Carnegie Mellon University

Ph.D. Thesis

A Reinforcement Learning Approach for Whole Building Energy Model Assisted HVAC Supervisory Control

Zhang, Zhiang

Doctoral Committee:

Khee Poh Lam, Chair

Professor Emeritus, Ph.D., FRIBA, IBPSA-Fellow

School of Architecture, Carnegie Mellon University

Mario Berges

Professor, Ph.D.

Civil and Environmental Engineering, Carnegie Mellon University

Gianni Di Caro

Associate Teaching Professor, Ph.D.

School of Computer Science, Carnegie Mellon University-Qatar

Adrian Chong

Assistant Professor, Ph.D.

School of Design and Environment, National University of Singapore

September 18, 2019

A dissertation submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy in Building Performance and Diagnostics

at the School of Architecture, Carnegie Mellon University

To myself, to my family, to everyone

士不可以不弘毅，任重而道远

Acknowledgements

Chinese culture encourages people to pursue knowledge to make positive impacts on society. "Pursuing knowledge" is hence sublime and respected. When I decided to do the Ph.D. study three years ago, my primary motivation was to pursue the most advanced knowledge in one field.

On my road to knowledge, I must first acknowledge Prof. Khee Poh Lam. Prof. Lam is my advisor, and mentor. He advised on my research as an academic advisor, but more importantly, he influenced my life like a mentor and a friend. As a world-renowned expert in my area, his guidance is crucial for my research without any questions. The research topic he recommended for me five years ago has prove to be pioneering and significant. But I appreciate more for his influence on my life. I am deeply impressed by his enthusiasm for work, passion for life, and care for the family. He helps and encourages me to get through the hard Ph.D. time. There is so much I need to learn from him. I wish I could be his student for more years.

I also appreciate the valuable guidance from my Ph.D. committee, including Prof. Mario Berges, Prof. Gianni Di Caro, and Prof. Adrian Chong. They are top scientists from civil engineering, computer science, and building engineering. They pushed me to rethink my research from different perspectives, and helped me to build a scientifically solid interdisciplinary research methodology. I am also grateful to them for polishing my thesis writing. We spent hours together just to determine the usage of a single word that may confuse readers. I am deeply impressed by their rigorous attitude for research. I would like to give my special thanks to Prof. Chong, who taught me Bayesian calibration step-by-step, helped me settle down during my visit to Singapore, and introduced me to the potential research opportunities from the National University of Singapore.

I would like to express my gratitude and love to my wife Linan Zhang. We have known each other for more than 10 years. She gives me a warm and sweet home that charges me when I lose power. Every day after work, we prepare dinner together, take a walk in our neighborhood, watch

TV dramas and so many other sweet family activities. That is just the best time of a day. Our first child was born in the second year of my Ph.D. study. Family is always the first reason for my hard work.

The support from friends and colleagues is also of significant importance in my journal of pursuing the Ph.D. I would like to thank Chenlu Zhang, Adrian Chong, Yuqi Pan, Omer Karaguzel, Weili Xu, Chao Ding, Jie Zhao, Siliang Lu, Jiarong Xie and all the friends from Carnegie Mellon University and the National University of Singapore. They are outstanding researchers in different fields. It is my great honor and luck to work with the best people.

Last but not least, I would like to thank China Scholarship Council and Prof. Khee Poh Lam for the financial support of my Ph.D. study. With their generous support, I can focus on the research without any distractions.

致谢

“学不可以已”。

中国的传统入世思想是一贯鼓励人学习的。“腹有诗书气自华”。追求知识是崇高的，是值得尊敬的。因此，三年前我决定攻读博士的主要动因，便是受中国传统思想的耳濡目染，想要追求某个领域的最高学术学位。

“古之学者必有师”。在我追求学术的道路上，我的导师林棋波教授是我首先要致谢的人。西方教育体系里的“导师（advisor）”和东方文化中的“老师”并不能等同。“导师”更侧重于对于某个学科的客观的指导，而“老师”更侧重于对于一个人的塑造。林教授对于我，是一位导师，更是一位老师。林教授是我所在领域的权威专家，他的指导对于我学术道路的重要性，我无须赘言。他在五年前给我提出的研究方向，目前证明是有极大的前瞻性和重要性的。学术之外，林教授的言传身教对我有更深远的影响。他在工作里的活力、在生活中的情趣、对家庭的责任无一不是我的楷模。他对学生关心并且爱护，帮助并鼓励我顺利度过艰苦的博士生涯。林教授的人生高度，我难以望其向背。“高山仰止，景行行止，虽不能至，心向往之”。

“兼听则明”。我要感谢我的博士委员会成员，包括林棋波教授、Mario Berges 教授、Gianni Di Caro 教授和 Adrian Chong 教授。他们是来自于建筑、土木工程、计算机科学和建筑工程四个不同领域的顶尖专家。他们督促我以不同的专业角度审视我的研究，并帮助我建立起一套严谨科学的跨学科研究方法。同时，我要感谢他们帮助我不断打磨我的论文，细致到每一个字的选用。他们严谨的治学精神让我肃然起敬。我想要特别感谢 Adrian Chong 教授，感谢他手把手地指导我贝叶斯模型校准的方法，感谢他在我初到新加坡访学时对我的帮助，并且感谢他无私地分享他在新加坡国立大学的科研资源。

“修身，齐家，治国，平天下”。我想感谢我的夫人张莉楠，我们相识相恋十多年，一起组建了一个温馨、自由的家庭。每天从实验室下班回家，我们一起准备晚饭、一起饭后散步、一起追剧.....家庭的温馨让我每天的工作都充满活力。我们的第一个孩子也在我博士期间出生。为爱人、为孩子创造美好生活，是我奋斗的源动力。

“三人行，必有我师焉”。与朋友、同学、同僚的相互学习与支持，也是我攻读博士期间的重要部分。张晨露、Adrian Chong、潘宇琦、Omer Karaguzel、徐为力、丁超、赵杰、卢思亮、解加荣和各位在卡耐基梅隆大学和新加坡国立大学的同事和朋友。他们都是在各自领域极为出色的研究者。与他们的合作与交流让我的研究更加坚实有力。他们对我的鼓励也让我在博士期间越来越有自信。能与他们一起共事是我的幸运。

我还要感谢中国国家留学基金委和林棋波教授的资金支持。他们慷慨地资助了我的博士研究，让我在没有任何金钱的压力下专注完成博士学业。

最终，我想感谢毛主席的诗词与文章。其中的革命乐观主义精神深深地感染了我，鼓励我直面困难、挑战困难、解决困难。

“西风烈，长空雁叫霜晨月。

霜晨月，马蹄声碎，喇叭声咽。

雄关漫道真如铁，而今迈步从头越。

从头越，苍山如海，残阳如血。”(毛泽东《忆秦娥 · 娄山关》，写于 1935 年长征途中)

Abstract

Buildings account for a significant portion of the total energy consumption of many countries. Energy efficiency is one of the primary objectives of today's building projects. Whole building energy model (BEM), a physics-based modeling method for building thermal and energy behaviors, is widely used by building designers to predict and improve the energy performance of building design. BEM also has potential for developing supervisory control strategies for heating, ventilation and air-conditioning (HVAC) systems. The BEM-derived control strategies may significantly improve HVAC energy efficiency compared to the commonly-used rule-based control strategies.

However, it is challenging to use BEM for HVAC control. This is because, firstly, BEM is a high-order model so classical model-based control methods cannot be directly applied. Heuristic search algorithms, such as genetic algorithm, are usually used for BEM-based control optimization. Secondly, BEM is computationally-intensive compared to other black-box or grey-box models, which limits its application for large-scale control optimization problems.

Model-free reinforcement learning (RL) is an alternative method to use BEM for HVAC control. Model-free RL is a "trial-and-error" learning method that is applicable for any complex systems. As a result, BEM can be used as a simulator to train an RL agent offline to learn an energy-efficient supervisory control strategy. However, reinforcement learning for HVAC control has not been adequately studied. Most existing studies are based on over-simplified HVAC systems and a limited number of experiment scenarios.

This study develops a BEM-assisted reinforcement learning framework for HVAC supervisory control for energy efficiency. The control framework uses a design-stage BEM to "learn" a control strategy via model-free RL. The RL agent is a neural network model which performs as a function approximator. Through computer simulations, the control framework is evaluated in different scenarios covering four typical commercial HVAC systems, four climates, and two building thermal

mass levels. The RL-trained control strategies are also evaluated for "versatility", i.e., the tolerance for the variations of HVAC operational conditions. Multiple "perturbed" simulators are created for this purpose, with varying weather conditions, occupancy and plug-load schedules, and indoor air temperature setpoint schedules.

The control framework has achieved better-then-baseline control performance in a variable-air-volume (VAV) system (a common type of air-based secondary HVAC system) for both cooling and heating under different climates and building thermal mass levels. Compared to a baseline rule-based control strategy, the RL-trained strategies can achieve obvious energy-savings and less "setpoint notmet time" (i.e., the cumulative time that indoor air temperature setpoints are not met). Also, the RL-trained strategies can tolerate the variations in weather conditions and occupancy/plug-load schedules. However, the RL-trained control strategies have worse-than-baseline energy performance if indoor air temperature setpoint schedules are significantly changed.

The control framework has also achieved reduced heating demand and improved-or-similar thermal comfort (compared to a baseline rule-based control) for a slow-response radiant heating system in all the experiment scenarios. The RL-trained strategies have also achieved improved control performance in different perturbed simulators. However, the reward function must include a specially-designed heuristic to deal with the slow thermal response and the imperfect energy metric of this system. The heuristic encourages low supply water temperature setpoint values and reward increasing trends of the predicted mean vote (PMV) if it is below the setpoint. This indicates that the reward function design is crucial for the control performance of this control framework.

Control performance may be poor if the reward function is over-complicated, as shown in the experiments related to a multi-chiller chilled water system. The reward function for this system consists of three complicated penalty functions corresponding to three operational constraints, including the chiller cycling time, the chiller partial-load-ratio, and the system supply water temperature. The RL-trained control strategies have violated some operational constraints significantly, and only achieved a limited amount of energy savings.

This thesis also studied the effects of the neural network model (the RL agent function approximator) complexity on the control and convergence performance of the control framework. It is found that a complex neural network model does not necessarily lead to better control performance compared to a simple neural network model. A complex neural network model may make the reinforcement learning hard to converge. Thus, "deep" reinforcement learning is not always a suitable choice, even though it is a popular concept in recent literature. As a general guideline, this study

recommends using a narrow and shallow non-linear neural network model for the control framework.

In future work, the control framework should be evaluated in more scenarios, such as more types of HVAC systems and more climate zones. It is also necessary to conduct a more comprehensive versatility analysis for a trained RL control policy. Future work should also develop an adaptive RL control method that could self-adapt to the changing characteristics of an HVAC system. Last but not least, theoretical investigations are needed to guide the future development of the control framework.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Building Energy Efficiency

Buildings are one of the major energy consumers in many countries. For example, buildings (including both residential and commercial buildings) account for nearly 39% of the total energy consumption in the U.S. (The U.S. Energy Information Administration, 2017), 19% of the total energy consumption in China (Huo et al., 2018), and 50% of the total electricity consumption in Singapore (Energy Market Authority of Singapore, 2018). Hence, building energy efficiency has been a popular topic in both research and practice for years.

The energy efficiency of today's buildings has been significantly improved thanks to the increasingly stringent codes and standards. Green building certification programs also motivate building owners and designers to pursue higher building energy efficiency. As a result, buildings now have better-optimized orientations, more insulated envelopes, better energy-efficient equipment, larger-scale sensor networks, and other improvements.

However, compared to the advancements in other areas, most buildings still use simple control strategies, e.g., fixed operation schedules and rule-based control. Control strategies can significantly affect building energy efficiency, especially for commercial heating, ventilation and air-conditioning (HVAC) systems. Hence, building energy efficiency can be further improved if HVAC systems adopt more sophisticated control strategies. It can be a "free" energy efficiency measure since it does not change building structures and hardware.

1.1.2 Control of HVAC Systems

The control of a commercial HVAC system can be divided into two levels, supervisory level and local level (Wang and Ma, 2007), as shown Figure 1.1.

- Supervisory level: Control at this level uses building observations to determine the high-level setpoints and operation states in an HVAC system. For example, a supervisory level control strategy may use outdoor air temperature and indoor occupancy state to determine system supply air temperature setpoints or a chiller's on/off state. Note that the setpoints and operation states are virtual points in a control system which are not directly related to

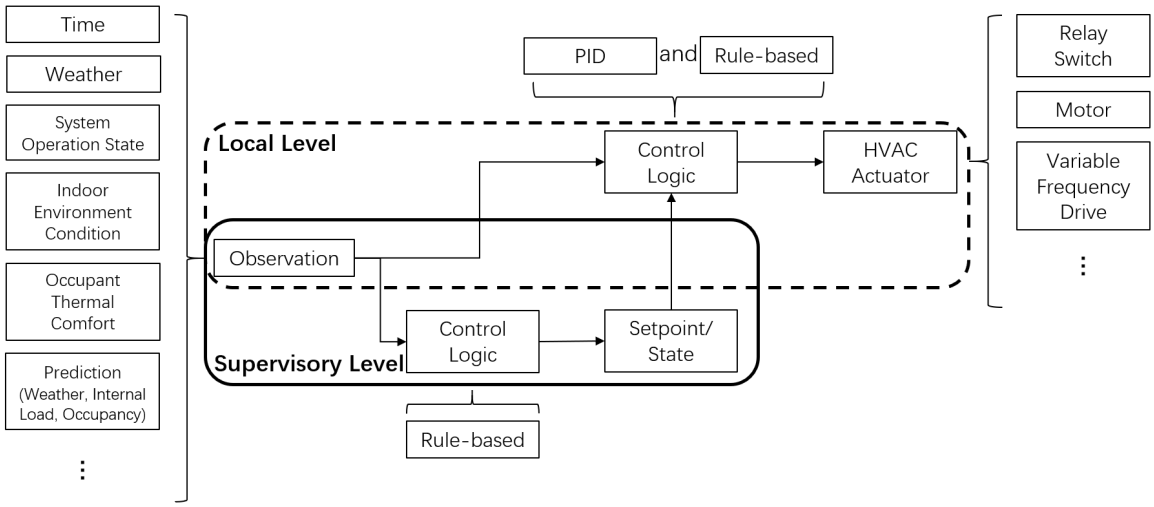

Figure 1.1: Supervisory and Local Level Control in HVAC Systems

any physical HVAC components.

In current practice, the supervisory level control is usually rule-based (i.e., based on static if-then-else rules). The control design is often determined by engineers' experience. However, an HVAC system is an integration of a plethora of components, and the detailed design of each HVAC system is unique. Thus, static and experience-based control strategies are not energy-efficient solutions for HVAC operations. Recent research tries to develop better supervisory control strategies to improve HVAC energy efficiency.

- Local level: Control at this level determines the state of an HVAC actuator based on building observations, the supervisory level setpoints, and the supervisory level operation states. The local level control aims to make an HVAC system meet the supervisory setpoints. For example, a local level controller controls an air damper to minimize the error between indoor air temperature and its setpoint. Note that the local level control is directly related to the operations of HVAC physical components. Rule-based control (RBC) and Proportional-Integral-Derivative (PID) control are common control strategies.

This thesis focuses on the supervisory-level control for HVAC energy efficiency. This is because, firstly, it is easier to manipulate the supervisory level setpoints since these setpoints do not directly control any HVAC actuators; secondly, compared to the local level control, the supervisory level control has a wider impact on HVAC operations since the change of one setpoint may affect multiple HVAC actuators.

1.1.3 Whole Building Energy Model

Whole building energy model (BEM) is a physics-based simulation program to predict the sub-hourly thermal and energy behaviors of a building, such as heating and cooling loads, energy consumption, indoor environmental conditions and even renewable energy generation. It does not require building operation data and is flexible to model different buildings and systems.

BEM is widely used for design decision support. Building designers use BEM to predict the energy performance of their design and compare different design alternatives to improve building energy efficiency. The popularity of BEM in building design is probably driven by the requirements in various energy efficiency standards and certification programs, such as LEED, BREEAM, ASHRAE 90.1, California Title 24, etc. For example, the latest version of LEED Building Design and Construction (The U.S. Green Building Council, 2019) (a green building certification program) suggests building designers to use building energy models to calculate the energy-saving potential of their design. ASHRAE 90.1 (American Society of Heating, Ventilating, and Air Conditioning Engineers, 2016) (a building energy efficiency standard in the U.S.) allows building designers to use BEMs to meet standard compliance requirements.

