6	0.6	65.3	190.5
5	4	64	128	0.99	0.001	2	0.075	128.7	59.7
6	2	128	128	0.95	0.001	8	0.03	76.4	199.4
7	4	128	128	0.9	0.00025	10	0.045	96.3	348.1
8	2	128	128	0.99	0.00075	4	0.045	61.4	110.2
9	2	64	128	0.99	0.001	8	0.045	56.6	54.7
10	4	64	64	0.95	0.0005	2	0.015	59.3	134.8
11	4	128	128	0.9	0.00075	6	0.075	79.1	143.9
12	2	128	64	0.9	0.00025	10	0.03	64.1	452.1
13	4	128	128	0.99	0.001	12	0.015	72.3	392.4
14	4	64	64	0.95	0.0005	6	0.045	60.4	121.8
15	4	64	128	0.99	0.001	12	0.03	50.1	106.2
16	2	64	128	0.9	0.00075	4	0.075	69.2	90.7
17	2	128	128	0.99	0.00025	10	0.09	67.6	110.1
18	4	128	64	0.99	0.0005	8	0.06	72.3	129.3
19	4	64	128	0.95	0.001	2	0.015	87.2	116.9
20	2	64	128	0.9	0.00025	2	0.045	68.4	63.2

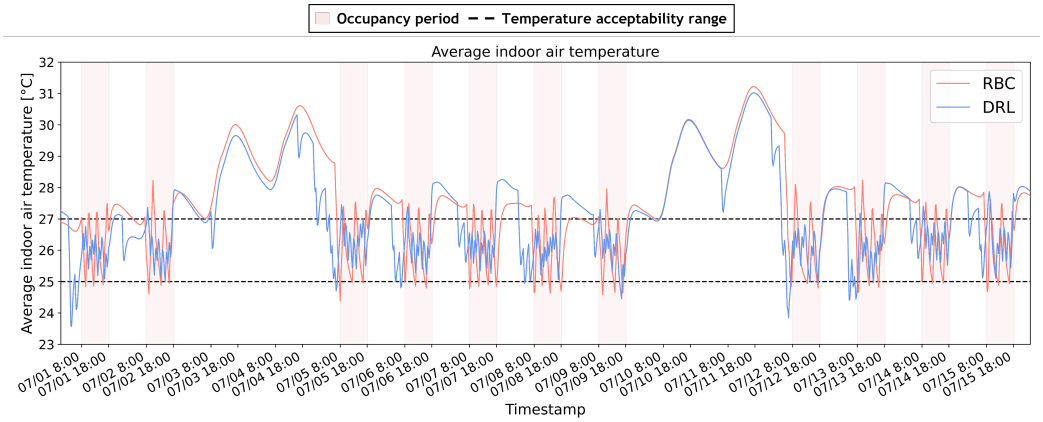

Figure 9: Indoor temperature profile with RBC and DRL for the source building

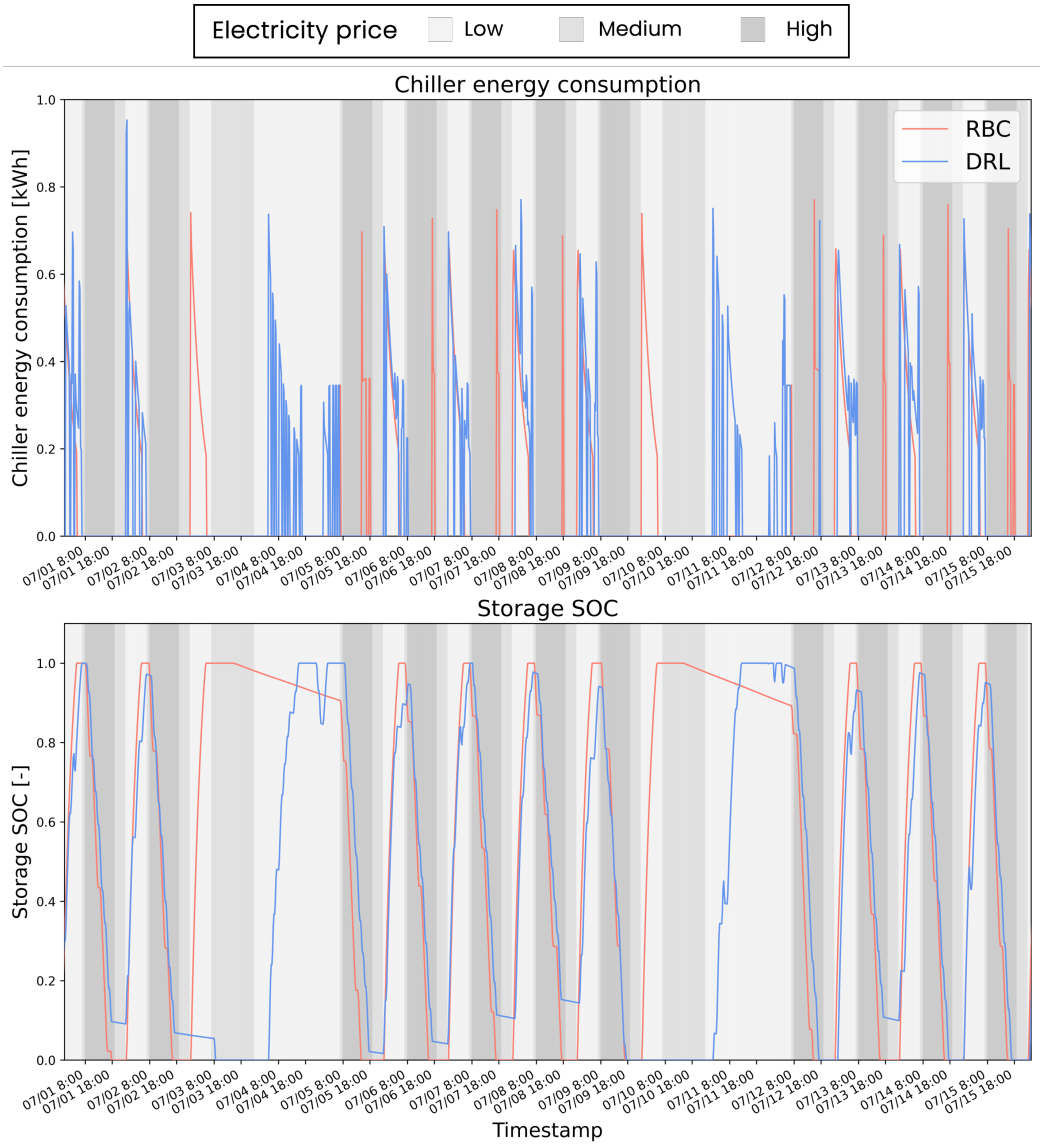

Figure 10: Chiller energy consumption and storage SOC evolution with RBC and DRL for the source building

The DRL controller achieved better performance in terms of indoor temperature control and reduction of electricity cost with respect to the RBC through a more accurate management of the energy system. As shown in Figure 9, the DRL agent scheduled in advance the supply of cooling energy

to the thermal zones compared to RBC. Moreover, the DRL controller maintained a better control over the indoor temperature values during the day limiting the number of times in which indoor temperature raised above 27 °C. Although the building was not occupied, the implementation of DRL controller results in some temperature drops during weekends. This pattern is linked to the formulation of the reward function, since the temperature drops are associated with the supply of cooling energy to thermal zones by means of the activation of the chiller during low-cost hours, as indicated in Figure 10. As a result, the indoor environment is pre-cooled to ensure that the temperature is maintained as close as possible to the temperature acceptability range during the early stages of occupancy period on Monday. Moreover, this behaviour allows to delay the operation of TES compared to RBC.