Software Tools for Building Energy Modeling In the U.S., the development of building energy modeling tools dates back to 1950s (Kusuda, 1999). Initially, computer programs were developed to only solve some simple and specific building simulation problems, such as calculating the overshadowing of a building and simulating the thermal environment of an underground shelter (Kusuda, 1999). In the early 1970s, a general-purpose tool NBSLD (Kusuda, 1976) was developed to calculate building heating and cooling loads based on the detailed heat balance calculation. NBSLD was then modified for annual energy simulation, but it can only model a single zone with a simple HVAC system (Kusuda, 1999). In the later 1970s, CAL-ERDA (Winkelmann et al., 1977) was developed to simulate the annual energy performance of a multi-zone building with HVAC systems. CAL-ERDA uses a simplified method, weighting factors, to calculate the heating and cooling loads of a building, which is not as sophisticated as the heat balance method in NBSLD. CAL-ERDA then evolved into DOE-2 (James J. Hirsch Associates, 2019), which is still one of the most popular BEM tools in the U.S. EnergyPlus (The U.S. Department of Energy, 2019a) was then launched in the later 1990s to combine the best features of the existing simulation tools at that time (e.g., DOE-2, BLAST-the successor of NBSLD). EnergyPlus also provides new features, such as integrated simulation of building thermal loads and HVAC systems (Crawley et al., 1998). After over 20 years

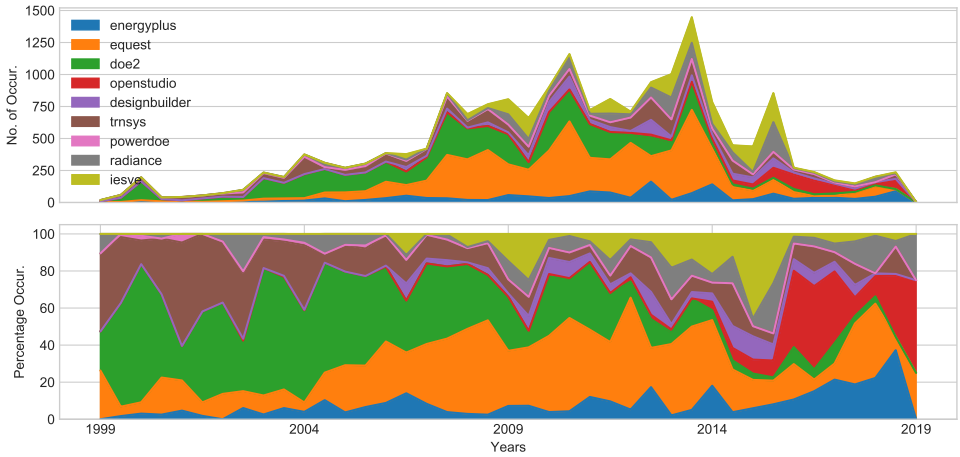

Figure 1.2: Mentioned Building Simulation Tools in Bldg-sim Mailing List Over the Years (Miller et al., 2019)

of continuous development, the recent versions of EnergyPlus can adequately model most buildings and HVAC systems. For the future, Spawn-of-EnergyPlus (SOEP) (The U.S. Department of Energy, 2019b) is potentially the next-generation BEM engine, which aims to integrate Modelica modeling language into the EnergyPlus workflow. SOEP has a specific goal to make BEM more accessible for building control.

Different BEM tools have different levels of complexity and capability that fit into different purposes. Miller et al. (2019) conducts text mining for BLDG-SIM public email archive (a popular mailing list for the building energy modeling professionals since 1999, managed by onebuilding.org (2019)). They find that EnergyPlus-based tools (including EnergyPlus, OpenStudio, and DesignBuilder) are mentioned more frequently than DOE-2-based tools (including DOE-2, eQUEST, and PowerDOE (James J. Hirsch Associates, 1998)) in recent years, as shown in Figure 1.2. This is probably because EnergyPlus is more accurate for the sub-hourly prediction of building thermal and energy performance. EnergyPlus is arguably the most sophisticated BEM tool in the industry.

1.1.4 Motivation

Buildings account for a significant share of the total energy consumption in many countries, and HVAC systems are major energy consumers in commercial buildings. The energy efficiency of HVAC systems can be affected by their supervisory control strategies. However, most HVAC systems use simple rules-based control as the supervisory control strategies, which may not be energy-efficient.

Whole building energy model is widely used for design decision support. However, it is rarely used for developing HVAC supervisory control strategies. As a detailed and physics-based model, BEM (simulation)-based building control can potentially deliver a complex control strategy that achieves "habitability, sustainability and feasibility" simultaneously (Mahdavi, 2001). The life-cycle cost of BEM development could also be reduced if BEM is used for HVAC control to further improve the energy efficiency of a building.

Thus, this thesis aims to develop a method to *using BEM for HVAC supervisory control* to improve HVAC energy efficiency. The following sections will review the relevant studies of model-based or model-assisted HVAC control methods for energy efficiency, including model-based predictive control, BEM-based predictive control and reinforcement learning control. As will be shown in the following section, reinforcement learning is selected as the control algorithm of this thesis because of its applicability on complex dynamic models such as BEM.

1.2 Literature Review

1.2.1 Model Predictive Control

Model predictive control (MPC) is one of the most popular model-based control methods for HVAC systems. Hence, it is worth briefly introducing this method and discussing its limitations.

MPC is defined as an optimization problem with the equality and inequality constraints (Fragkiadaki, 2018),

$$\min_{x,\mu} \sum_{k=t}^{T} c_k(x_k, \mu_k) \tag{1.1a}$$

$$s.t.\ x_t = x_t \tag{1.1b}$$

$$x_{k+1} = f(x_k, \mu_k) \quad \forall k \in t, t+1, t+2, ..., T-1 \tag{1.1c}$$

$$g(\mu_t, \mu_{t+1}, ..., \mu_T, x_t, x_{t+1}, ..., x_T) = 0 \tag{1.1d}$$

$$h(\mu_t, \mu_{t+1}, ..., \mu_T, x_t, x_{t+1}, ..., x_T) \leq 0 \tag{1.1e}$$

where x is the state variable, μ is the control variable, c is the cost function, f is the system dynamics (a.k.a. the model), T is the prediction horizon, t is one control time step, x_t is the actual state observation at the time step t, and g and h are the additional equality and inequality constraints such as the thermal comfort constraints and the operational constraints of an HVAC system. The result of the above optimization problem is a control trajectory $\{\mu_t, x_{t+1}, \mu_{t+1}, x_{t+2}, ..., x_T, \mu_T\}$ but only μ_t is actually executed. The optimization problem is repeated for each control time step of a process. This repeated optimization mechanism is also called receding horizon because the prediction horizon is moving forward at each control time step.

MPC was initially popular in the oil processing industry in 1970s (García et al., 1989), and the research of its application in HVAC dates back to the early 1990s (MacArthur and Foslien, 1993). Since MPC has solid theoretical foundations (e.g., optimality guarantee), abundant research studies can be found to use MPC for HVAC supervisory control for energy efficiency, such as controlling supply heating/cooling power setpoint (O'Dwyer et al., 2017; Fielsch et al., 2017; Vana et al., 2014; Coninck and Helsen, 2016; Huang et al., 2015; Li et al., 2015), supply water temperature setpoint (Lindelöf et al., 2015; Killian and Kozek, 2018), supply air temperature setpoint (Liang et al., 2015; Chen et al., 2016; Ma et al., 2014b; Razmara et al., 2015), supply airflow rate setpoint (Liang et al.,

2015; Ma et al., 2014b; Shi et al., 2017), indoor temperature setpoint (Ma et al., 2014a; Yu et al., 2017; West et al., 2014).

Despite of the popularity of MPC, scalability becomes an issue when the optimization solvers of MPC are applied to complex models and cost functions. One example is the linear quadratic regulator which requires a linear model and quadratic cost function. As a result, BEM cannot be directly used for MPC because BEM is a high-order simulation program. Besides, commercial HVAC systems may have very complex dynamics due to their convoluted system configurations. For example, for an all-air based system, the relationship between the air-handling-unit (AHU) supply air temperature (control variable) and the system energy consumption may be nonlinear and even non-continuous because of the nonlinear dynamics of the HVAC components and the complicated interrelationships among their operations. Thus, to use MPC for HVAC supervisory control, control problems must be simplified or workaround methods must be proposed, for example:

- Since the control goal of MPC is to minimize the energy consumption of an HVAC system, some studies directly use HVAC heating/cooling demand as the control variable. However, simplified models have to be used to approximate the nonlinear relationships between HVAC heating/cooling demand and HVAC electricity/natural gas consumption (Fielsch et al., 2017; Vana et al., 2014; Coninck and Helsen, 2016; Huang et al., 2015). Besides, HVAC heating/cooling demand is not a directly controllable setpoint in some HVAC systems. Hence, a second-level controller is needed to transform the heating/cooling demand into a controllable setpoint (e.g., supply water flow rate, supply water temperature) (Huang et al., 2015; Vana et al., 2014). The workarounds weaken the theoretical advantages of MPC.

- Another simplification method is to use a linear function to identify the relationships between the control variable and HVAC supply heating/cooling demand (e.g., using specific heat equation), and minimize the HVAC supply heating/cooling demand during the optimization (Liang et al., 2015; Chen et al., 2016; Ma et al., 2014b,a; Yu et al., 2017; West et al., 2014). However, minimizing HVAC heating/cooling demand is not equivalent to minimizing HVAC energy consumption (electricity/gas consumption) because complex non-monotonic relationships may exist between the two variables.

- Some studies propose more complicated workaround methods to deal with the complex dynamics of HVAC systems, including nonlinear black-box model and heuristic search based MPC (Lindelöf et al., 2015; Chen et al., 2018a), cooperative distributed MPC (Killian and Kozek, 2018), MPC with prioritized objectives (O'Dwyer et al., 2017), and data-driven MPC (Smarra

et al., 2018; Jain et al., 2018). However, these methods add another layer of complexity to the basic MPC, and cannot guarantee a globally optimal solution sometimes.

1.2.2 Whole Building Energy Model (BEM) based Predictive Control

Since MPC can potentially deliver significant energy efficiency improvements for HVAC systems, researchers have been exploring whether BEM can be used in the MPC framework. However, BEM is a high-order simulation program and cannot be used in existing MPC methods directly. Workarounds have to be proposed. For example:

- Zhao et al. (2015) have proposed a real-time EnergyPlus model-based predictive control (EPMPC) method that uses an EnergyPlus model directly for HVAC real-time control. Heuristic-search is used as the optimization solver of the method. However, the prediction horizon of the study is one (i.e., predict for next time step only) because the EnergyPlus-based heuristic-search is too computationally heavy for real-time control. A successive real-life field test of EPMPC (Zhang and Lam, 2017) concludes that, even though EPMPC can be implemented in real-life for real-time control, its scalability is limited due to the over-intensive computation of EnergyPlus.
- Ascione et al. (2016) and May-Ostendorp et al. (2011) have developed non-real-time BEM-based predictive control methods that use heuristic search to pre-calculate the optimized set-point schedules for next day. Since the control methods are non-real-time, more time is available for the BEM-based heuristic search.
- Aftab et al. (2017) and Miezis et al. (2017) have used BEM to calculate the optimized start or stop time of an HVAC system. This is a simpler control problem because it does not require real-time computation. The control method only needs to run once a day every morning.
- Kwak et al. (2015) develops a simple, optimization-free control algorithm based on EnergyPlus, in which EnergyPlus-predicted next-time-step thermal conditions are used as feedback signals for current-time-step control. This control method only shows limited energy savings in a simulation case study.

It can be seen in the existing studies that, building energy models are difficult to be used for real-time model-based control. The major constraints are its high-order nature and relatively intensive computation.

1.2.3 Reinforcement Learning Control

Reinforcement learning (RL) is divided into model-based and model-free approaches. In this study, the term "reinforcement learning" refers to model-free reinforcement learning. "Model-free" means a reinforcement learning algorithm does not use the "transition probability distribution" of an environment. This means an RL agent does not try to learn the environment dynamics. Instead, the RL agent develops a control strategy by "trial-and-error", i.e., it tries different control actions and improves itself by the feedback from the environment.

A Brief History of Reinforcement Learning As a natural learning phenomenon, the concept of "trial-and-error" learning is among the earliest thoughts for artificial intelligence (Sutton and Barto, 2017). Turing (1948) described a design of "trial-and-error" learning, which is conceptually similar to today's reinforcement learning methods:

> "*When a configuration is reached for which the action is undetermined, a random choice for the missing data is made and the appropriate entry is made in the description, tentatively, and is applied. When a pain stimulus occurs all tentative entries are canceled, and when a pleasure stimulus occurs they are all made permanent.*"

To demonstrate "trial-and-error" learning, researchers built electro-mechanical devices in the 1950s and a famous example is Shannon's mouse (AT&T, 1950), which is an electro-mechanical mouse that can find its way to the target in a maze by itself. Digital computer-based studies followed and one of the most influential studies is BOXES (Michie and Chambers, 1968), in which reinforcement learning is used to solve a pole-balancing problem (as shown in Figure 1.3). The pole-balancing problem is a classical reinforcement learning problem with incomplete knowledge, and it influences the later work of temporal difference learning (Sutton and Barto, 2017).

Temporal difference learning has its origin from "secondary reinforcers", which is a concept in animal learning psychology (Sutton and Barto, 2017). In short, a secondary reinforcer is a stimulus that is related to a primary reinforcer. For example, "death" is a primary reinforcer for a goat, and "sounds of predators" are the secondary reinforcer because it means predators are nearby to threaten the goat's life. Thus, the goat can identify the secondary reinforcer and predict the potential threat in the future. Inspired by this concept, Barto et al. (1983) develops the actor-critic method that combines temporal difference learning with trial-and-error learning, where an RL agent is trained to predict a system's future behavior using past observations. Q-learning (Watkins, 1989) is another

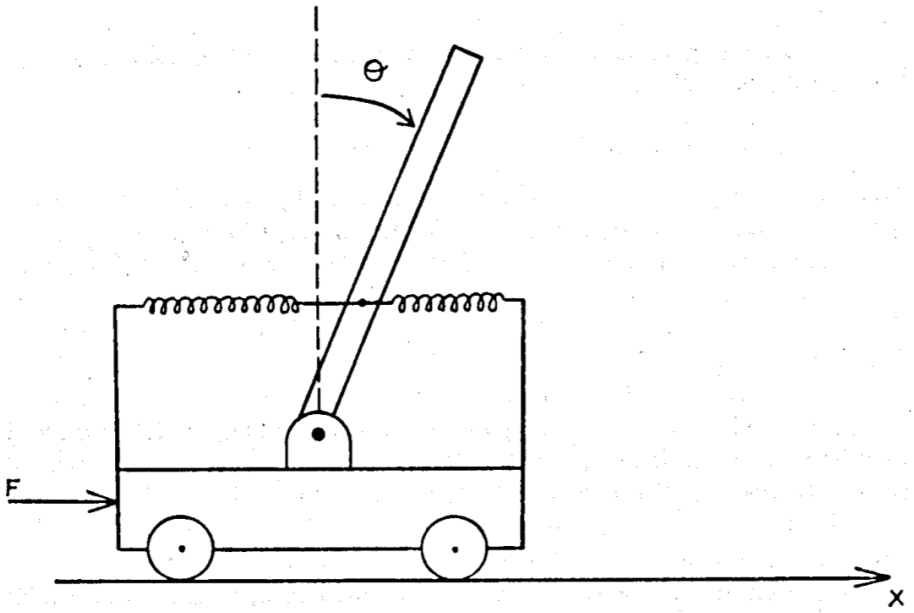

Figure 1.3: Cart-Pole Balancing Problem (Michie and Chambers, 1968)

milestone in the development of reinforcement learning because it bridges the gap between temporal difference learning and optimal control (Sutton and Barto, 2017). Since then, reinforcement learning problems are formulated as a Markov decision process (an optimal control problem firstly studied by Bellman (1957)). Actor-critic and Q-learning are still widely used in today's reinforcement learning.

Deep reinforcement learning (DRL) becomes a popular branch in reinforcement learning since 2013, when Mnih et al. (2013) demonstrates it can match or beat human's performance on some Atari video games by just observing raw game frames. The success of AlphaGo (Silver et al., 2017) in Go games further elevates people's expectations in deep reinforcement learning. The most recent important achievement of deep reinforcement learning is mastering StarCraft II, a real-time multiplayer strategy game (DeepMind, 2019). Deep reinforcement learning is fundamentally the same as classic reinforcement learning methods. The major difference is that DRL uses a deep neural network as the function approximation for an RL agent. The use of a neural network in reinforcement learning is not news. For example, it was studied by Anderson (1986) in the 1980s and used in the famous TD-Gammon program (Tesauro, 1995) (a backgammon playing program). However, only simple neural networks were studied at that time rather than "deep" neural networks (actually TD-Gammon is the "deep" RL of the 1990s). Deep neural networks are now possible thanks to the advancement in computing power. With the help of deep neural networks, deep reinforcement learning can achieve "end-to-end" learning for some complicated tasks. This means that a DRL agent

can use raw sensor data (e.g., raw video frames) as input to make control decisions.

Deep reinforcement learning is still facing several challenges. Firstly, the optimization of a complex neural network is harder than a simple neural network (Goodfellow et al., 2016). A number of methods are proposed in recent literature to make the optimization more effective, such as replay memory (Mnih et al., 2013), double Q-learning (van Hasselt et al., 2015), dueling network (Wang et al., 2015), consistent Bellman operator (Bellemare et al., 2015), optimality tightening (He et al., 2016), proximal policy optimization (Schulman et al., 2017), etc. Secondly, deep reinforcement learning still cannot perform well in an environment with sparse or delayed rewards. Thus, effective exploration has been a popular research topic in the recent years, such as episodic control (Blundell et al., 2016), bootstrapped Q-learning (Osband et al., 2016), count-based exploration (Ostrovski et al., 2017), distributional Q-learning (Bellemare et al., 2017), noisy neural networks (Fortunato et al., 2017), etc. However, the listed methods cannot always achieve better performance in empirical experiments, and extensive hyperparameter tuning is still necessary.

Reinforcement Learning for HVAC Supervisory Control Most HVAC systems have complex dynamics. Compared to model-based control methods, model-free reinforcement learning is easier to implement for HVAC control because it has no restriction for the complexity of a dynamical system. Research in this area dates back to 1990s (e.g., Anderson et al. (1997); Mozer (1998)) and it gained popularity after 2013 (Vázquez-Canteli and Nagy, 2019) when DeepMind successfully applied deep reinforcement learning to play Atari games.

Table 1.1 summarizes some recent studies on reinforcement learning for HVAC supervisory control. The applied environment (or simulator), HVAC system type, climate, experiment design, and RL agent function approximation model of the studies are summarized in the table. There are several research gaps:

1. Most existing studies are based on over-simplified single-zone building simulation models, such as lumped parameter model (LPM) or steady-state model. The simple modeling methods cannot realistically simulate the complicated transient thermal behaviors of a building, and a single-zone building has much simpler thermal behaviors than a multi-zone building. A few studies use multi-zone EnergyPlus models, but the models are simplified with box-shaped geometries and simple thermal zoning. Only one study (Zhang and Lam, 2018) uses a detailed EnergyPlus model as a thermal simulator.

2. A majority of the studies focuses on the control of a heating/cooling unit (e.g., window A/C

unit), which is one of the simplest HVAC systems. Compared to other larger-scale HVAC systems, a heating/cooling unit does not have complex operation characteristics, such as the interrelationship among chillers, pumps, fans, and air dampers.

3. Most of the papers study a single type of HVAC system in a single climate zone. Such studies are incomplete for HVAC-related research since different HVAC systems may have different characteristics in different climates.

4. Several studies have tested only one scenario in the experiment. To validate the robustness of a control method, it is valuable to evaluate it under different possible scenarios, such as different operation schedules, different buildings, etc.

5. The necessity is not justified about the use of "deep" neural networks (NN) as the function approximation of an RL agent. It is interesting to find that, since 2017, almost all the studies have claimed the use of "deep" NN. However, none of them provide evidence of the benefits of a "deep" NN versus a "shallow" NN. In addition, the definition of "deep" is blurred. A two-layer NN is also named "deep" NN in some studies.

Online vs. Offline Learning Online learning means a reinforcement learning agent learns to control when a building is in operation. Theoretically, an online reinforcement learning agent can learn an energy-efficient control policy without any human intervention, and it can adapt to changing building operation characteristics. These advantages are appealing for the practical implementation of a control method. However, as a trial-and-error learning method, a reinforcement learning agent needs sufficient explorations to converge to a stable control policy. Thus, a reinforcement learning agent may take random and wrong control actions when it initially interacts with an HVAC system, and its convergence speed may be slow.

Table 1.2 summarizes the convergence performance of the recent online learning studies which have reported the convergence results. It can be seen that online reinforcement learning requires a significant amount of time to converge even for single-zone buildings with simple HVAC systems. Besides, comfort is compromised during the initial stage of learning. Some recent studies (Ruelens et al., 2015; Dong Li et al., 2015; Costanzo et al., 2016; Nagy et al., 2018) uniformly report half a month for an agent to converge. Even though half a month is a short period compared to the life-cycle of a building, in practice, it may be difficult to convince a building owner if occupants' thermal comfort will be significantly compromised and the learning convergence time is not certain. Moreover, Nagy et al. (2018) shows that online reinforcement learning cannot adapt to the change

Table 1.1: Summary of the Studies on Reinforcement Learning for HVAC Supervisory Control

	Thermal Environment	HVAC System	Climate Zone	Exp Design	RL Agent
Liu and Henze (2006)	SZ LPM	VAV	5A	1 scenario	Tabular
Dalamagkidis et al. (2007)		Heating/ cooling unit	N/K	2 insulation levels	Linear
Yu and Dexter (2010)		VAV	5	Simulator errors	Tabular
Urieli and Stone (2013)		Heating/ cooling unit	N/A	1 scenario	NN
Sun et al. (2013)	MZ LPM	FCU	4	2 building scales	Tabular
Fazenda et al. (2014)	SZ LPM	Heating unit	N/K	1 scenario	
Barrett and Linder (2015)	SZ steady-state	Heating/ cooling unit	4		
Ruelens et al. (2015)	SZ LPM		5	2 insulation levels	Regression (type N/K)
Yang et al. (2015)	System model only	PV/T+ GSHP	5	1 scenario	NN
Dong Li et al. (2015)	MZ Eplus (simplified)	VAV	N/K		Tabular
Peng and Morrison (2016)	SZ LPM	Heating/ cooling unit	2B	Inaccurate simulator	
Costanzo et al. (2016)		Heating unit	N/K	1 scenario	Regression tree
Ruelens et al. (2017)		Heating/ cooling unit	5		
Schmidt et al. (2017)	SZ real-life	Heating (boiler)	4		
Wang et al. (2017)	MZ Eplus (simplified)	VAV	7		"Deep" LSTM
Wei et al. (2017)			3B		"Deep" NN (4 layers)
Chen et al. (2018b)	SZ steady-state	Cooling unit	1A, 3B		N/K
Nagy et al. (2018)	SZ LPM	Heating unit	5	3 operation scenarios	"Deep" NN
Zhang and Lam (2018)	MZ Eplus (detailed)	Radiant heating	5	1 scenario	"Deep" NN (4 layers)
Jia et al. (2019)	MZ Eplus (simplified)	VAV	3C		"Deep" NN
Valladares et al. (2019)	SZ Eplus (simplified)	Cooling unit	2	2 buildings	"Deep" NN (6 layers)
Vázquez-Canteli et al. (2019)	MZ LPM	TES +PV	2A	5 operation scenarios	"Deep" NN (2 layers)
Gao et al. (2019)	SZ TRNSYS (simplified)	Cooling unit	1	1 scenario	"Deep" NN (2 layers)
Zhang (2019)	**MZ Eplus (detailed)**	**VAV, radiant heating, chilled water**	**1,3, 4,5**	**Multiple scenarios**	**Simple to deep NN**

1. Abbreviations: Exp: experiment, SZ: single zone, MZ: multi-zone, LPM: lumped parameter model, Eplus: EnergyPlus, VAV: Variable air volume, FCU: fac coil unit, PV/T: photovoltaic/thermal, GSHP: ground source heat pump, TES: thermal energy storage, NN: neural network, LSTM: long short-term memory neural network.
2. Climate zones are defined in American Society of Heating, Ventilating, and Air Conditioning Engineers (2016).

of indoor air temperature setpoint fast enough, which weakens one of the major advantages of online reinforcement learning. Note that, all of the studies in Table 1.2 are based on simplified computer simulations. Real-life HVAC systems are much more complicated, so the learning convergence speed may be even slower. An unsuccessful recent attempt (Kazmi et al., 2019) to implement online reinforcement learning in real-life admits that the learning time is very long and is of high complexity.

Offline reinforcement learning is another branch for HVAC supervisory control. In this approach, an environment simulator is used to train a reinforcement learning agent offline, and the pre-trained agent is deployed in an actual HVAC system. An example of this approach is shown in Figure 1.4. In this example, a calibrated EnergyPlus model is used as a simulator to train an RL agent offline, and the trained RL agent is deployed in an actual system as a static control policy. Offline reinforcement learning avoids "surprises" because an RL agent can be thoroughly tuned and evaluated in a simulator before deployment. This approach has been successfully validated in the real-life experiments including Liu and Henze (2006); Zhang and Lam (2018); Valladares et al. (2019).

Table 1.2: Summary of the Convergence Performance of Online Reinforcement Learning for HVAC Supervisory Control

	Learning Scenario	**Convergence Speed**	**Other Comments**
Dalamagkidis et al. (2007)	Optimize comfort and energy in a single-zone building with a heating/cooling unit	>36 months	PMV is out of the acceptable bounds during learning
Yu and Dexter (2010)	Optimize comfort and energy by learning the parameters in a fuzzy controller	>12 months	Comfort is worse than the baseline during learning
Urieli and Stone (2013)	Optimize comfort and energy in a house with a heating unit	3-day random exploration	Detailed convergence history not reported
Fazenda et al. (2014)	Optimize comfort and energy in a house with a heating unit	>1 month	Detailed comfort performance not reported
Ruelens et al. (2015)	Optimize comfort and energy in a single-zone building with a heating/cooling unit	>0.5 month	Significant IAT violation during learning
Dong Li et al. (2015)	Optimize comfort and energy in a multi-zone building with VAV cooling	>0.5 month	Detailed comfort performance not reported
Costanzo et al. (2016)	Optimize comfort and energy in a single-zone building with a heating unit	>0.5 month	Detailed comfort performance not reported
Chen et al. (2018b)	Optimize comfort and energy in a single-zone building with a cooling unit	"sufficiently long"	Detailed convergence history not reported
Nagy et al. (2018)	Optimize comfort and energy in a single-zone building with a heating unit	>0.5 month	Significant IAT violation during learning; RL cannot adapt to IAT setpoint change fast enough

Abbreviations: PMV: predicted mean vote (a calculated thermal comfort metric), IAT: indoor air temperature, VAV: Variable air volume system (a common HVAC system in the U.S.).

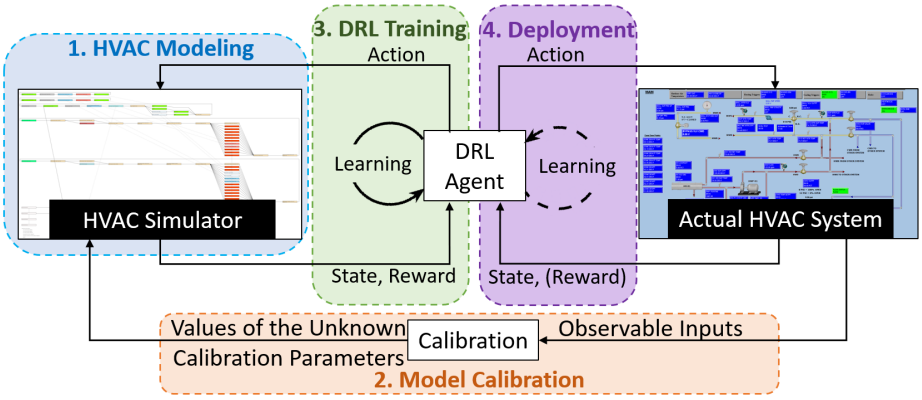

Figure 1.4: A Framework of Offline Reinforcement Learning for HVAC Supervisory Control (Zhang and Lam, 2018)

1.2.4 Summary of Literature Review

Model predictive control (MPC) is the most popular model-based control methods for HVAC systems. However, it has strict requirements for the order of system dynamics and cost functions. This means whole building energy models (BEM) cannot be directly used for MPC because BEMs are high-order physics-based simulation programs. Workarounds have to be proposed to integrate BEMs into the MPC framework, which makes these methods difficult to implement and scale-up.

Reinforcement learning has no restrictions for the order of system dynamics. Online reinforcement learning is theoretically possible for HVAC control, but it may take too long to converge and may significantly compromise thermal comfort in the initial stage of learning. Practically, an HVAC simulator is used to pre-train a reinforcement learning agent offline. BEM is an ideal candidate for the simulator since it can simulate HVAC behaviors in detail.

Research gaps are existed in the field of reinforcement learning for HVAC supervisory control, including:

- Most existing studies are based on over-simplified building models and HVAC systems.
- Most proposed control methods are evaluated in a single type of HVAC system under a single climate and a single experimental scenario.
- The use “deep” reinforcement learning is not justified.

This study (Zhang, 2019) will fill the gaps, as shown in Table 1.1.

1.3 Research Scope and Objectives

Whole building energy model (BEM) has been widely used for new building design but is seldom used to develop energy-efficient HVAC supervisory control strategies. As a detailed physics-based energy simulation model, BEM-based (or BEM-assisted) control can potentially reduce HVAC energy consumption and reduce the life-cycle cost of BEM development. However, BEM cannot be directly used in classical model-based control methods, such as model predictive control, because of its high-order nature. Hence, this study uses reinforcement learning to "learn" energy-efficient control strategies via BEM. The general objective of this study is to *develop a reinforcement learning-based method to use BEMs to develop energy-efficient supervisory control strategies for complicated commercial HVAC systems.*

The assumed application scenario of the proposed method is:

> *During the design phase of a new commercial building, control engineers use the proposed method to develop an energy-efficient strategy for HVAC supervisory control using the detailed whole building energy model created by thermal modelers.*

Based on the general objective and assumed application scenario, this study has the following sub-tasks:

1. Develop a general framework and its related software programs to use BEM and reinforcement learning to develop energy-efficient HVAC supervisory control strategies;
2. Systematically evaluate the proposed framework through computer simulations, including:
   - Evaluate the proposed framework in detailed whole building energy models.
   - Evaluate the proposed framework for four typical commercial HVAC systems, including a variable-air-volume (VAV) system for cooling, a VAV system for heating, a radiant heating system, and a multi-chiller chilled water system.
   - Evaluate the proposed framework for four different climate zones, including climate zone 5 (Pittsburgh, PA), 4 (Beijing, China), 3 (Shanghai, China) and 1 (Singapore)[1].
   - Evaluate the proposed framework for two different levels of building thermal mass.

[1]The climate zones are defined in American Society of Heating, Ventilating, and Air Conditioning Engineers (2016)

- Evaluate the versatility of the RL-trained control strategies under different variations in HVAC operational conditions, including weather conditions, internal load schedules, and indoor air temperature setpoints.
- Evaluate the convergence and control performance of different function approximation models in reinforcement learning, from simple to complex neural networks.

3. Implement the control framework in an actual radiant heating system to demonstrate its practical feasibility and effectiveness of the control framework.
4. Provide a usage guideline for the proposed control framework.

This study has the following hypotheses:

1. The proposed reinforcement learning-based control framework can directly use a whole building model to develop an HVAC supervisory control strategy to achieve reduced HVAC energy consumption and reduced operational constraint violations than rule-based control strategies.

2. Simple neural network models of reinforcement learning are easier to train (i.e., they converge faster and are more robust to different learning rates), but complex neural network models can achieve better control performance (i.e., they achieve less energy consumption and less operational constraint violations).

1.4 Summary of the Chapters

There are ten chapters in this thesis:

Chapter 1, Introduction: Provide the necessary background and motivation of this study; Review relevant literature on model predictive control, whole building energy model-based predictive control and reinforcement learning control for HVAC systems; identify research gaps and determine the scope and objectives of this study.

Chapter 2, Control Framework Development: Present the control framework based on Asynchronous Advantage Actor-critic method, including step-by-step workflow, theoretical backgrounds, and the architecture of related software programs.

Chapter 3, Experimental Design for the Control Framework Evaluation: Present the general objectives of the experimental design; describe experiment scenarios and a general experimental procedure.