From Figure 10 can be observed how the DRL controller managed the cooling system by operating the chiller in charging mode during low-price periods (in light grey) to supply energy to the environment and to charge the TES. In detail, the TES was fully charged at the end of the low price period to operate the system in discharging mode during the medium or high electricity price periods to save energy related to the operation of chiller by maximizing the utilization of the TES. Contrarily, the RBC usually operated the system in chiller mode during high-price periods (dark grey) since TES was discharged before the end of the occupancy period when the building required cooling energy to meet indoor temperature requirements. Furthermore, during weekends (i.e., 07/03-07/04 and 07/10-07/11) RBC and DRL controllers managed differently the cooling system. In particular, RBC charged the TES during the early stages of weekend, while DRL controller charged the TES by the end of the weekend. As a result, DRL minimized TES losses as well as holding the maximum SOC at the beginning of the occupancy period in contrast to RBC.

6.2. Performance benchmarking of online transfer learning with RBC and DRL control strategies on target buildings

This section presents the results derived from the implementation for a cooling season lasting 90 days (i.e., from 1 June to 29 August) of the OTL strategy on the target buildings analyzed, providing a benchmark with the performance achieved with the RBC and the offline and online DRL controllers developed. OTL and online DRL were implemented over a single episode as specified in Section 5.6. Conversely, offline DRL involved a training phase performed for 30 episodes followed by a testing phase performed

for 1 episode in which the agent was statically deployed to evaluate the performance of the learned control policy.

Table 8 summarizes the performance achieved from the implementation of RBC, Offline DRL (Off-DRL), Online DRL (On-DRL) and OTL (highlighted in yellow) in all target buildings in terms of electricity cost and temperature violations. Each target building is denoted by the code T_{wxyz}, where indices refer respectively to the considered climatic condition (w), price schedule (x), occupancy schedule (y) and envelope efficiency (z). Overall, it can be noticed that the OTL agent performed better in terms of both total electricity cost and cumulated sum of temperature violations with respect to RBC and online DRL. However, the OTL agent was outperformed by offline DRL solution. This pattern was expected since offline DRL controllers had at their disposal several episodes (i.e., 30 episodes) for each target building to converge to the optimal control policy. Conversely online DRL and OTL strategies relied on a single simulation episode to emulate the direct implementation of these controllers in physical systems.

Table 8: Total electricity cost and cumulated sum of temperature violations for all investigated control strategies for each target building

Target buildings	Total electricity cost C_E [€]				Cumulated sum of T violations T_{viol} [°C]			
	RBC	Off-DRL	On-DRL	OTL	RBC	Off-DRL	On-DRL	OTL
T_{0100}	26.0	22.6	32.2	25.4	175.4	60.4	504.9	71.5
T_{0010}	95.5	82.3	116.9	84.7	288.3	106.8	607.6	125.0
T_{0110}	40.9	34.4	42.5	34.6	288.3	123.0	652.6	127.1
T_{0114}	34.5	31.0	40.3	31.4	297.0	111.6	790.2	128.3
T_{1000}	115.9	88.1	135.9	92.8	234.5	33.9	579.4	108.6
T_{1100}	40.9	37.4	47.8	38.1	237.7	36.6	550.8	89.4
T_{1010}	141.6	108.8	173.7	122.5	352.2	99.3	855.9	180.0
T_{1110}	54.2	47.4	61.4	48.2	352.7	34.4	657.1	148.6
T_{1111}	57.4	48.7	62.8	50.3	362.9	57.0	634.9	144.0
T_{2000}	64.1	56.2	34.8	56.9	158.5	56.8	703.6	71.7
T_{2100}	23.9	23.5	32.9	23.7	165.3	82.4	673.7	106.1
T_{2010}	93.5	84.8	110.4	87.1	278.3	90.1	714.6	116.6
T_{2110}	36.6	33.5	40.6	34.0	275.7	102.1	607.0	108.3
T_{2112}	29.8	28.0	48.4	28.4	319.2	169.7	774.3	178.4
T_{3000}	61.3	54.2	89.7	57.5	153.6	54.9	550.2	108.7
T_{3100}	22.6	21.3	31.8	22.4	153.4	23.9	673.7	106.1
T_{3010}	90.2	77.9	121.2	83.3	276.0	117.5	871.1	179.1
T_{3110}	35.0	32.5	54.7	33.5	275.5	65.1	639.9	161.6
T_{3113}	36.1	33.3	43.0	34.8	276.8	40.9	810.4	145.4

In detail, OTL implementation led to an electrical cost higher between 1 % and 13 % as well as worse performance in terms of indoor temperature control compared to offline DRL (e.g., cumulated sum of temperature violations are twice or three times higher than those obtained from the offline DRL implementation for target buildings located in Palermo and Helsinki). However, it should be mentioned that the offline DRL agent training process involves the definition of a model that emulates the behaviour of the building, contrarily to the case of the OTL. Conversely, the OTL control strategy performed better than the online DRL controller since it had at its disposal information about the control policy pre-trained on the source building. In particular, the OTL was capable to achieve better performance than the online DRL since the boundary conditions between source and target buildings

were similar. Thus, part of the knowledge of the control policy trained on the source building was effectively exploited to reduce learning time in the target buildings by the OTL agent.

As a result, OTL agent achieved cost savings ranging between 16 % (target building T_{2110}) and 41 % (building T_{2112}) as well as ensured an average reduction of the cumulated sum of temperature violations over all experiments up to 82 % compared to online DRL agent. Furthermore, OTL controller achieved savings up to 20 % (building T_{1000}) in terms of electricity cost and a reduction between 30 % and 60 % in temperature violations compared to RBC.

A more detailed overview of the results achieved by RBC, offline DRL and OTL is provided in Figures 11 and 12. In these figures the target buildings were grouped according to the implemented price tariff. These figures show the target buildings arranged in ascending order with respect to the degree of change in weather conditions from the source building. Therefore, from left to right the buildings located in locations with equal (Turin), similar (Paris), colder (Helsinki), and warmer (Palermo) climates are clustered.

Thereby, these figures show how the performance of the OTL agent varied as a function of the differences between the source and target buildings. The performance of the online DRL agent has not been reported in these figures since it was worse compared to the other implemented controllers.

The scatter plot in Figure 11 displays the results in terms of total electricity cost and cumulated sum of temperature violations for the 7 target building configurations in which the TOU price schedule was implemented. Conversely, Figure 12 provides the same details for the 12 target buildings in which on-off price schedule was applied.