Chapter 4, Experiment 1: Variable-air-volume (VAV) System for Cooling: Present the detailed experimental design of a VAV system for cooling, including the configurations of the VAV system, simulation assumptions, state/action/reward design for reinforcement learning, evaluation simulators, the choice of the hyperparameters of RL and the baseline control strategy; analyze the results of both convergence performance and control performance.

Chapter 5, Experiment 2: Variable-air-volume (VAV) System for Heating: Present the detailed experimental design of a VAV system for heating (which is similar to Experimental 1); analyze the results of both convergence performance and control performance.

Chapter 6, Experiment 3: Radiant Heating System: Present the detailed experimental design of a slow-response radiant heating system, including the system configurations, simulation assumptions, state/action/reward design for reinforcement learning, approaches to solve reward design challenges, evaluation simulators, the choice of the hyperparameters of RL and the baseline control strategy; analyze the results of both convergence performance and control performance.

Chapter 7, Experiment 4: Multi-chiller Chilled Water System: Present the detailed experimental design of a multi-chiller chilled water system, including the system configurations, simulation assumptions, practical operation requirements, state/action/reward design for reinforcement learning, evaluation simulators, the choice of the hyperparameters of RL and the baseline control strategy;

analyze the results of both convergence performance and control performance.

Chapter 8, A Real-life Deployment Case Study: Present an implementation and deployment case study of the control framework in an actual radiant heating system, including EnergyPlus modeling, model calibration, offline RL training, and deployment; analyze the energy efficiency performance of the deployment via a data-driven normalized energy saving analysis method.

Chapter 9, Usage Guidelines for the Control Framework: List the usage guidelines of the control framework based on the simulation experiments.

Chapter 10, Conclusion and Future Work: Summarize the major findings of this study; summarize the limitations of this study; list the future work.

Chapter 2

Control Framework Development

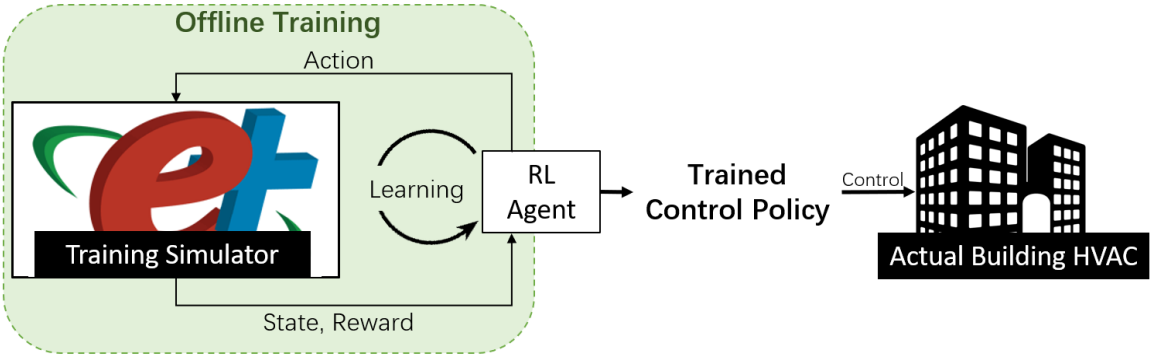

Figure 2.1: Use a Whole Building Energy Model (EnergyPlus) to Develop HVAC Supervisory Control Strategies via Reinforcement Learning

2.1 Overview

The control framework aims to use a whole building energy model to develop energy-efficient strategies for HVAC supervisory control via reinforcement learning (RL). Figure 2.1 schematically shows the control framework, where a whole building energy model is used as a simulator to train a reinforcement learning agent offline, and the outcome of the offline training is a supervisory control policy. The RL-trained control policy can be used to control an actual building HVAC system. EnergyPlus is used as the whole building modeling engine throughout this study because it is one of the most popular and sophisticated programs in the industry.

Three components are needed to realize this control framework, including:

1. A reinforcement learning algorithm;
2. Definition of the state/action/reward;
3. Connection between an EnergyPlus simulator and an RL agent.

The following sections will describe the three components in detail.

2.2 Key Terminologies

Four key terminologies are defined as follows:

- Control time step: A time interval when an RL agent observes the state and the reward, executes a control action and waits for the resulting state and reward of the next time step.
- Simulation time step (e.g., EnergyPlus): The time step defined in EnergyPlus simulators. It is independent of the control time step. For example, if the control time step is 15-min and the simulation time step is 5-min, this means an RL agent interacts with the simulator every three simulation time steps.
- Simulation episode: The simulation time period of a simulator, e.g., Jan 1st-Mar 31st. During the RL training, an EnergyPlus simulation will be repeated for multiple episodes.
- Interaction steps: The number of times that an RL agent (including its local RL agent "workers" if using an asynchronous method, will be explained later) interacts with its environment (one interaction means an RL agent finishes one control time step).
- Control policy: the control strategy given by an RL agent after the RL training. It is a function that takes the state as its input and outputs a control action.

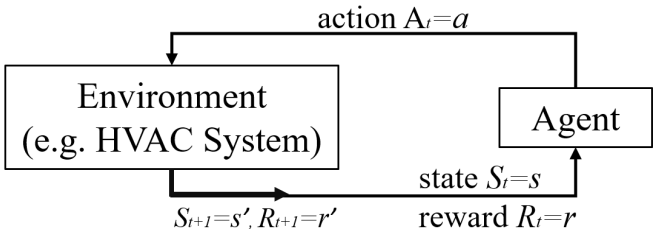

Figure 2.2: A Standard Reinforcement Learning Problem

2.3 Reinforcement Learning Algorithm

2.3.1 Standard Reinforcement Learning Problem

A standard reinforcement learning problem, as shown in Figure 2.2, is formulated as a Markov decision process (MDP) where, at a control time step t, an agent observes the state S_t and reward R_t to provide the control action A_t (Sutton and Barto, 2017). In this formulation:

- *State* represents an RL agent's observations, typical observations include indoor air temperature, outdoor air temperature, HVAC energy consumption, etc.;
- *action* represents the control actions that an RL agent takes for its environment, such as "turn on heating";
- *reward* is a numeric value representing the degree of goodness of a state/action pair, for example, [comfortable indoor environment, turn off heating] may lead to a high reward value.

By formulating the control problem as a MDP, it is assumed state transitions meet the Markov property, i.e.,

$$P(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t) = P(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t, ..., S_0 = s_0, A_0 = a_0), \quad (2.1)$$

which means the next state is only dependent on the last state and last action, and is conditionally independent of all the previous states and actions. However, empirically, some problems can be well solved by reinforcement learning even though the input state is not a full information state (a.k.a., Markov property), such as the Atari games (Mnih et al., 2013).

The goal of reinforcement learning is to learn a control policy $\pi : S_t \rightarrow A_t$ that maximizes the cumulative reward $\sum_t^{T_\infty} R_t$ at each control time step. The control policy can be a stochastic one,

that is:

$$\pi(s, a) = P(a|s), \tag{2.2}$$

which means the probability of taking the action a given the state s.

2.3.2 Value Functions and Function Approximation

There are two closely-related value functions to describe the degree of goodness of a control policy π, including the state-value function $v_\pi(s)$ and the action-value function $q_\pi(s, a)$, as shown below (Sutton and Barto, 2017):

$$v_\pi(s) \equiv \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\Big] \tag{2.3a}$$

$$\equiv \mathbb{E}[R_{t+1} + \gamma v_\pi(s') | S_t = s, S_{t+1} = s'], \tag{2.3b}$$

$$q_\pi(s, a) \equiv \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\Big] \tag{2.4a}$$

$$\equiv \mathbb{E}[R_{t+1} + \gamma v_\pi(s') | S_t = s, A_t = a, S_{t+1} = s'], \tag{2.4b}$$

where γ is the reward discount factor. The state-value function $v_\pi(s)$ is the expected cumulative discounted reward of a control policy π given the state $S_t = s$, which intuitively represents "how good is the state". The action-value function $q_\pi(s, a)$ is similar to the state-value function, but it is conditioned on a $\{S_t = s,\ A_t = a\}$ tuple. Intuitively, this function represents "how good is the action".

Classical reinforcement learning uses a table to store the state-values and action-values for each state and state/action pair (tabular reinforcement learning). The tabular method is simple and fast to compute, but the table becomes excessively large for complicated continuous-state problems. Hence, parameterized functions are used to represent the state-value function, the action-value function and the control policy (Sutton and Barto, 2017), i.e.,

$$v_\pi(s) \approx v(s; \boldsymbol{\theta_v}), \tag{2.5}$$

$$q_\pi(s, a) \approx q(s, a; \boldsymbol{\theta_q}), \tag{2.6}$$

$$\pi(s, a) \approx \pi(s, a; \boldsymbol{\theta}), \tag{2.7}$$

where $\boldsymbol{\theta_v}$, $\boldsymbol{\theta_q}$, and $\boldsymbol{\theta}$ are weight vectors in the parameterized functions. Any parameterized regression models can be used as the parameterized functions, such as linear regression, multi-layer perceptron, and deep neural networks.

2.3.3 Policy Gradient Method

In general, mainstream reinforcement learning methods are divided into value-based methods and policy gradient methods. Value-based methods, such as Q-learning, learn the value functions to derive a control policy. For example, a greedy control policy always selects the action with the highest action-value. Another branch is policy gradient. Policy gradient methods directly learn a parameterized control policy, rather than derive a control policy from the value functions. Thus, policy gradient methods have better convergence property and can develop a stochastic control policy (Sutton and Barto, 2017).

A policy gradient method aims to learn the parameter $\boldsymbol{\theta}$ in the parameterized stochastic control policy:

$$P(a|s) = \pi(s,a) \approx \pi(s,a;\boldsymbol{\theta}), \tag{2.8}$$

that maximizes the average reward per control time step. The problem can be formulated as:

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_s d_\pi(s) \sum_a R_s^a \pi(s,a;\boldsymbol{\theta}), \tag{2.9}$$

where $d_\pi(s)$ is the stationary distribution for the state s of the Markov chain starting from the initial state following a policy π, and R_s^a is the environment reward at a state s taking an action a (Sutton and Barto, 2017).

Equation (2.9) is an unconstrained optimization problem which can be solved by gradient descent (GD), i.e.,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}), \tag{2.10}$$

where α is the learning rate. The key part of gradient descent is to calculate $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$, which is the

gradient of $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Sutton and Barto (2017) derives it as:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_s d_\pi(s) \sum_a R_s^a \pi(s,a;\boldsymbol{\theta}) \frac{\nabla_{\boldsymbol{\theta}} \pi(s,a;\boldsymbol{\theta})}{\pi(s,a;\boldsymbol{\theta})} \tag{2.11a}$$

$$= \sum_s d_\pi(s) \sum_a R_s^a \pi(s,a;\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log \pi(s,a;\boldsymbol{\theta}) \tag{2.11b}$$

$$= \mathbb{E}_\pi \left[\nabla_{\boldsymbol{\theta}} \log \pi(s,a;\boldsymbol{\theta}) q(s,a;\boldsymbol{\theta_q}) \right] \tag{2.11c}$$

$$= \mathbb{E}_\pi \left[\nabla_{\boldsymbol{\theta}} \log \pi\left(s,a;\boldsymbol{\theta}\right) \left(q\left(s,a;\boldsymbol{\theta_q}\right) - v\left(s;\boldsymbol{\theta_v}\right) \right) \right]. \tag{2.11d}$$

In the above derivation, Equation (2.11a) and Equation (2.11b) follow the properties of derivatives of logarithmic functions, Equation (2.11c) is obtained based on policy gradient theorem (Sutton and Barto, 2017). Equation (2.11c) is then subtracted by a zero-valued function $\mathbb{E}_\pi[\nabla_{\boldsymbol{\theta}} \log \pi(s,a;\boldsymbol{\theta}) v(s;\boldsymbol{\theta_v})]$ because it reduces the variance of $q(s,a;\boldsymbol{\theta_q})$ for better learning stability (Sutton and Barto, 2017). Equation (2.11c) is called actor-critic method, since there is a "critic" ($q(s,a;\boldsymbol{\theta_q})$) to evaluate the "actor" ($\pi(s,a;\boldsymbol{\theta})$). Equation (2.11d) is called advantage actor-critic because the advantage-value function ($q(s,a;\boldsymbol{\theta_q}) - v(s;\boldsymbol{\theta_v})$) is used as the "critic" instead of the action-value function.

2.3.4 Gradient Descent Formulation

After the gradient is known, Equation (2.10) becomes

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \mathbb{E}_\pi \big[\nabla_{\boldsymbol{\theta}} \log \pi(s,a;\boldsymbol{\theta}) \big(q(s,a;\boldsymbol{\theta_q}) - v(s;\boldsymbol{\theta_v}) \big) \big] \tag{2.12a}$$

$$= \boldsymbol{\theta} + \alpha \mathbb{E}_\pi [\nabla_{\boldsymbol{\theta}} \log \pi(s,a;\boldsymbol{\theta}) \big(R_b(s) - v(s;\boldsymbol{\theta_v}) \big) \big], \tag{2.12b}$$

where,

$$\mathbb{E}_\pi[R_b(s)] = \sum_{i=1}^{n} \gamma^{n-1} r'_{i,\pi} + \gamma^n v(s'_{n,\pi};\boldsymbol{\theta_v}) \tag{2.13}$$

in which $r'_{n,\pi}$ and $s'_{n,\pi}$ are the reward and state at n-control-time-step after the current state s following the policy π. Usually, n is a small number. Equation (2.13) follows from the definition of Equation (2.4).

Equation (2.12b) has a new unknown variable $\boldsymbol{\theta_v}$, which is the weight in the parameterized state-value function. This parameter can also be learned using gradient descent to minimize the mean

squared error between the "true" and the "predicted" state-value, that is,

$$\boldsymbol{\theta_v} \leftarrow \boldsymbol{\theta_v} - \alpha \mathbb{E}_\pi \big[\nabla_{\boldsymbol{\theta_v}} \big(v_{true} - v(s; \boldsymbol{\theta_v}) \big)^2 \big] \tag{2.14a}$$

$$\approx \boldsymbol{\theta_v} - \alpha \mathbb{E}_\pi \big[\nabla_{\boldsymbol{\theta_v}} \big(R_b(s) - v(s; \boldsymbol{\theta_v}) \big)^2 \big], \tag{2.14b}$$

where $\mathbb{E}_\pi[R_b(s)]$ is used as a biased estimate of v_{true}. This is called temporal-difference learning (Sutton and Barto, 2017), which uses a bootstrapped state-value function $\mathbb{E}_\pi[R_b(s)]$ as a biased estimation of the true state-value function. As n in $\mathbb{E}_\pi[R_b(s)]$ increases, the above bootstrapped function infinitely approaches the true value. If $n = \infty$, it is called Monte-Carlo reinforcement learning. While Monte-Carlo reinforcement learning has better convergence property, temporal difference learning is computationally more efficient.

To reduce the complexity of the problem, the parameterized functions $v(;\boldsymbol{\theta_v})$ and $\pi(;\boldsymbol{\theta})$ can be partially combined with parameter sharing in a single neural network (Mnih et al., 2016), as shown in Figure 2.3. We then have the state-value function $v(s;\boldsymbol{\theta_v}) = v(s;\boldsymbol{\theta_{\pi,v}})$ and the policy distribution $\pi(s,a;\boldsymbol{\theta}) = \pi(s,a;\boldsymbol{\theta_{\pi,v}})$[1]. As a result, the one-step learning update of the gradient descent of $\boldsymbol{\theta_{\pi,v}}$ becomes:

$$\boldsymbol{\theta_{\pi,v}} \leftarrow \boldsymbol{\theta_{\pi,v}} - \alpha \mathbb{E}_\pi \big[\beta \nabla_{\boldsymbol{\theta_{\pi,v}}} \big(R_b(s) - v(s;\boldsymbol{\theta_{\pi,v}}) \big)^2 \tag{2.15a}$$

$$- \nabla_{\boldsymbol{\theta_{\pi,v}}} A_d \log \pi(s,a;\boldsymbol{\theta_{\pi,v}}) \big], \tag{2.15b}$$

where β is the value loss weight and A_d is a constant representing the advantage-value function (not used in the gradient calculation):

$$A_d = R_b(s) - v(s;\boldsymbol{\theta_{\pi,v}}). \tag{2.16}$$

2.3.5 Exploration Methods

A reinforcement learning agent needs to explore different control policies to converge to a near-optimal one. Random exploration (i.e., take random control actions) is the most commonly used one, such as ϵ-greedy and entropy regularization. However, random exploration is inefficient and requires extensive hyperparameter tuning. This study adopts two structured and easy-to-implement methods for the exploration, including NoisyNet (Fortunato et al., 2017) and Asynchronous method

[1] The notations here are slightly abused since the state-value function and the policy distribution do not share all the function parameters.