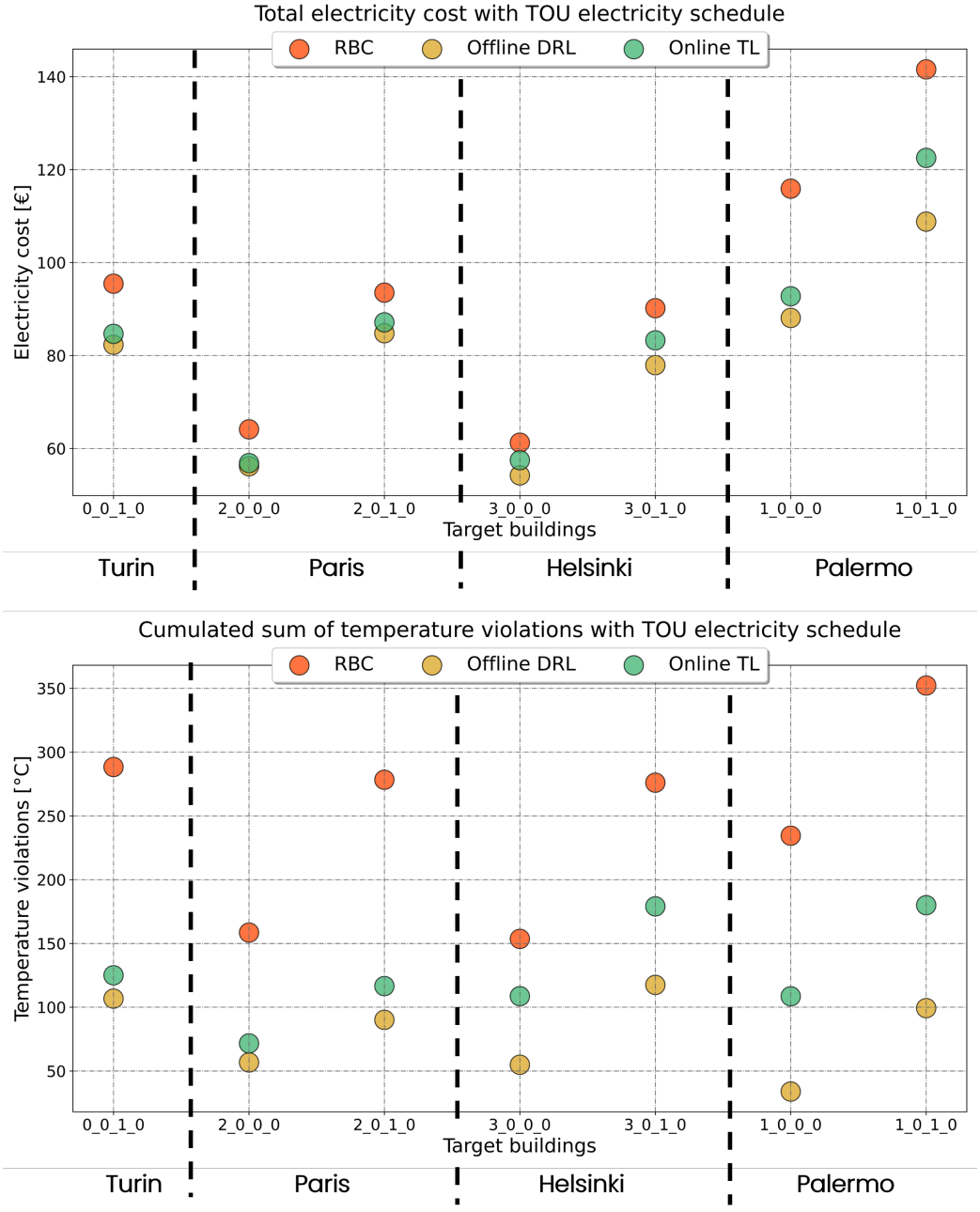

Figure 11: Total electricity cost and cumulated sum of temperature violations for target buildings with TOU pricing schedule

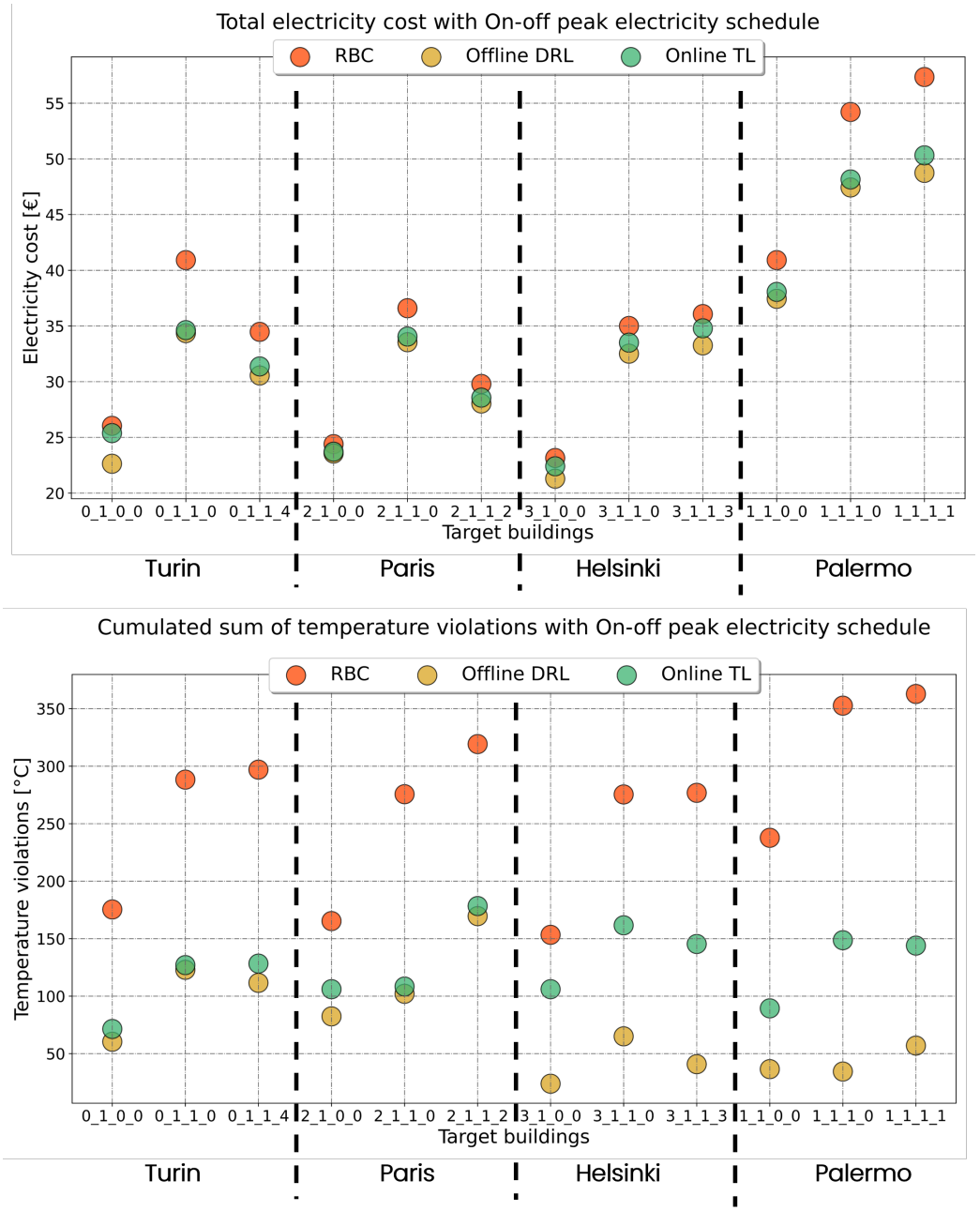

Figure 12: Total electricity cost and cumulated sum of temperature violations for target buildings with on-off peak pricing schedule

Figures 11 and 12 show that the performance of the control policy transfer on target buildings matched or was slightly worse than those obtained using offline DRL for the buildings located in Turin or Paris. Differences in total electricity cost and cumulated sum of temperature violations increased between the OTL and offline DRL controllers as weather differences (colder and warmer climates than the source building climate) increase. Moreover, considering the same climate but different occupancy schedules led to an increased gap between OTL and offline DRL as well, but to a smaller extent than modifying the weather conditions.

Following these considerations, weather differences between source and targets were identified as the most important influencing factor of the performance of the proposed transfer learning methodology. Since the analyzed case study focuses on thermal energy management, the importance of weather differences on the effectiveness of the proposed OTL methodology can be explained by the influence of the climatic conditions on the patterns of the thermal loads of the considered building.

However, if the price schedule implemented was of the on-off peak type and the climate was similar to the source building (i.e., Turin or Paris), the trend was reversed, as the change of the building occupancy pattern from schedule 0 to 1 causes a reduction in the performance gap between OTL and offline DRL in terms of cost and temperature violations (e.g., T_{0100} vs T_{0110}).

In the case of target buildings implementing the occupancy schedule 1 (i.e., building for which the y-value in the code T_{wxyz} is equal to 1) the OTL strategy achieved a greater improvement of the performance with respect to the RBC strategy than in the case in which the occupancy schedule 0 was selected considering the same weather conditions.

Figures 13 and 14 compare the cumulative curves of the daily total electricity cost and sum of temperature violations over the considered cooling season for the online and offline DRL (at the 30th episode) controllers and OTL controller. In particular, this outcome is shown for 4 particular target buildings, each related to investigated weather conditions (Turin and Paris in Figure 13 and Palermo and Helsinki in Figure 14) along with all possible differences compared to source building: electricity price schedule, occupancy pattern and envelope efficiency.

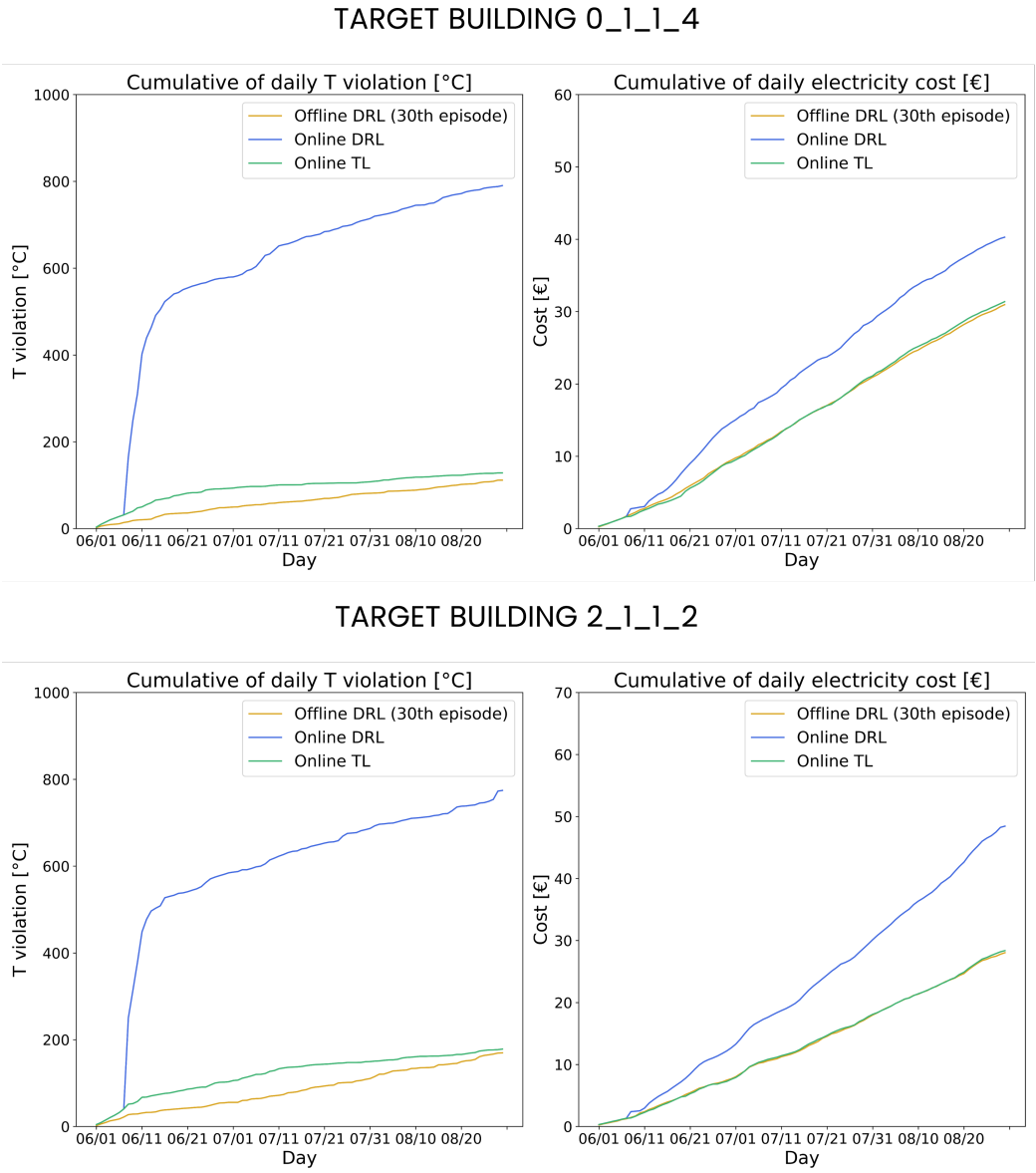

Figure 13: Daily cumulative curve of electricity cost and temperature violation for target buildings in Turin and Paris

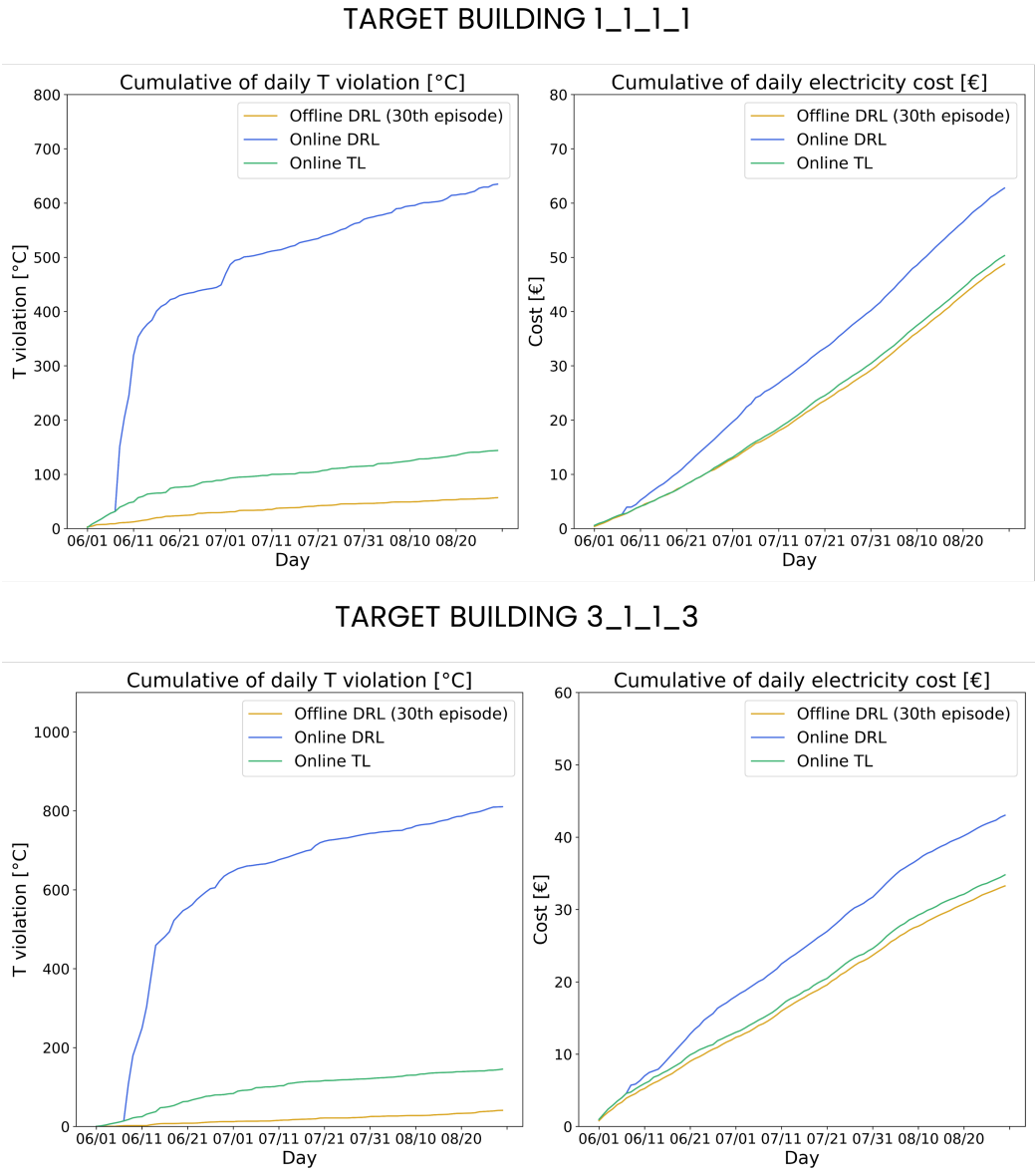

Figure 14: Daily cumulative curve of electricity cost and temperature violation for target buildings in Palermo and Helsinki