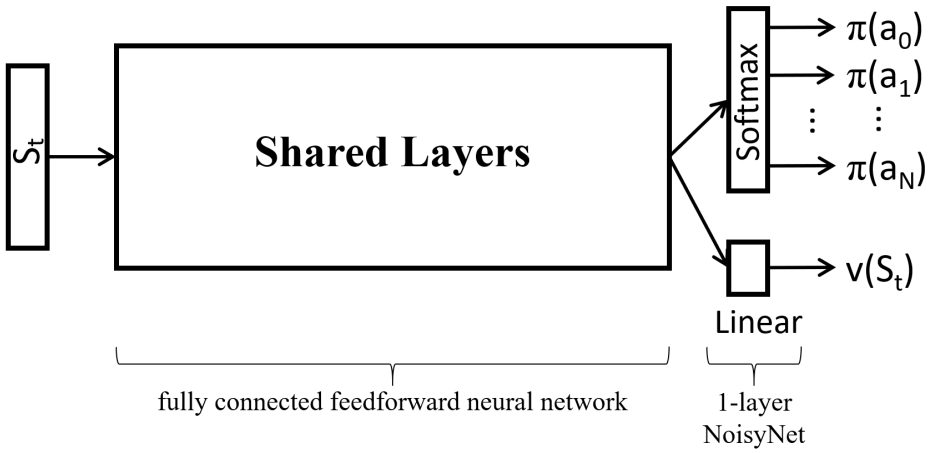

Figure 2.3: Policy and State-value Function Architecture

(Mnih et al., 2016).

NoisyNet

A layer in a conventional feed-forward fully-connected neural network is written as (without activation):

$$\boldsymbol{h} = \boldsymbol{\omega}\boldsymbol{x} + \boldsymbol{b} \tag{2.17}$$

where $\boldsymbol{h}$ is the layer output, $\boldsymbol{\omega}$ is the weight matrix, $\boldsymbol{x}$ is the layer input, and $\boldsymbol{b}$ is the bias vector. During reinforcement learning, $\boldsymbol{\omega}$ and $\boldsymbol{b}$ are learned through gradient descent optimization.

A layer in NoisyNet (Fortunato et al., 2017) is written as (without activation):

$$\boldsymbol{h} \equiv (\boldsymbol{\mu_\omega} + \boldsymbol{\sigma_\omega}\boldsymbol{\epsilon_\omega})\boldsymbol{x} + \boldsymbol{\mu_b} + \boldsymbol{\sigma_b}\boldsymbol{\epsilon_b}, \tag{2.18}$$

where $\boldsymbol{\mu_\omega}$, $\boldsymbol{\sigma_\omega}$, $\boldsymbol{\mu_b}$, $\boldsymbol{\sigma_b}$ are learnable variables, and $\boldsymbol{\epsilon_b}$ and $\boldsymbol{\epsilon_\omega}$ are noise random variables. Compared to the conventional layer in Equation (2.17), NoisyNet introduces noises in the weight matrix and bias vector. During reinforcement learning, the learnable variables are also learned through gradient descent optimization, but the noise random variables are changing for each learning step.

Unlike dithering approaches (e.g., ϵ-greedy) that add independent randomness to a control policy, NoisyNet can "induce a consistent, and potentially very complex, state-dependent change in policy over multiple time steps" (Fortunato et al., 2017). Intuitively, this is because stochasticity is

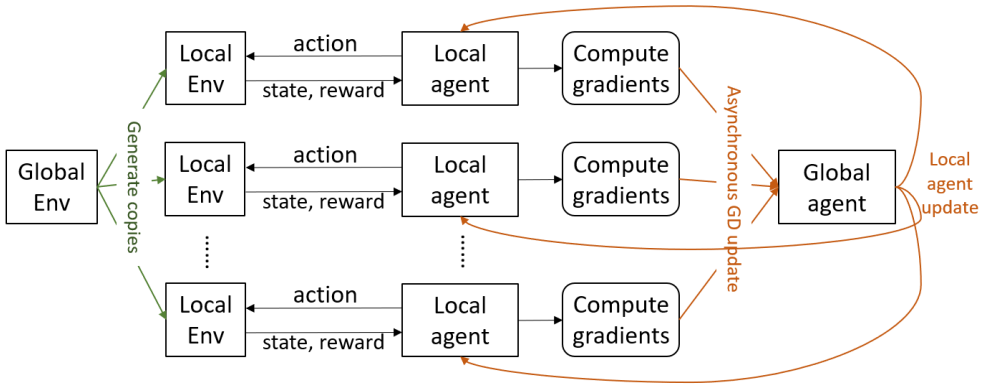

Figure 2.4: Schematic Diagram of Asynchronous Reinforcement Learning

embedded in the reinforcement learning process so an RL agent has more structured explorations. In addition, the level of noises does not require tuning since it will be learned in the learning process (via $\boldsymbol{\sigma_\omega}$ and $\boldsymbol{\sigma_b}$ in Equation (2.18)). NoisyNet shows obvious improvements in a wide range of Atari games (Fortunato et al., 2017).

In this study, NoisyNet is used as the last layer of the function approximation model, as shown in Figure 2.3. During reinforcement learning training, $\boldsymbol{\epsilon_b}$ and $\boldsymbol{\epsilon_\omega}$ are sampled from a Gaussian distribution with 0 mean and 1 standard deviation. After reinforcement learning training, $\boldsymbol{\epsilon_b}$ and $\boldsymbol{\epsilon_\omega}$ are set to zero for inference.

Asynchronous Method

Conventional reinforcement learning has only one agent to interact with the environment. When the environment has complex dynamics, it may take a long time for a single agent to explore the whole state space. Mnih et al. (2016) develops an asynchronous reinforcement learning method which fires many local RL agents in parallel to explore the environment. Since each local RL agent has a stochastic control policy, they will explore different regions of the state space. At a certain frequency, the local RL agents asynchronously perform gradient descent update for the global RL agent, and the global RL agent also updates the local RL agents. The principle of this method is schematically shown in Figure 2.4.

In addition to the advantage of efficient exploration, the asynchronous method is computationally efficient. Thus, it can be performed using only CPUs.

2.3.6 Summary of the Reinforcement Learning Algorithm

This study formulates the reinforcement learning problem into a Markov decision process, where an RL agent observes the current state, take an action, and receives the next state observation and reward. The goal of reinforcement learning is to develop a control policy $\pi : S_t \rightarrow A_t$ that maximizes the cumulative reward.

Policy gradient is used to directly optimize a parameterized control policy. More specifically, advantage actor-critic (A2C) is the algorithm for the policy gradient reinforcement learning. In addition to the optimization of the control policy, A2C needs to solve an unknown state-value function at the same time. To make the problem simpler, this study uses a shared neural network architecture to approximate both control policy and state-value function. Gradient descent is used for optimization.

Efficient exploration is important for efficient reinforcement learning. Rather than using random exploration, this study adopts NoisyNet and the asynchronous method. The asynchronous version of A2C is called A3C.

2.4 Definition of State, Action and Reward

As presented in section 2.3.1, a standard reinforcement learning problem should define the state, action, and reward.

2.4.1 State Design

The state represents an RL agent's observation for the environment. In addition, the state should be designed to make the whole process obey the Markov property, i.e., the state transition depends only on the current state and the current control action, and is conditionally independent of the previous states. In other words, an RL agent can sufficiently make a control decision by just observing the current state.

Many building HVAC systems have delayed responses, e.g., indoor air temperature may remain high even after a heater is turned off, or supply water temperature may remain low even after a boiler is turned on. The delayed responses are firstly caused by the thermal mass of building structures, and are also caused by the insufficient capacity or non-ideal operations of HVAC components. Thus, observations at one single control time step are not sufficient for an RL agent to make control decisions.

This study designs the state as a stack of the observations at the current and past control time steps, as shown below:

$$S_t = \left\{ob_t, ob_{t-1}, ..., ob_{t-n}\right\}, \tag{2.19}$$

where ob_t is the environment observation at the control time step t, n is the length of the history to be considered. All items in ob are normalized to 0-1 for the optimization purpose of neural networks. Min-max normalization is used, as shown below:

$$ob_{norm} = \frac{ob - ob_{min}}{ob_{max} - ob_{min}}, \tag{2.20}$$

where ob_{min} and ob_{max} are determined based on the physical limitations or the operational ranges of an item.

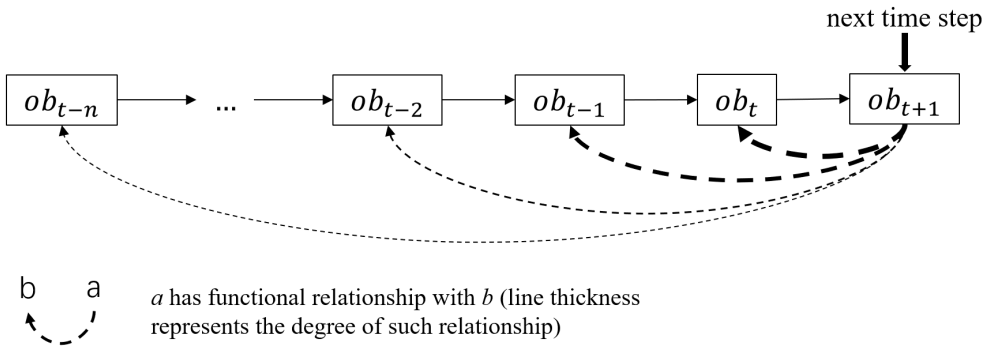

Figure 2.5: Relationship between the Observation at the Next Time Step and the Historical Time Steps

Determine the Length of the History Ideally, the length of the history (n in Equation (2.19)) should be determined based on the Markov property testing, i.e., the value of n should make the process obey the Markov property. However, Markov property testing is highly complicated. This study proposes a simpler alternative approach to determine the length of history to be considered in the state.

In HVAC systems, the environment observation at the time step $t+1$ usually depends more on recent historical observations, and less on older historical observations. For example, the indoor air temperature of the next time step is closely related to the current indoor air temperature, and may not be related to the indoor air temperature of 3 hours ago. This is schematically illustrated in Figure 2.5. Hence, it is only necessary to include the historical observations that have a strong relationship with the next-time-step observation. The relationship (dependence) can be measured by distance correlation (Székely et al., 2008). Distance correlation can measure both linear and nonlinear associations between two variables, and it ranges from 0 to 1 where 0 means no association and 1 means linear association between two variables.

Since HVAC simulators are available, they can be used to generate necessary data to determine the length of the history in the state. The full procedure is as follows:

1. Run the simulation of an HVAC system for one episode using a random control policy.

2. Record the observations of all control time steps from the simulation , i.e., $\mathbf{ob_{all}} \in \mathbb{R}^{p \times q}$ where p is the number of the observations (i.e., the total number of control time steps in one simulation episode) and q is the number of items in each observation; The first row in $\mathbf{ob_{all}}$ is the observation of the first control time step in a simulation episode.

3. Test the dependence between the next-control-time-step observations ($\mathbf{ob_{t+1}}$) and the observations at n control time steps before the current control time step ($\mathbf{ob_{t-n}}$). $\mathbf{ob_{t+1}}$ and $\mathbf{ob_{t-n}}$ are $\mathbb{R}^{(p-1-n)\times q}$ matrices where each row in $\mathbf{ob_{t+1}}$ is the observation at the control time step $t+1$ and each row in $\mathbf{ob_{t-n}}$ is the observation at the control time step $t-n$, as shown below:

$$\mathbf{ob_{t+1}} = [ob_{t=n+2}, ob_{t=n+3}, ..., ob_{t=p}]^T, \tag{2.21}$$

$$\mathbf{ob_{t-n}} = [ob_{t=1}, ob_{t=2}, ..., ob_{t=p-1-n}]^T, \tag{2.22}$$

The distance correlation between the observations at $t+1$ and the observations at $t-n$ ($dcor_n$) is:

$$dcor_n = dcor(\mathbf{ob_{t+1}}, \mathbf{ob_{t-n}}), \tag{2.23}$$

where $dcor()$ is the function to calculate the distance correlation.

$dcor_n$ is calculated for all choices of n from $n = 1$ upwards (i.e., $n = 1, 2, 3, ...$), until $dcor_n < dcorThres$ whereby the dependence is not significant anymore. Then the last choice of n that makes $dcor_n \geq dcorThres$ is the length of the history in the state. $dcorThres$ is a tunable hyperparameter. Since $dcor_n$ ranges from 0 to 1, this thesis simply uses $dcroThres = 0.5$ to save the computational cost related to the hyperparameter tuning.

The above procedures are summarized in Algorithm 1.

Algorithm 1 Determine the Length of the History in the State

```
procedure DETERMINEN(ob_all, dcorThres)
    ▷ ob_all is all the observations in one simulation episode, which has p rows and q cols
    ▷ dcorThres is the threshold for the distance correlation
    dcorThis = 1                          ▷ init with the maximum distance correlation value
    n = 0
    while dcorThis ≥ dcorThres do
        n = n + 1
        obNext = ob_all[n+2:p,:]          ▷ slice ob_all from the row n+2 to the row p (inclusive)
        obHist = ob_all[1:p-1-n,:]        ▷ slice ob_all from the first row to the row p-1-n (inclusive)
        dcorThis = dcor(obNext, obHist)
    return n - 1                          ▷ return the last choice of n
```

2.4.2 Action Space Design

The action space in this study is a discrete set of control choices, as shown below,

$$A_t = \{a_1, a_2, ..., a_n\}, \tag{2.24}$$

where a_n is a control action choice, such as "turn on chiller1 and chiller2".

2.4.3 Reward Function Design

The reward is a function of the state at the last time step, the state at the current time state and the action at the last time step, i.e.,

$$R_{t+1} = [f_{reward}(S_t, S_{t+1}, A_t)]_0^1. \tag{2.25}$$

The reward is a scalar value in the range of $[0, 1]$ representing how good are the states and/or the action. The reward function ($f_{reward}()$) is user-defined and can be dramatically different for different scenarios. The following chapters will show some examples of the reward function design. In general, this function gives a small reward value when HVAC energy consumption is high and/or HVAC operational constraints are not met (e.g., thermal comfort constraints, equipment safety constraints), and gives a large value when the energy consumption is low and the operational constraints are met.

2.5 EnergyPlus Simulator for Reinforcement Learning (EPRL)

EnergyPlus (The U.S. Department of Energy, 2019a) is a dynamic simulation program for building energy performance. In this study, an EnergyPlus model is viewed as the virtual environment for an HVAC system. As shown in section 2.3.1, a reinforcement learning agent needs to interact with the environment by observing the state and reward, and taking control actions. This study develops a Python-based program to realize such interactions.

The program is named EnergyPlus Simulator for Reinforcement Learning (EPRL), which wraps an EnergyPlus model in the Python-based OpenAI Gym interface (Brockman et al., 2016). OpenAI Gym is a prevailing programming interface for reinforcement learning. EPRL is based on the existing inter-program communication function of EnergyPlus (ExternalInterface in EnergyPlus) and BCVTB middleware (Lawrence Berkeley National Laboratory, 2016). The architecture of EPRL is shown in Figure 2.6. The main component of this architecture is the OpenAI gym interface, which is a Python object with three functions: object constructor (*___init___*), *reset()* and *step(a)*. The operation sequence of EPRL is as follows:

1. When an OpenAI gym interface object (gym object) is initiated (by calling the object constructor), a server socket for inter-program communications is created.

2. The function *reset()* is called once by an RL agent when the learning first starts. When it is called, an EnergyPlus instance is first created using the EnergyPlus input definition file (IDF) and data exchange file (with the .cfg extension) stored in the local drive.

   The data exchange file specifies the types of control actions related to the input to and the output from the EnergyPlus simulation.

3. The gym object then creates a TCP connection with the EnergyPlus instance, in which ExternalInterface of EnergyPlus performs as a client through BCVTB.

4. The gym object reads the initial output from the EnergyPlus simulation using the TCP connection, and returns the output to the RL agent. The RL agent is responsible to process the output to extract the state and reward for reinforcement learning.

5. The function *step(a)* is called once at each control time step. When it is called, the gym object uses the TCP connection to send the action $\mathbf{a}$ to the EnergyPlus simulation and read the resulting simulation output. The gym object then returns the output to the RL agent. The output should be processed by the RL agent to extract the state and reward for the learning.