For all target buildings examined, the costs linked to the implementation of the online DRL controller were higher than those obtained via the offline DRL agent and the controller trained with OTL strategy as expected, since

it needed to explore behaviours in the early period that could be harmful, lacking a priori knowledge of the system. Moreover, the cumulative cost curves for OTL and offline DRL were similar over the cooling season. The cumulative curve of daily temperature violations exhibited a similar trend in all buildings analyzed in Figures 13 and 14 for the online DRL agent. In detail, after the first week of RBC implementation, the slope of the cumulative curve for online DRL agent (in blue) tended to infinity for approximately 10-15 days, due to the need for the controller to gain experience from interacting with the environment. Thereafter, the slope of the cumulative curve decreased since the control policy starts to converge to a near-optimal solution. For the OTL and offline DRL agents, the difference between the daily cumulative violation value for these controllers reached a maximum in the first 20 days of June, and then kept on the same deviation over the entire period for buildings located in Helsinki and Palermo as represented in Figure 14 or decreased for those located in Turin or Paris as indicated in Figure 13.

In summary, the implementation of the proposed OTL strategy achieved acceptable performance during early stages of deployment in all target buildings. The OTL performed better in terms of both electricity cost and temperature violations than the online DRL control strategy. This comparison between OTL and online DRL is fair since for both strategies the offline pre-training phase was not performed for the target buildings and highlights the benefits of exploiting transfer learning techniques for DRL controllers. The offline DRL performed better in terms of both electricity cost and temperature violations than OTL. However, this comparison is not fair since the offline DRL controller had the possibility to refine its control policy over multiple episodes for the target buildings. Eventually, the results obtained showed that differences in climatic conditions have the greatest impact on the performance of the OTL, as the performance gap with the offline DRL increases with the differences in climatic conditions between the source and the target buildings.

7. Discussion

This paper focused on the development of an effective transfer learning methodology to share a DRL-based control policy for the management of a TES-based cooling system in office buildings.

A DRL agent was pre-trained offline on a source building to minimize electricity cost associated to chiller operation while maintaining indoor air

temperature conditions. To this purpose, the controller can decide the operating mode of the cooling system and whether or not to supply cooling energy to the thermal zone.

The best pre-trained control policy was transferred to several target buildings characterized by the same geometry and the same energy system of the source but with different weather conditions, electricity price schedules, occupancy patterns and building envelope levels of efficiency.

The proposed transfer learning methodology was designed to enhance the scalability and deployment of DRL-based control strategies for the built environment that minimized utility costs while providing occupant comfort.

The innovative aspect of the proposed methodology with respect to the current scientific literature relies in the approach adopted to transfer a pre-trained control policy. Conventional applications of transfer learning reported in literature evaluate the performance of the controller by applying a fine-tuning process performed over several episodes. However, it is the authors' opinion that this approach is unable to fully demonstrate the applicability of transfer learning in improving the scalability of a DRL agent in a real-world context. If multiple episodes (i.e., a cooling season in the present study) are required for the transferred control policy to reach acceptable performance, then, in a real-world context, it would possibly take years of deployment.

To overcome this limitation, an OTL controller was conceived to effectively assess if a transferred control policy is capable to guarantee acceptable performance within a reasonable amount of time thus enhancing the scalability of DRL control strategies in real-world context.

The proposed OTL control strategy was benchmarked against an RBC, an offline DRL controller (i.e., which corresponds to the same control strategy employed during the training phase of the source DRL agent) and online DRL controller. The proposed solution showed excellent performance on the nineteen considered target buildings. The OTL controllers were capable to reduce electricity cost and to enhance indoor air temperature control with respect to RBC and online DRL controllers.

The online DRL control strategy was conceived to emulate the implementation of a control agent with no prior training in a real building. Considering that the development of a simulation model of the controlled environment to perform pre-training is not required, this strategy represents the fairest comparison for the proposed OTL approach to highlight the benefits of applying transfer learning.

Several KPIs were introduced in literature reviews on transfer learning regardless of the application domain and type of transfer model. However, for building control systems many Key Performance Indicators (KPIs) are problem-dependent, so can not be used directly and should be readjusted according to each case study. The KPI *Performance with fixed number of epochs* defined in [65] can be employed to compare the performance of OTL and online DRL controllers. Therefore, in this paper this metric was adopted to assess the performance comparison between the online DRL agent and OTL at the end of the single episode in which they were evaluated. This KPI can be computed per each target building (and separately in terms of total electricity cost and cumulated sum of temperature violations) as the relative percentage difference between the performance of the OTL and online DRL controllers reported in Table 8.

The offline DRL controller was developed to emulate the performance of an agent pre-trained offline for several episodes. This strategy performs better than the OTL strategy since it could learn a mapping between states and actions that is more effective considering the opportunity of interacting with the building for a longer time. However, the modeling effort required to produce a model representing the environment to be controlled for the offline DRL controller constitutes a drawback that prohibits its deployment in real buildings even though it performs better than the other controllers. The development of a building surrogate model demands an amount of time not compatible with the real-time implementation of DRL controller, as well as in-depth domain expertise.

In the present work, the offline DRL agent was trained from scratch and requires several training episodes (i.e., several cooling season) to achieve the same performance level that the online TL achieves in one cooling season. To this end, Figure 15 shows a lollipop chart specifying for each target building the number of episodes needed for the offline DRL to perform as well as the OTL. Target buildings were arranged in ascending order with respect to the degree of change in weather conditions from the source building (i.e., Turin, Paris, Helsinki, Palermo). This analysis suggests that OTL implementation was more effective as the number of episodes required for offline DRL to achieve the same performance as OTL increases. It can be observed that offline DRL requires a higher number of training episodes to achieve similar performance to OTL when the climatic conditions of the target buildings are more similar to those of the source building. In fact, the offline DRL controller was pre-trained for almost 30 episodes to achieve performance in

terms of electricity costs and indoor temperature conditions equal to OTL for target buildings located in Turin and Paris (indicated in Figure 15 respectively in dark green and light green). Conversely, the implementation of OTL was less effective as weather condition differences increase, as in the case of target buildings located in Helsinki (in blue) or Palermo (in orange), since the offline DRL control strategy was trained for a lower number of episodes to achieve OTL performance.