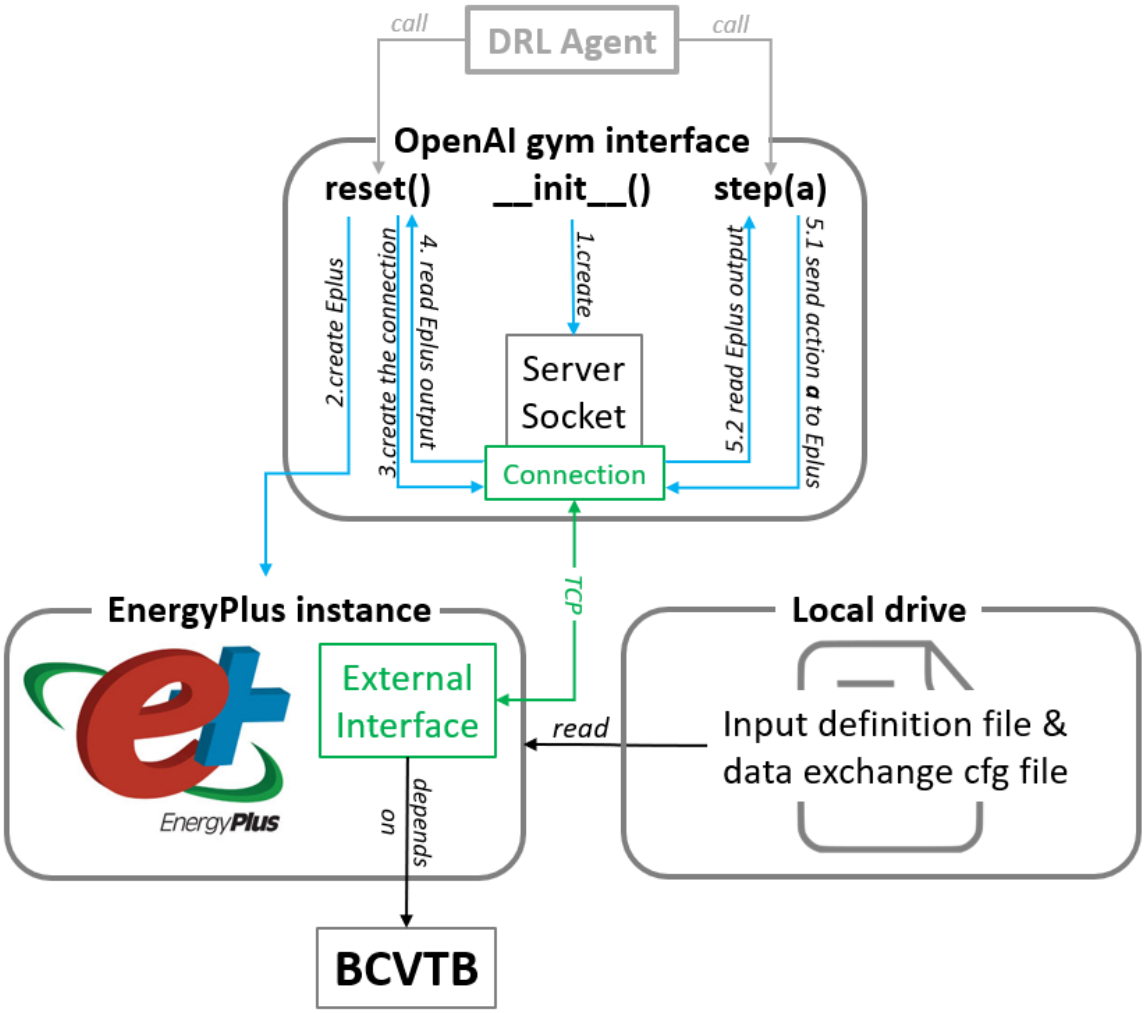

Figure 2.6: Architecture of the EnergyPlus Simulator for Reinforcement Learning

EPRL and a demo are available at https://github.com/zhangzhizza/Gym-Eplus.

Chapter 3

Experimental Design for the Control Framework Evaluation

This chapter describes the overall experimental design to sufficiently evaluate the proposed control framework for its convergence performance and control performance (i.e., energy saving and operational constraint fulfillment). The experiments are solely based on computer simulations.

3.1 Experimental Design Objectives

The behavior of an HVAC system is affected by several factors, including system type, climate, building thermal mass, operational pattern, etc. For example, an air-based system (supply hot/cold air to the rooms) usually has a much faster thermal response than a water-based system (supply hot/cold water to the rooms). Thus, the proposed control method should be evaluated under a variety of conditions to provide convincing conclusions on its effectiveness. However, most existing studies only evaluate their reinforcement learning methods under limited conditions. Hence, the first objective of the experimental design is to evaluate the proposed control framework under various conditions, including different system types, different climates, and different building thermal mass levels.

As a "learning" method, reinforcement learning uses gradient descent to solve a non-convex optimization problem. The convergence of the learning cannot be guaranteed, and is sensitive to the choice of hyperparameters, such as neural network model architecture, learning rate, optimizer, etc. The recent success of "deep" reinforcement learning motivates researchers to use complex neural network models, assuming complex models can achieve better energy efficiency performance. However, the reinforcement learning with complex neural network models is more difficult to converge. The existing studies have never compared the performance of different neural network models to justify the use of "deep" reinforcement learning. Hence, the second objective of the experimental design is to compare the performance of simple and complex neural network models for their convergence performance and control performance.

Table 3.1: Conditions for the Control Framework Evaluation

HVAC System Type	variable-air-volume terminal reheat system with an air-cooled heat pump in cooling season (VAVCooling)
	variable-air-volume terminal reheat system with an air-cooled heat pump in heating season (VAVHeating)
	radiant heating system (RadiantHeating)
	three-chiller chilled water system (ChilledWater)
Climate Zone	Pittsburgh (ASHRAE Climate Zone 5A)
	Beijing (ASHRAE Climate Zone 4)
	Shanghai (ASHRAE Climate Zone 3)
	Singapore (ASHRAE Climate Zone 1)
Thermal Mass Level	lightweight structure (metallic cladding based structure)[1]
	heavyweight structure (concrete based structure)[2]

1. The outmost layer of external walls is metallic cladding with thickness 0.006m, thermal conductivity 290W/m-K, density 1250kg/m^3, specific heat 1000J/kg-K, thermal absorptance 0.9, solar absorptance 0.4. 2. The outmost layer of external walls is concrete block with thickness 0.15m, thermal conductivity 1.63 W/m-K, density 2300kg/m^3, specific heat 1000J/kg-K, thermal absorptance 0.9, solar absorptance 0.6.

3.2 Experiment Scenarios

The control framework is evaluated for different HVAC systems, different climate zones, and different thermal mass levels, as shown in Table 3.1.

Four HVAC systems are selected, including a variable-air-volume (VAV) system for cooling, a VAV system for heating, a radiant heating system, and a three-chiller chilled water system. They represent some of the most common commercial system types. The details of the systems can be found in the following chapters.

Four climate zones are selected, ranging from the cool climate of Pittsburgh to the hot climate of Singapore. Figure 3.1 and 3.2 show the summary of the climates in the four locations from both typical meteorological year (TMY) data and the actual meteorological year (AMY) data of 2017. It can be seen that Pittsburgh has the coolest outdoor air temperature and is humid in both heating and cooling seasons. Beijing and Shanghai are warm and humid in cooling season, but Beijing is significantly drier and colder than Shanghai in heating season. Singapore is warm and humid,

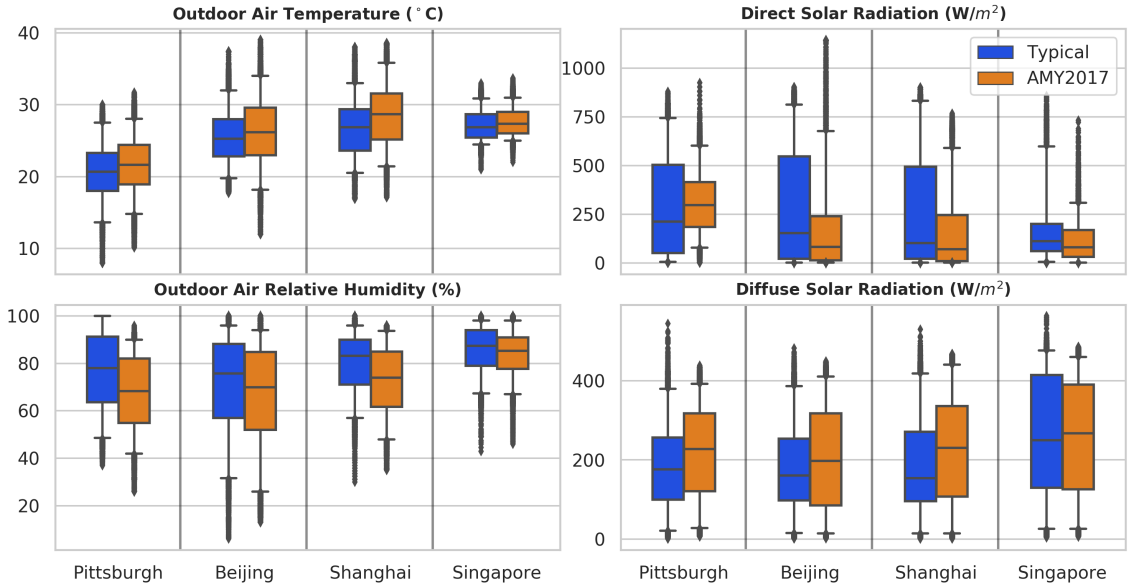

Figure 3.1: Boxplots for the Climates of the Four Locations in Cooling Season (the data is from June 1st to Aug 31st for Pittsburgh, Beijing and Shanghai, Sept 1st to Nov 30th for Singapore)

and the outdoor air temperature and relative humidity have small variations. The solar radiation data does not show a clear trend across the four locations, except that the direct solar radiation of Singapore has a smaller variation than the other locations.

Two thermal mass levels will be experimented, including a light metallic structure and a heavy concrete structure. Thermal mass levels affect a building's thermal response. The light structure has a faster thermal response than the heavy structure.

By combing all the possible conditions, 24 experiment scenarios are obtained as shown in Figure 3.3 and Figure 3.4. Building thermal mass is not applicable to ChilledWater because it is a primary-system (i.e., generate cooling source only) without considering room conditioning. Singapore is not included in the heating system scenarios because heating is not needed in a hot climate.

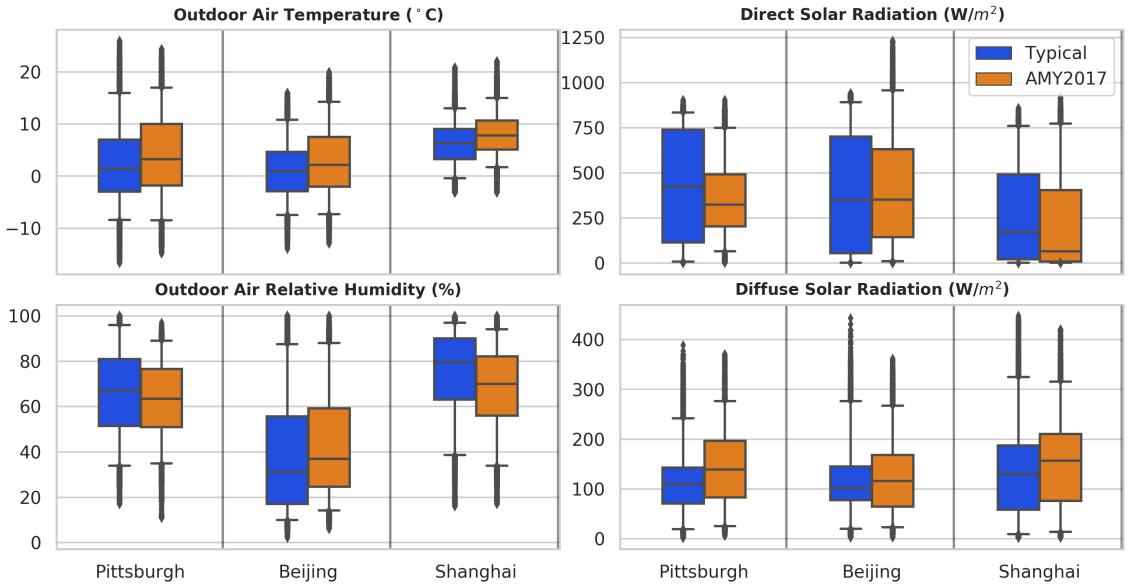

Figure 3.2: Boxplots for the Climates of the Three Locations in Heating Season (the data is from Jan 1st to Mar 31st for Pittsburgh, Beijing and Shanghai)

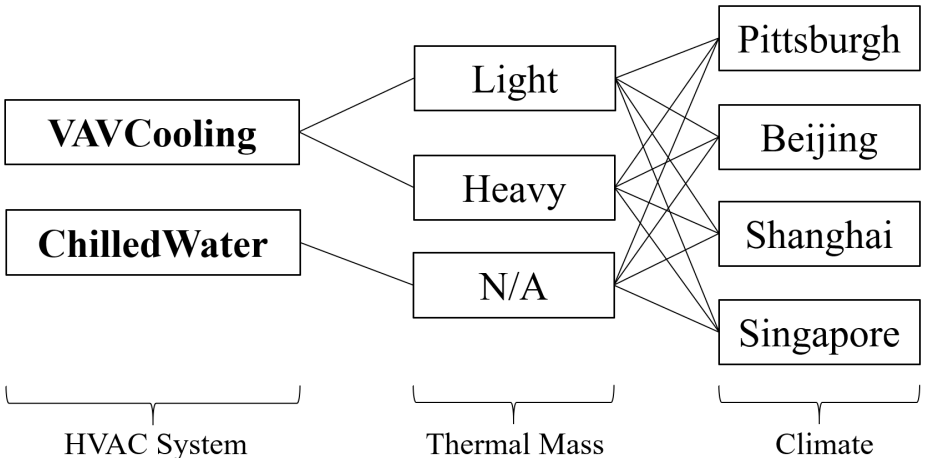

Figure 3.3: Experiment Scenarios for the Cooling Systems

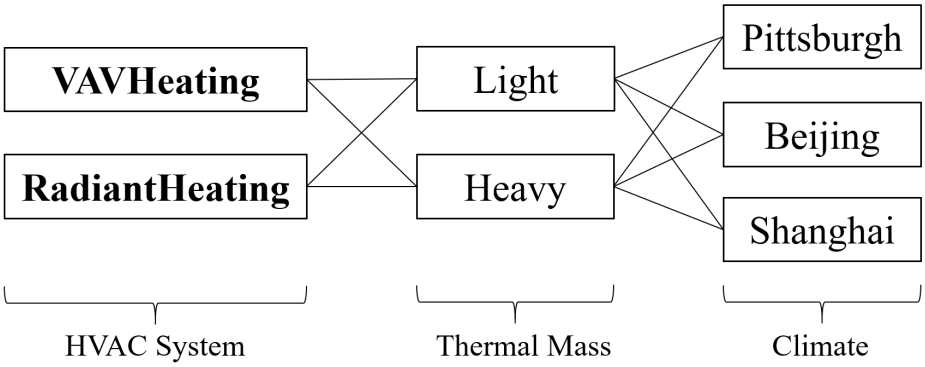

Figure 3.4: Experiment Scenarios for the Heating Systems

3.3 Neural Network Models

The control framework uses a neural network as the function approximation for an RL agent. The neural network architecture has the shared layers followed by a Softmax layer for the control policy and a linear layer for the state-value function, as shown in Figure 2.3. Different neural network models will be evaluated for the shared layers, ranging from a linear model to "deep" neural network models. Since neural network optimization is highly sensitive to the learning rate (α in Equation (2.10)), different learning rates will be tuned for each neural network model. The experiments on neural network models and learning rates are shown in Figure 3.5. In this figure, "ReLu x-y" means a feed-forward fully connected neural network that has y layers (the depth of a neural network) with x neurons (the width of a neural network) at each layer, and ReLu is the nonlinear activation of each neuron. The tuning for the neural network model and the learning rate is repeated for each experiment scenario listed in the previous section.