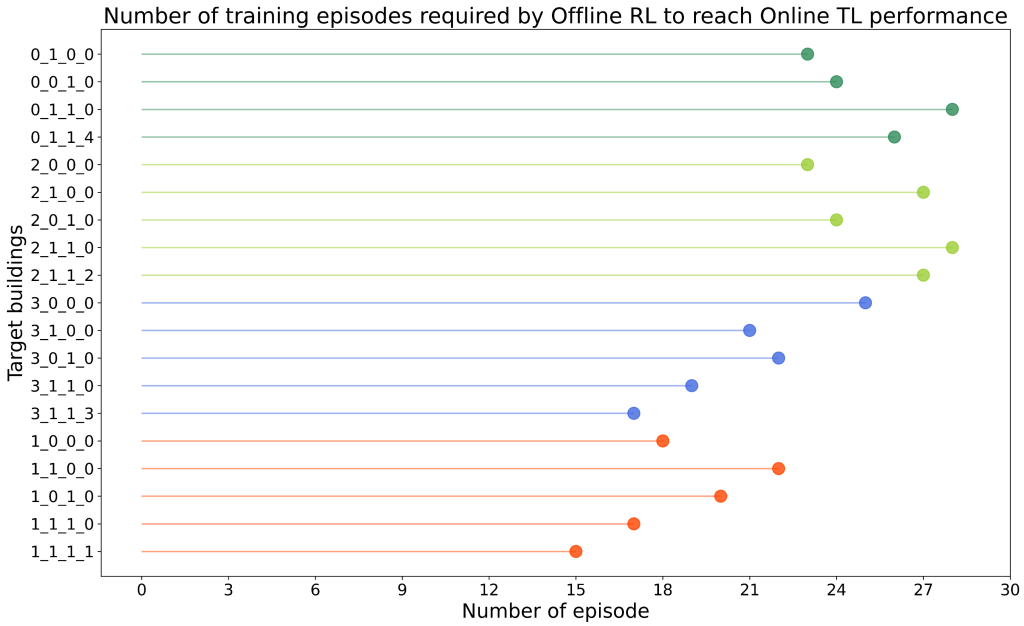

Figure 15: Number of training episodes required by offline DRL to reach OTL performance

As a result, climatic differences have the greatest impact on the quality of the transfer learning process considering that they impact on the magnitude and distribution of cooling load [46]. Therefore, the definition of building archetypes by climate type could be useful to transfer the control policy between buildings of the same group, improving the performance of the developed knowledge sharing methodology.

Eventually, the results obtained demonstrate that the proposed OTL methodology can lead to DRL agents performing better that their RBC counterparts while considerably reducing the implementation effort compared to the offline DRL training framework.

8. Conclusion

The present paper proposes an OTL strategy for enhancing the generalizability and scalability of DRL controllers in buildings. The proposed methodology was employed to test the performance of homogeneous transductive transfer learning, considering that DRL controllers operate in office buildings with the same geometry and energy system but different weather conditions, electricity price schedules, occupancy patterns and building thermophysical properties. This application exploits a simulation environment in which the BCVTB operates as middleware between EnergyPlus and Python. First of all, a DRL agent based on discrete Soft Actor-Critic was developed for the control of a cooling system consisting of an electric chiller and a cold thermal storage in a source building. The objective of the proposed controller is to maintain adequate indoor temperature conditions during occupancy hours while reducing electricity cost with respect to a RBC, through the management of the operation mode of the cooling system and deciding whether to supply cooling energy to the building. An automated procedure was performed during the training phase of the DRL controller in source building to identify the best configuration of hyperparameters. As required in the OTL framework, the best pre-trained agent was transferred to several target buildings, initializing the weights of the networks that approximate the control policy for advanced controller. Then, the agent is further fine-tuned to update the control policy in relation to the boundary conditions of each building. The performances of the OTL agent were benchmarked with those of the RBC and two DRL control strategies, offline DRL and online DRL. The best-trained agent on source building was more effective than the RBC and provides a 20 % reduction in total electricity cost while enhancing the indoor temperature conditions through a 69 % reduction in cumulated sum of temperature violations compared to the RBC. Therefore, the implementation of the OTL strategy on the considered target buildings led to an average reduction in cumulated sum of temperature violations of 50 % and total electrical cost of 10 % compared to a RBC, as well as being more efficient than the online DRL controller with electricity cost savings of between 20 % and 40 % and an average reduction in the cumulated sum of temperature violations of more than 80 %. Conversely, the OTL agent performed worse than the offline DRL controller, with total electricity cost and cumulated temperature violations during occupancy hours higher than those of the offline DRL agent with an average of 4 % and 80 %. However,

this comparison is not fair, as the performance of the OTL agent is evaluated on a single episode, while that of the offline DRL controller is the result of an offline pre-training process on 30 episodes of an agent trained from scratch on each target building. This feature constitutes the major advantage of using transfer learning since the definition of a surrogate model of the environment to pre-train the control policy is not required as in the case of the offline DRL agent.

Future works will focus on the following aspects:

- Evaluation of OTL on more complex case studies, including the generation from renewable energy sources and batteries and extending the analyzed period to the heating season.
- Comparison of the OTL with an advanced model-based optimization method that ensures optimal solution instead of the rule-based controller. In that case, the TL process for a model-based controller would also involve the transfer of a model of system dynamics.
- Exploitation of the proposed methodology for heterogeneous and/or inductive transfer learning, assessing the performance of the knowledge sharing process when control policy is transferred between DRL controllers operating in different domains (e.g., different energy systems or different buildings) or having different objective functions.
- Definition of robust metrics and KPIs to benchmark transfer learning performance and to quantify the similarity between source and target buildings to avoid negative transfer learning, possibly leading to worse performance in advanced controllers than in the case without transfer.
- Development of an accurate simulation environment, modeling the energy system through Modelica and integrating it with the building model developed in EnergyPlus and the control system developed in Python through the use of Spawn of EnergyPlus [79]. The use of Spawn enhances the accuracy of the simulation results by providing semi-realistic performance, since the energy system in Modelica and the physical model of the building in EnergyPlus are represented in detail.
- Implementation of the proposed transfer learning strategy in a real-world testbed. In this case the development of an infrastructure to

enable the implementation of DRL controllers, as well as their further transfer process, will be required.

References

[1] L. Yang, Z. Nagy, P. Goffin, A. Schlueter, Reinforcement learning for optimal control of low exergy buildings, Applied Energy 156 (2015) 577 – 586. doi:https://doi.org/10.1016/j.apenergy.2015.07.050.

[2] G. Martinopoulos, K. T. Papakostas, A. M. Papadopoulos, A comparative review of heating systems in eu countries, based on efficiency and fuel cost, Renewable and Sustainable Energy Reviews 90 (2018) 687 – 699. doi:https://doi.org/10.1016/j.rser.2018.03.060.

[3] A. Baniasadi, D. Habibi, W. Al-Saedi, M. A. Masoum, C. K. Das, N. Mousavi, Optimal sizing design and operation of electrical and thermal energy storage systems in smart buildings, Journal of Energy Storage 28 (2020) 101186. doi:https://doi.org/10.1016/j.est.2019.101186.

[4] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Applied Energy 269 (2020) 115036. doi:https://doi.org/10.1016/j.apenergy.2020.115036.

[5] Guang Geng, G. M. Geary, On performance and tuning of pid controllers in hvac systems, in: Proceedings of IEEE International Conference on Control and Applications, 1993, pp. 819–824 vol.2. doi:10.1109/CCA.1993.348229.

[6] T. I. Salsbury, A survey of control technologies in the building automation industry, IFAC Proceedings Volumes 38 (1) (2005) 90–100, 16th IFAC World Congress. doi:https://doi.org/10.3182/20050703-6-CZ-1902.01397.

[7] S. Brandi, M. Fiorentini, A. Capozzoli, Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management, Automation in Construction 135 (2022) 104128. doi:https://doi.org/10.1016/j.autcon.2022.104128.

[8] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, A. Bemporad, Model predictive control (mpc) for enhancing building and hvac system

energy efficiency: Problem formulation, applications and opportunities, Energies 11 (3) (2018). doi:10.3390/en11030631.

[9] G. Serale, M. Fiorentini, A. Capozzoli, P. Cooper, M. Perino, Formulation of a model predictive control algorithm to enhance the performance of a latent heat solar thermal system, Energy Conversion and Management 173 (2018) 438–449. doi:https://doi.org/10.1016/j.enconman.2018.07.099.

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

[11] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, M. Morari, Use of model predictive control and weather forecasts for energy efficient building climate control, Energy and Buildings 45 (2012) 15 – 27. doi:https://doi.org/10.1016/j.enbuild.2011.09.022.

[12] G. P. Henze, R. H. Dodier, M. Krarti, Development of a predictive optimal controller for thermal energy storage systems, HVAC&R Research 3 (3) (1997) 233–264. arXiv:https://www.tandfonline.com/doi/pdf/10.1080/10789669.1997.10391376, doi:10.1080/10789669.1997.10391376.

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

[14] G. D. Kontes, G. I. Giannakis, V. Sánchez, P. De Agustin-Camacho, A. Romero-Amorrortu, N. Panagiotidou, D. V. Rovas, S. Steiger, C. Mutschler, G. Gruen, Simulation-based evaluation and optimization of control strategies in buildings, Energies 11 (12) (2018). doi:10.3390/en11123376.

[15] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, 2nd Edition, The MIT Press, 2018.
URL `http://incompleteideas.net/book/the-book-2nd.html`

[16] S. Brandi, A. Gallo, A. Capozzoli, A predictive and adaptive control strategy to optimize the management of integrated energy systems in buildings, Energy Reports 8 (2022) 1550–1567. doi:https://doi.org/10.1016/j.egyr.2021.12.058.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
URL http://dx.doi.org/10.1038/nature14236

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

[19] Z. Zou, X. Yu, S. Ergan, Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network, Building and Environment 168 (2020) 106535. doi:https://doi.org/10.1016/j.buildenv.2019.106535.

[20] Y. Du, H. Zandi, O. Kotevska, K. Kurte, J. Munk, K. Amasyali, E. Mckee, F. Li, Intelligent multi-zone residential hvac control strategy based on deep reinforcement learning, Applied Energy 281 (2021) 116117. doi:https://doi.org/10.1016/j.apenergy.2020.116117.

[21] Y. Wang, K. Velswamy, B. Huang, A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems, Processes 5 (3) (2017). doi:10.3390/pr5030046.

[22] Z. Zhang, A. Chong, Y. Pan, C. Zhang, K. P. Lam, Whole building energy model for hvac optimal control: A practical framework based on deep reinforcement learning, Energy and Buildings 199 (2019) 472 – 490. doi:https://doi.org/10.1016/j.enbuild.2019.07.029.

[23] S. Brandi, M. S. Piscitelli, M. Martellacci, A. Capozzoli, Deep reinforcement learning to optimise indoor temperature control and heating en-

ergy consumption in buildings, Energy and Buildings 224 (2020) 110225. doi:https://doi.org/10.1016/j.enbuild.2020.110225.

[24] J. R. Vázquez-Canteli, S. Ulyanin, J. Kämpf, Z. Nagy, Fusing tensorflow with building energy simulation for intelligent energy management in smart cities, Sustainable Cities and Society 45 (2019) 243 – 257. doi:https://doi.org/10.1016/j.scs.2018.11.021.

[25] G. Pinto, M. S. Piscitelli, J. R. Vázquez-Canteli, Z. Nagy, A. Capozzoli, Coordinated energy management for a cluster of buildings through deep reinforcement learning, Energy 229 (2021) 120725. doi:https://doi.org/10.1016/j.energy.2021.120725.

[26] G. Pinto, D. Deltetto, A. Capozzoli, Data-driven district energy management with surrogate models and deep reinforcement learning, Applied Energy 304 (2021) 117642. doi:https://doi.org/10.1016/j.apenergy.2021.117642.

[27] Modelica Association, Modelica® - a unified object-oriented language for physical systems modeling. Tutorial (Dec. 2000).
URL `http://www.modelica.org/documents/ModelicaTutorial14.pdf`

[28] D. B. Crawley, L. K. Lawrie, C. O. Pedersen, F. C. Winkelmann, Energy plus: energy simulation program, ASHRAE journal 42 (4) (2000) 49–56.

[29] G. Pinto, Z. Wang, A. Roy, T. Hong, A. Capozzoli, Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives, Advances in Applied Energy 5 (2022) 100084. doi:https://doi.org/10.1016/j.adapen.2022.100084.

[30] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. on Knowl. and Data Eng. 22 (10) (2010) 1345–1359. doi:10.1109/TKDE.2009.191.

[31] F. L. Da Silva, A. H. R. Costa, A survey on transfer learning for multiagent reinforcement learning systems, J. Artif. Int. Res. 64 (1) (2019) 645–703. doi:10.1613/jair.1.11396.

[32] T. Peirelinck, H. Kazmi, B. V. Mbuwir, C. Hermans, F. Spiessens, J. Suykens, G. Deconinck, Transfer learning in demand response: A review of algorithms for data-efficient modelling and control, Energy and AI 7 (2022) 100126. doi:https://doi.org/10.1016/j.egyai.2021.100126.

[33] Y. Himeur, M. Elnour, F. Fadli, N. Meskin, I. Petri, Y. Rezgui, F. Bensaali, A. Amira, Next-generation energy systems for sustainable smart cities: Roles of transfer learning, Sustainable Cities and Society 85 (2022) 104059. doi:https://doi.org/10.1016/j.scs.2022.104059.

[34] J. Leon-Malpartida, J. D. Farfan-Escobedo, G. E. Cutipa-Arapa, A new method of classification with rejection applied to building images recognition based on transfer learning, in: 2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), 2018, pp. 1–4. doi:10.1109/INTERCON.2018.8526392.

[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2014). doi:10.48550/ARXIV.1409.1556.

[36] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, nature 529 (7587) (2016) 484–489.

[37] S. Singh, M. Kearns, D. Litman, M. Walker, Reinforcement learning for spoken dialogue systems, Advances in neural information processing systems 12 (1999).

[38] Z. Ren, X. Wang, N. Zhang, X. Lv, L.-J. Li, Deep reinforcement learning-based image captioning with embedding reward, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 290–298.

[39] C. Fan, Y. Lei, Y. Sun, M. S. Piscitelli, R. Chiosa, A. Capozzoli, Data-centric or algorithm-centric: Exploiting the performance of transfer learning for improving building energy predictions in data-scarce context, Energy 240 (2022) 122775. doi:https://doi.org/10.1016/j.energy.2021.122775.

[40] P. Oliveira, B. Fernandes, C. Analide, P. Novais, Forecasting energy consumption of wastewater treatment plants with a transfer learning approach for sustainable cities, Electronics 10 (10) (2021). doi:10.3390/electronics10101149.

[41] X. Fang, G. Gong, G. Li, L. Chun, W. Li, P. Peng, A hybrid deep transfer learning strategy for short term cross-building energy prediction, Energy 215 (2021) 119208. doi:https://doi.org/10.1016/j.energy.2020.119208.

[42] W.-H. Chen, P.-C. Cho, Y.-L. Jiang, Activity recognition using transfer learning, Sens. Mater 29 (7) (2017) 897–904.

[43] B. Pardamean, H. H. Muljo, T. W. Cenggoro, B. J. Chandra, R. Rahutomo, Using transfer learning for smart building management system, Journal of Big Data 6 (1) (2019) 1–12.

[44] Y. Chen, Z. Tong, Y. Zheng, H. Samuelson, L. Norford, Transfer learning with deep neural networks for model predictive control of hvac and natural ventilation in smart buildings, Journal of Cleaner Production 254 (2020) 119866. doi:https://doi.org/10.1016/j.jclepro.2019.119866.

[45] M. Demianenko, C. I. De Gaetani, A procedure for automating energy analyses in the bim context exploiting artificial neural networks and transfer learning technique, Energies 14 (10) (2021). doi:10.3390/en14102956.

[46] G. Pinto, R. Messina, H. Li, T. Hong, M. S. Piscitelli, A. Capozzoli, Sharing is caring: An extensive analysis of parameter-based transfer learning for the prediction of building thermal dynamics, Energy and Buildings (2022) 112530,doi:https://doi.org/10.1016/j.enbuild.2022.112530.