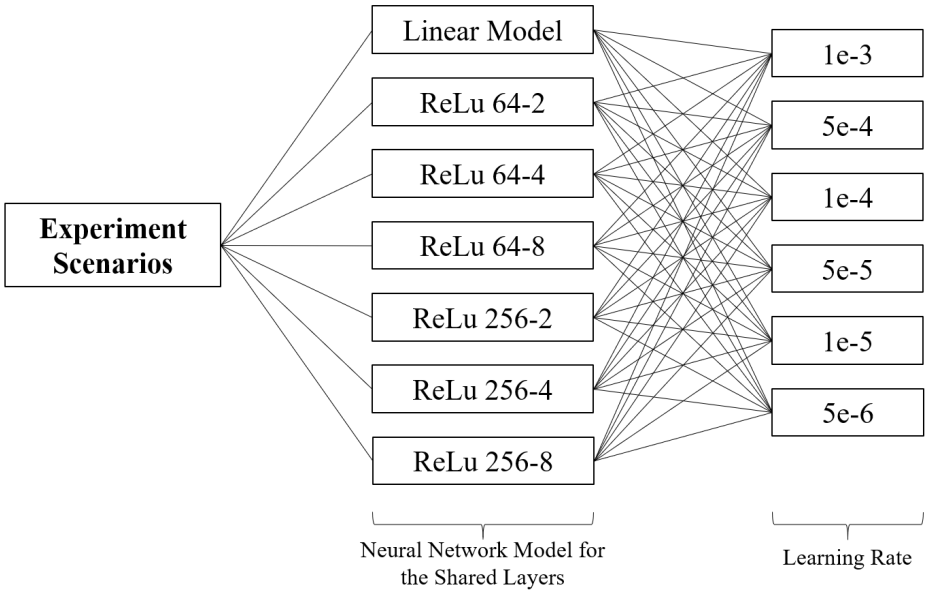

Figure 3.5: Experiments for the Neural Network Model Architecture and the Learning Rate

3.4 Experimental Procedure

This section presents a general experimental procedure for all the experiment scenarios. The detailed procedure for each specific experiment scenario can be found in the following chapters.

3.4.1 Preparation of a Training Simulator

1. Build a whole building energy model: An EnergyPlus model is built for each experiment scenario.

2. Set the simulation episode length: One simulation episode lasts for one cooling or heating season which is two or three months. Compared to a short simulation episode (e.g., several days), the long one could cover more weather variations to generate a more robust control policy and provide more comprehensive control performance results.

3. Set the simulation time step size: 10 minutes for all the experiment scenarios.

3.4.2 Offline Reinforcement Learning Training

1. Design the state/reward/action: They are designed based on the definitions in Section 2.4. Same HVAC system type will have the same design for the state, reward and action. More detailed information can be found in the following chapters.

2. Set the control time step size: it is 10 minutes for VAVCooling, VAVHeating, and ChilledWater systems; 20 minutes for RadiantHeating system because of its slow thermal response.

3. Set the neural network model and the learning rate: Use one of the combinations in Figure 3.5 as the neural network model and learning rate.

4. Set checkpoints during the training: Checkpoints are set at different interaction steps of an RL training process. At each checkpoint, the training is paused and the current control policy is backed up. Then, the current control policy is used to control the training simulator for one simulation episode to record the cumulative reward. This value is named "training cumulative reward" or $R_{trainCumulative}$.

5. Assess the convergence: The convergence is assessed by plotting the training evaluation history, which shows the training cumulative reward at different RL interaction steps. The training converges if the training cumulative reward increases and becomes stable.

6. Set the maximum interaction steps: Different RL agents need different numbers of interaction steps to converge. The maximum interaction steps are determined by trial-and-error for each RL agent. Basically, a large number is firstly guessed, and the number increases if $R_{trainCumulative}$ is not stable and has an increasing trend. If $R_{trainCumulative}$ has no sign of convergence (e.g., fluctuating at a low level), the maximum interaction steps will not be increased and the training is terminated.

7. Select a control policy: If the training converges, the "checkpointed" control policy with the maximum $R_{trainCumulative}$ is selected as the final output.

3.4.3 Control Performance Evaluation

Baseline Control Strategies The RL-trained control policies are compared with rule-based control (RBC) strategies as the baseline. RBC is widely used in practice for HVAC supervisory control,

so it is used as the baseline to evaluate the control framework against the industry common practice. The baseline control strategy stays the same for the experiment scenarios with the same HVAC system type.

The RL control policies are not compared with other optimal control methods, such as model predictive control (MPC), because they cannot be directly applied for the complex HVAC control problems as shown in this study. For example, the system dynamics of VAVCooling is nonlinear and even non-continuous, which is not solvable using common MPC methods.

Evaluation Metrics The RL-trained control policies are evaluated on HVAC energy consumption savings and operational constraints fulfillment. The HVAC energy consumption metric is system total electricity consumption (in kW, for VAVCooling, VAVHeating and ChilledWater) or system total heating demand (in kW, for RadiantHeating, because it is the only available energy metric for this system). The operational constraints vary in different HVAC system types. The details will be presented in the respective chapters of the experiment scenarios.

3.4.4 Evaluation Simulators

Each RL-trained control policy is evaluated in two types of simulators, including the training simulator for training evaluation and different "perturbed" simulators for "versatility" evaluation. The simulations last for one episode in both types of simulators.

Training Evaluation Simulator The training simulator is the same as the one that trains an RL agent. It is used to evaluate the "ideal" control performance of a trained RL control policy.

Versatility Evaluation Simulators An RL-trained control policy is obtained through the training based a design-stage BEM. The design-stage BEM contains assumed information for some HVAC operational conditions, such as weather conditions and building operation schedules. However, actual operational conditions may be different from the assumptions. For example, it is impossible to 100% accurately predict the weather conditions of a building even for a short future, and it is difficult to accurately predict the occupancy schedules of a building before it is built.

Hence, each RL-trained control policy is evaluated for its "versatility", which is the ability to tolerate the variations in HVAC operational conditions. The versatility evaluation is conducted based

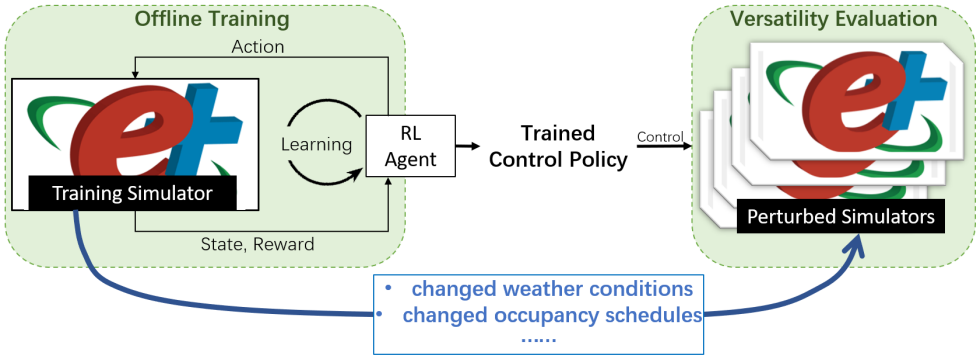

Figure 3.6: Versatility Evaluation for an RL-trained Control Policy

on different "perturbed" simulators, which are built based on the training simulator with changed HVAC operational conditions. Each RL-trained control policy is used to control the perturbed simulators, and the control performance is compared with that in the training simulator. This process is demonstrated in Figure 3.6.

The perturbed simulators contain the following variations in HVAC operational conditions:

- **Weather conditions**:
  - Training: typical weather data, such as Typical Meteorological Year (TMY) data or The International Weather for Energy Calculation (IWEC) data. These weather data are commonly used for building energy modeling because they are the most accessible and maybe the only available weather data during building design.
  - Perturbed: two different weather conditions will be used, including: 1) the Actual Meteorological Year (AMY)-2017 data; 2) the typical weather data (TMY3 or IWEC) with additive white Gaussian noises.

  The comparison between the typical weather conditions (without additive white Gaussian noise) and AMY-2017 weather conditions is shown in Figure 3.1 (cooling season) and Figure 3.2 (heating season). It can be seen that the distributions of the two weather conditions are different.

- **Occupancy schedules**:
  - Training: simple deterministic schedules with additive white Gaussian noises. Occupancy schedules are impossible to predict before a building is in operation. Thus, simple deterministic schedules are widely used for building energy modeling. Since it is known that

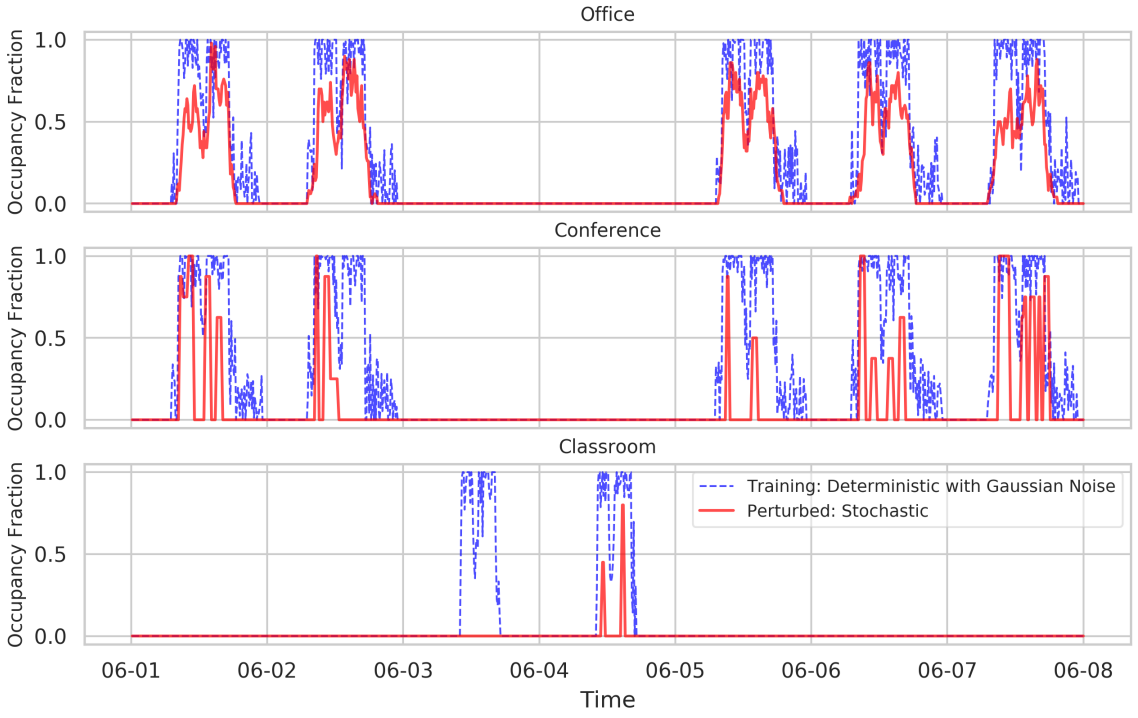

Figure 3.7: Comparisons of the Occupancy Schedules in the Training and Perturbed Simulators in a Selected Period (June 3rd and June 4th are weekends, all the other days are weekdays)

occupancy schedules are stochastic, additive white Gaussian noises are added to the basic deterministic schedules to increase the robustness of a reinforcement learning agent.

- Perturbed: 1) stochastic schedules generated by the Occupancy Simulator (Chen et al., 2018a).

Figure 3.7 shows the occupancy schedules in the training and perturbed simulators. It is assumed that office and conference rooms are only occupied in weekdays, and classrooms are only occupied in weekends (for outreach programs). Note that some experiment scenarios may not have all the room types. The detailed configuration of room types is presented in the respective chapter of each experiment.

- **Plug-load schedules**:
  - Training: simple deterministic schedules with additive white Gaussian noises. Similar to occupancy schedules, plug-load schedules are impossible to predict in the building design stage. Thus, deterministic schedules are used and white Gaussian noises are added for robustness.

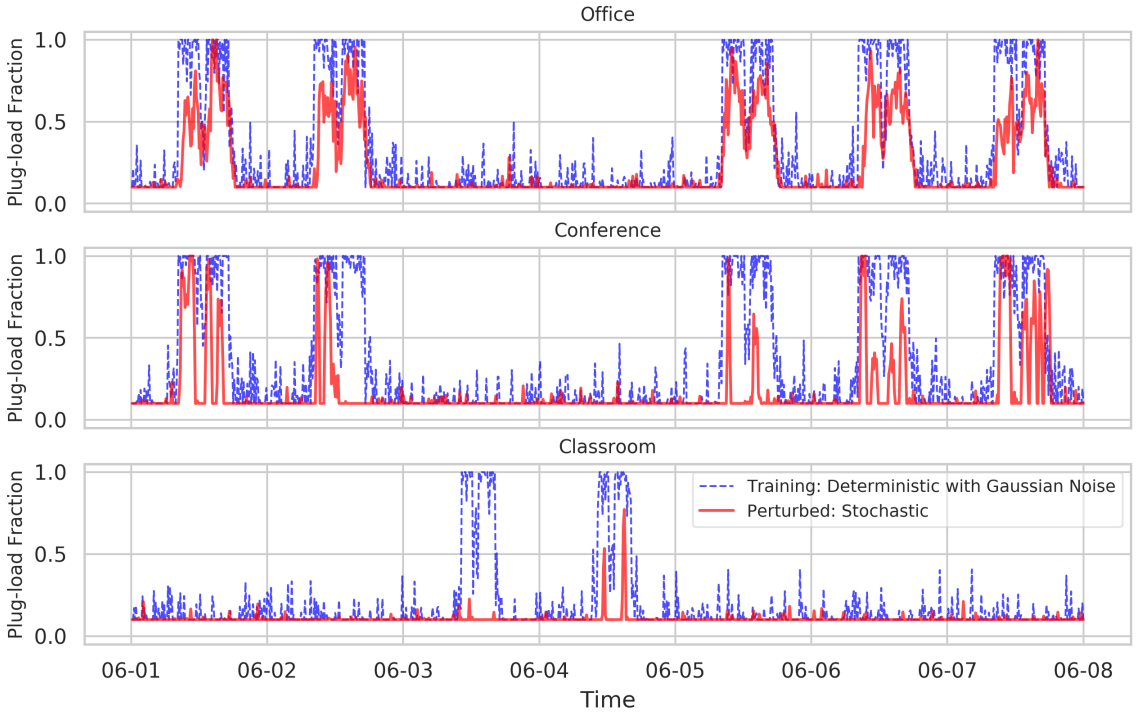

Figure 3.8: Comparisons of the Plug-load Schedules in the Training and Perturbed Simulators in a Selected Period (June 3rd and June 4th are weekends, all the other days are weekdays)

  - Perturbed: 1) stochastic schedules. It is generated by the stochastic occupancy schedules with additive white Gaussian noises because the usage of plug-load equipment is closely related to occupancy.

Figure 3.8 shows the plug-load schedules in the training and perturbed simulators. The profiles are similar to the occupancy schedules. Some experiment scenarios may not have all the room types. The detailed room type configuration is presented in the respective chapter of each experiment.

- **Indoor air temperature (IAT) setpoint schedules**:
  - Training: simple deterministic schedules with additive white Gaussian noises. The IAT setpoint of a building is also subjective to the actual building operation. During building design, engineers usually assume a constant temperature with night setback (e.g., 24°C in the day and 18°C in the night).
  - Perturbed: two different IAT setpoint schedules will be used, including: 1) Predicted Mean Vote (PMV)[1]-based schedules (this reflects a condition that occupants have access

[1]Predicted Mean Vote is a calculated thermal comfort metric based on Fanger's model (Fanger, 1970). Its value

to thermostats so IAT setpoint changes according to their thermal comfort responses); 2) deterministic schedules (this reflects a condition that occupants have no access to thermostats so IAT setpoint is pre-set by facility managers).

Note that the PMV-based setpoint is a dynamic schedule. Its profile is affected by several building operational conditions, such as weather conditions, internal loads, HVAC control strategies, etc. The equation of the PMV-based IAT setpoint is shown below,

$$StoStpt_{heating,t,i} = DetStpt_{heating,t,i} + Adj_{t,i}, \tag{3.1a}$$

$$StoStpt_{cooling,t,i} = DetStpt_{cooling,t,i} + Adj_{t,i}, \tag{3.1b}$$

$$Adj_{t,i} = \begin{cases} -\left(2.0 * PMV_{t,i} + [\mathcal{N}(0, 0.5)]_{-1}^{1}\right) & Occp = 1 \text{ and } |PMV_{t,i}| > 0.5, \\ Adj_{t-1,i} & \text{All other cases,} \end{cases} \tag{3.1c}$$

where subscript t and i represent a control time step and a zone, $StoStpt_{heating}$ and $StoStpt_{cooling}$ are the PMV-based IAT heating and cooling setpoint, $DetStpt_{heating}$ and $DetStpt_{cooling}$ are the deterministic IAT heating and cooling setpoint, PMV is calculated predicted mean vote based Fanger's model (Fanger, 1970), $\mathcal{N}(0, 0.5)$ is a number sampled from a Gaussian distribution with 0 mean and 0.5 standard deviation, $Occp$ is occupancy flag. Intuitively, the PMV-based setpoint mimics the behaviors of occupants, who increase IAT setpoint when they feel cool (when $PMV < -0.5$), decrease IAT setpoints when they feel warm (when $PMV > 0.5$), and do nothing when they feel neural (when $-0.5 \leq PMV \leq 0.5$).