[47] L. Fan, J. Zhang, Y. He, Y. Liu, T. Hu, H. Zhang, Optimal scheduling of microgrid based on deep deterministic policy gradient and transfer learning, Energies 14 (3) (2021). doi:10.3390/en14030584.

[48] P. Lissa, M. Schukat, M. Keane, E. Barrett, Transfer learning applied to drl-based heat pump control to leverage microgrid energy efficiency, Smart Energy 3 (2021) 100044. doi:https://doi.org/10.1016/j.segy.2021.100044.

[49] B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and Autonomous Systems 57 (5) (2009) 469–483. doi:https://doi.org/10.1016/j.robot.2008.10.024.

[50] J. Xu, Z. Li, G. Du, Q. Liu, L. Gao, Y. Zhao, A transferable energy management strategy for hybrid electric vehicles via dueling deep deterministic policy gradient, Green Energy and Intelligent Transportation (2022) 100018.doi:https://doi.org/10.1016/j.geits.2022.100018.

[51] P. Lissa, M. Schukat, E. Barrett, Transfer learning applied to reinforcement learning-based hvac control, SN Computer Science 1 (3) (2020) 1–12.

[52] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, W. Li, X. Shi, Cross temporal-spatial transferability investigation of deep reinforcement learning control strategy in the building hvac system level, Energy (2022) 125679.doi:https://doi.org/10.1016/j.energy.2022.125679.

[53] S. Xu, Y. Wang, Y. Wang, Z. O'Neill, Q. Zhu, 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, BuildSys '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 230–239. doi:10.1145/3408308.3427617.

[54] T. Zhang, A. K. G. S, M. Afshari, P. Musilek, M. E. Taylor, O. Ardakanian, Diversity for transfer in learning-based control of buildings, in: Proceedings of the Thirteenth ACM International Conference on Future Energy Systems, e-Energy '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 556–564. doi:10.1145/3538637.3539615.

[55] N. Tsang, C. Cao, S. Wu, Z. Yan, A. Yousefi, A. Fred-Ojala, I. Sidhu, Autonomous household energy management using deep reinforcement learning, in: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 2019, pp. 1–7. doi:10.1109/ICE.2019.8792636.

[56] X. Zhang, X. Jin, C. Tripp, D. J. Biagioni, P. Graf, H. Jiang, Transferable reinforcement learning for smart homes, in: Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities, 2020, pp. 43–47.

[57] B. V. Mbuwir, K. Paridari, F. Spiessens, L. Nordström, G. Deconinck, Transfer learning for operational planning of batter-

ies in commercial buildings, in: 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2020, pp. 1–6. doi:10.1109/SmartGridComm47815.2020.9303016.

[58] P. Zhao, S. C. Hoi, J. Wang, B. Li, Online transfer learning, Artificial Intelligence 216 (2014) 76–102. doi:https://doi.org/10.1016/j.artint.2014.06.003.

[59] T. Grubinger, G. C. Chasparis, T. Natschläger, Generalized online transfer learning for climate control in residential buildings, Energy and Buildings 139 (2017) 63–71. doi:https://doi.org/10.1016/j.enbuild.2016.12.074.

[60] P. Christodoulou, Soft actor-critic for discrete action settings, CoRR abs/1910.07207 (2019). arXiv:1910.07207.
URL http://arxiv.org/abs/1910.07207

[61] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2623–2631. doi:10.1145/3292500.3330701.

[62] R. Bellman, Dynamic programming, Science 153 (3731) (1966) 34–37. arXiv:https://science.sciencemag.org/content/153/3731/34.full.pdf, doi:10.1126/science.153.3731.34.

[63] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, S. Levine, Soft actor-critic algorithms and applications (2019). arXiv:1812.05905.

[64] M. E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: A survey, Journal of Machine Learning Research 10 (56) (2009) 1633–1685.
URL http://jmlr.org/papers/v10/taylor09a.html

[65] Z. Zhu, K. Lin, A. K. Jain, J. Zhou, Transfer learning in deep reinforcement learning: A survey (2020). doi:10.48550/ARXIV.2009.07888.
URL https://arxiv.org/abs/2009.07888

[66] Arera - andamento del prezzo dell'energia elettrica per il consumatore domestico tipo in maggior tutela, https://www.arera.it/it/dati/eep35.htm, accessed: 2022-08-23.

[67] D2.2 european climate zones and bio-climatic design requirements - report: Pvsites-wp2-t21-d22-m03-bear-20160831-v01, https://www.pvsites.eu/downloads/category/project-results?page=4, accessed: 2022-08-23 (2016).

[68] K. Tsikaloudaki, K. Laskos, D. Bikas, On the establishment of climatic zones in europe with regard to the energy performance of buildings, Energies 5 (1) (2012) 32–44. doi:10.3390/en5010032.

[69] Austin energy. electricity tariff pilot programs, https://austinenergy.com/ae/, accessed: 2022-08-23.

[70] Ministry of economic development - interministerial decree of 26 june 2015: Application of energy performance calculation methodologies and definition of prescriptions and minimum requirements for buildings. appendix a: General criteria and requirements for the energy performance of buildings, https://www.mise.gov.it/index.php/it/normativa/decreti-interministeriali/decreto-interministeriale-26-giugno-2015-applicazione-delle-metodologie-di-calcolo-delle-prestazioni-energetiche-e-definizione-delle-prescrizioni-e-dei-requisiti-minimi-degli-edifici?cldee=ZW5lcmdpYS5kZW1hcmNvQGxpYmVyby5pdA%3D%3D&urlid=0?hitcount=0, accessed: 2022-08-23 (2015).

[71] Ministry of economic development - interministerial decree of 26 june 2015: Application of energy performance calculation methodologies and definition of prescriptions and minimum requirements for buildings. appendix b: Specific requirements for existing buildings subject to energy rehabilitation, https://www.mise.gov.it/index.php/it/normativa/decreti-interministeriali/decreto-interministeriale-26-giugno-2015-applicazione-delle-metodologie-di-calcolo-delle-prestazioni-energetiche-e-definizione-delle-prescrizioni-e-dei-requisiti-minimi-degli-edifici?cldee=ZW5lcmdpYS5kZW1hcmNvQGxpYmVyby5pdA%3D%3D&urlid=0?hitcount=0, accessed: 2022-08-23 (2015).

[72] D. Bienvenido-Huertas, M. Oliveira, C. Rubio-Bellido, D. Marín, A comparative analysis of the international regulation of thermal properties in building envelope, Sustainability 11 (20) (2019). doi:10.3390/su11205574.

[73] A. Huynh, R. Dias Barkokebas, M. Al-Hussein, C. Cruz-Noguez, Y. Chen, Energy-efficiency requirements for residential building envelopes in cold-climate regions, Atmosphere 12 (3) (2021). doi:10.3390/atmos12030405.

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

[75] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, Curran Associates Inc., Red Hook, NY, USA, 2011, p. 2546–2554.

[76] Q. Xin, 3 - optimization techniques in diesel engine system design, in: Q. Xin (Ed.), Diesel Engine System Design, Woodhead Publishing, 2013, pp. 203–296. doi:https://doi.org/10.1533/9780857090836.1.203.

[77] M. Zelany, A concept of compromise solutions and the method of the displaced ideal, Computers & Operations Research 1 (3) (1974) 479–496. doi:https://doi.org/10.1016/0305-0548(74)90064-1.

[78] S. L. Smith, P.-J. Kindermans, C. Ying, Q. V. Le, Don't decay the learning rate, increase the batch size, arXiv preprint arXiv:1711.00489 (2017).

[79] M. Wetter, K. S. Benne, A. Gautier, T. S. Nouidui, A. Ramle, A. Roth, H. Tummescheit, S. G. Mentzer, C. Winther, Lifting the garage door on spawn, an open-source bem- controls engine, in: 2020 Building Performance Modeling Conference and SimBuild co-organized by ASHRAE and IBPSA-USA, 2020.