Figure 3.9 and 3.10 show the comparisons of different IAT cooling and heating setpoint schedules. It should be noted that the PMV-based setpoint does not have a fixed profile. The resulting schedule is different in different conditions because PMV is affected by weather conditions, internal loads, HVAC control strategies, etc. The PMV-based setpoint of the figures is from a specific room in a specific simulator. It can be seen that the PMV-based setpoint has large differences from the deterministic one in some periods.

Note that the above perturbations may not apply to all the experiment scenarios. The details about the perturbed simulators are described in the respective chapter of each experiment.

ranges from -3 to +3 where -3 represents "cold" and +3 represents "warm".

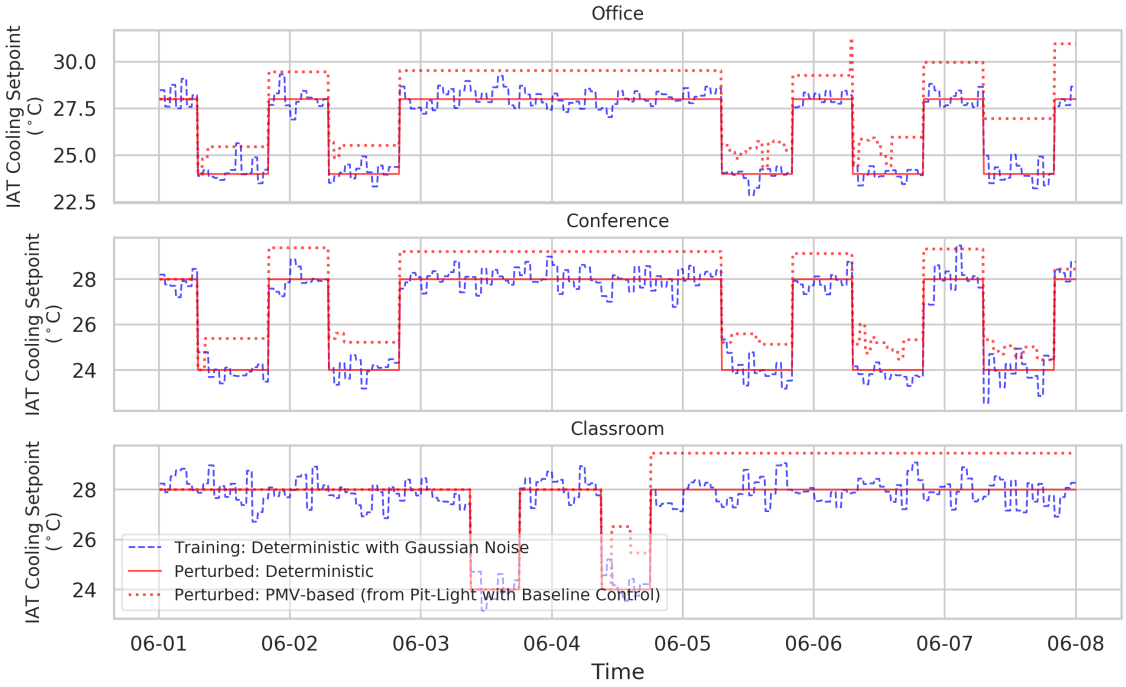

Figure 3.9: Comparisons of the IAT Cooling Setpoint Schedules the Training and Perturbed Simulators for a Selected Time Period in Cooling Season (Note: 1: the shown PMV-based schedule is from a zone in the training simulator of VAVCooling with Pittsburgh climate, lightweight structure and baseline control strategy; 2: June 3rd and 4th are weekends and the other days are weekdays)

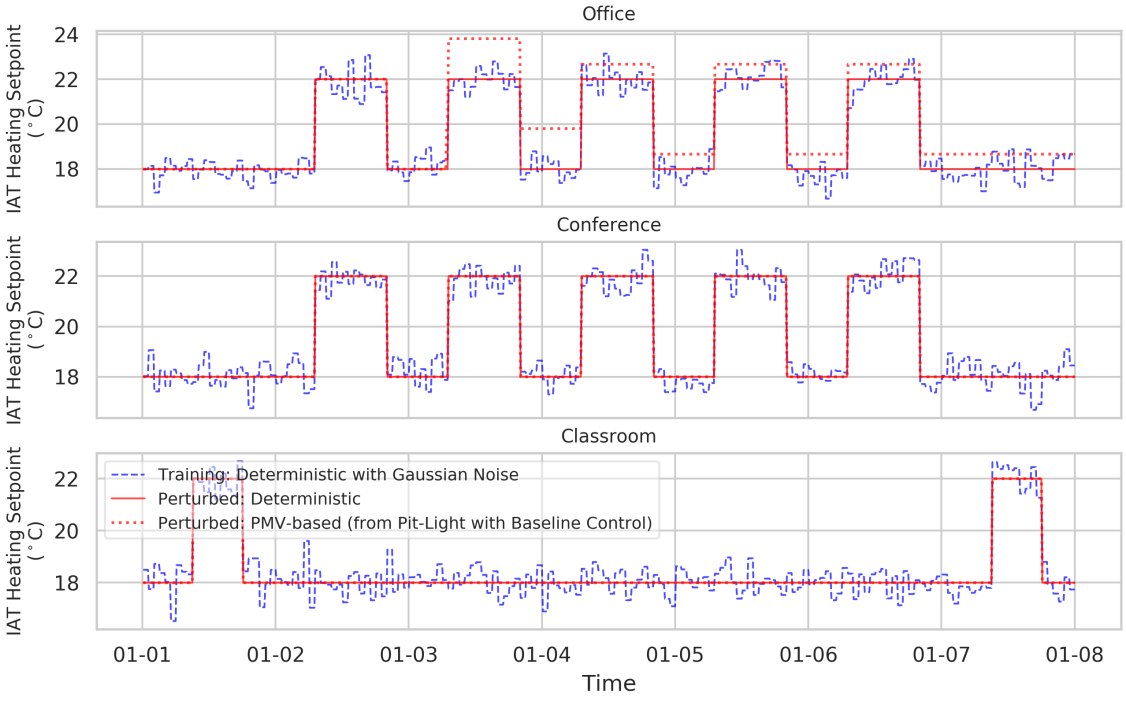

Figure 3.10: Comparisons of the IAT Heating Setpoint Schedules the Training and Perturbed Simulators for a Selected Time Period in Heating Season (Note: 1: the shown PMV-based schedule is from a zone in the training simulator of VAVHeating with Pittsburgh climate, lightweight structure and baseline control strategy; 2: Jan 1st and 7th are weekends and the other days are weekdays)

Chapter 4

Experiment 1: VAVCooling

This chapter presents the experiments related to VAVCooling, as shown in Figure 4.1. There are 8 scenarios related to this system, for two thermal mass levels and four climate zones. For each scenario, seven different neural network models and six learning rates are tuned.

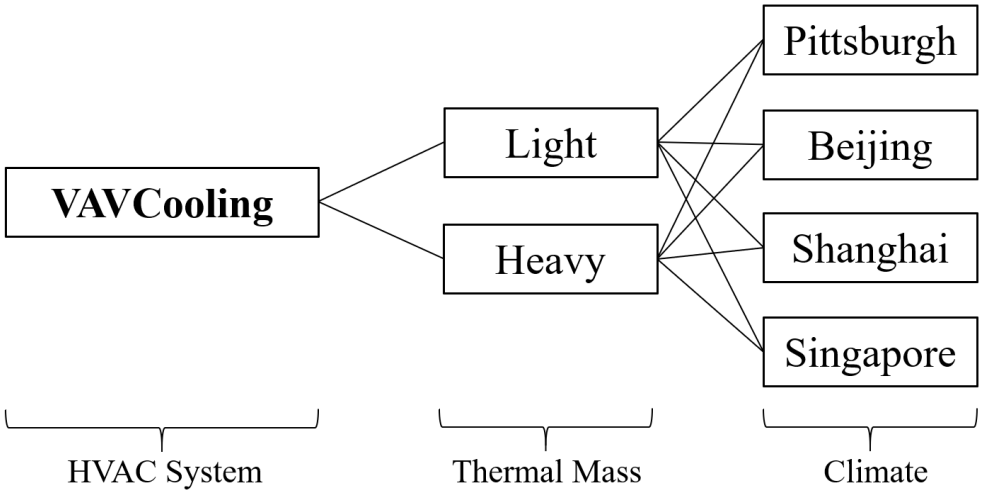

Figure 4.1: VAVCooling Experiment Scenarios

4.1 HVAC System Description

4.1.1 System Layout

VAVCooling is a variable-air-volume system with terminal reheat. The system layout is shown in Figure 4.2. This system has a centralized air handling unit to condition outdoor air and return air. The conditioned air is supplied to all building zones. In the air handling unit, there is a solid-desiccant air dehumidifier, a heat-recovery wheel, an air-cooled heat pump (provide either heating or cooling) and a variable-speed fan. In each zone, there is an air damper with electric heating coils.

The system follows common operation strategies of VAV systems. In cooling mode, the air damper at each zone adjusts the air flow rate to meet the zone air temperature setpoints. Basically, if the zone air temperature is above its cooling setpoint, the air damper opens more to supply more conditioned air; if the zone air temperature is below its heating setpoint, the air damper opens less to reduce the air flow rate. If the zone air flow rate is at the minimum (10% of the maximum airflow rate) but the zone air temperature is still below its heating setpoint (the zone is over-cooled), the electric heater in the air damper will be turned on to provide additional heating (this may happen in some zones even in cooling season).

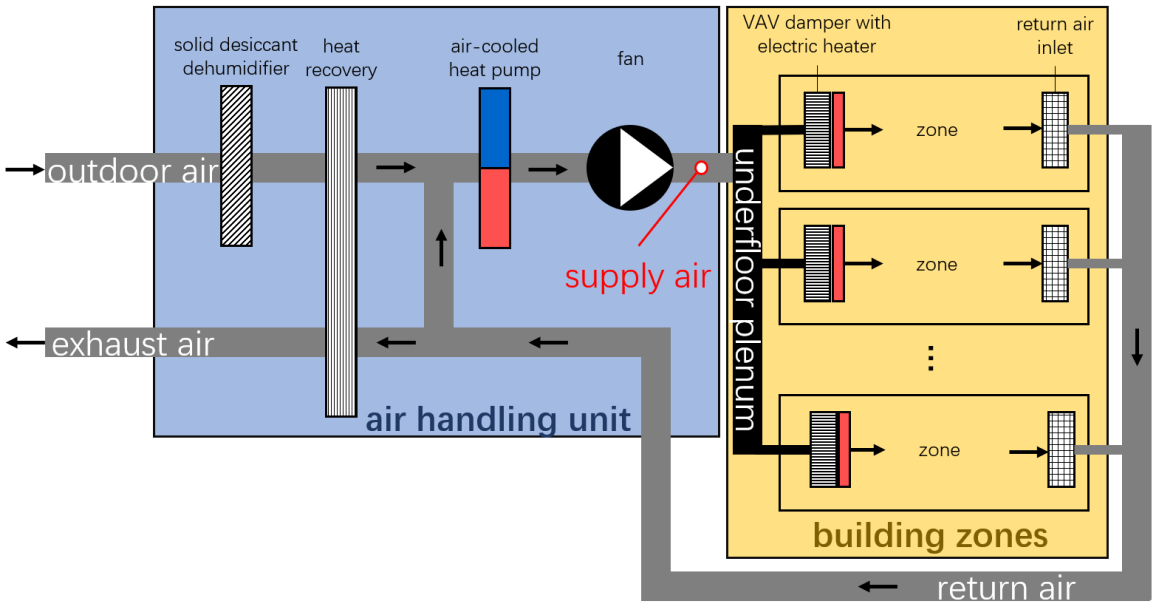

Figure 4.2: System Layout of VAVCooling

4.1.2 *Thermal Zones and Envelopes*

This system serves a two-level building with 22 conditioned thermal zones. The thermal zones include open-plan office rooms, conference rooms, and a classroom, as shown in Figure 4.3. The office rooms and conference rooms are occupied at regular work hours, and the classroom is only occupied on weekends.

The thermal properties of the envelopes (external walls, roofs, ground floors, and windows) follow the requirements of ASHRAE 90.1-2016 (American Society of Heating, Ventilating, and Air

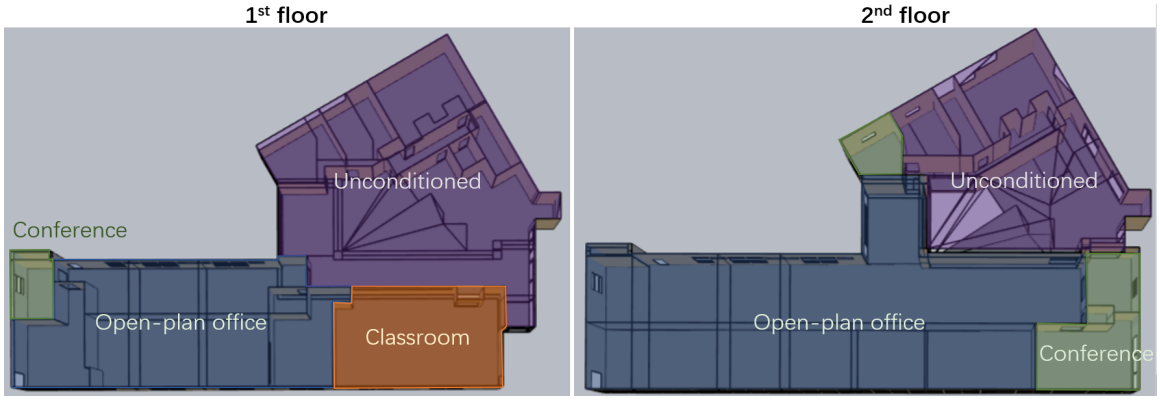

Figure 4.3: Room Functions of the Thermal Zones Served by VAVCooling

Conditioning Engineers, 2016).

4.1.3 Target Control Variable and Baseline Control Strategy

The target supervisory control variable is the supply air temperature setpoint (as illustrated in Figure 4.2). This setpoint is selected because it has deep effects on the operation of all HVAC components (Zhao, 2015; Jia et al., 2019). For example, in cooling season, the colder supply air temperature may lead to reduced supply airflow rate, reduced outdoor air intake, reduced fan power, reduced dehumidification load, increased or reduced cooling power and increased terminal reheating power if some zones are overcooled.

The baseline control strategy for this setpoint is a built-in function of EnergyPlus called "warmest", which use the cooling load of the warmest zone to determine the supply air temperature setpoint (T_{sa}), as shown below:

$$T_{sa,t} = \left[\min_{i \in \text{all zones}} T_{ia,i,t-1} + \frac{\dot{Q}_{cooling,i,t-1}}{C_{p,air} \dot{m}_{max,i}} \right]_{T_{sa}}^{T_{sa}}, \tag{4.1}$$

where subscript t is one control time step, i is one zone, T_{ia} is zone indoor air temperature, $\dot{Q}_{cooling}$ is zone cooling load, $C_{p,air}$ is specific heat of air, $\dot{m}_{max}$ is zone maximum supply air mass flow rate, T_{sa} and T_{sa} are the high and low limit of the supply air temperature setpoint (in this case, 24°C and 12°C). The baseline control strategy provides the highest possible supply air temperature setpoint that could satisfy all zones' cooling demands. This strategy may reduce cooling energy consumption but may result in increased fan energy consumption. This is a commonly used strategy among EnergyPlus users.

4.1.4 Whole Building Energy Model

EnergyPlus version 8.3 is used to generate whole building energy models for the system. The geometry 3D rendering is shown in Figure 4.4.

The capacities of the components in the system (e.g., fan, heat pump, air dampers, etc.) are auto-sized by EnergyPlus based on the design conditions of each climate zone. One simulation episode lasts for 3-month with 10-min as the simulation time step, as shown in Table 4.1.

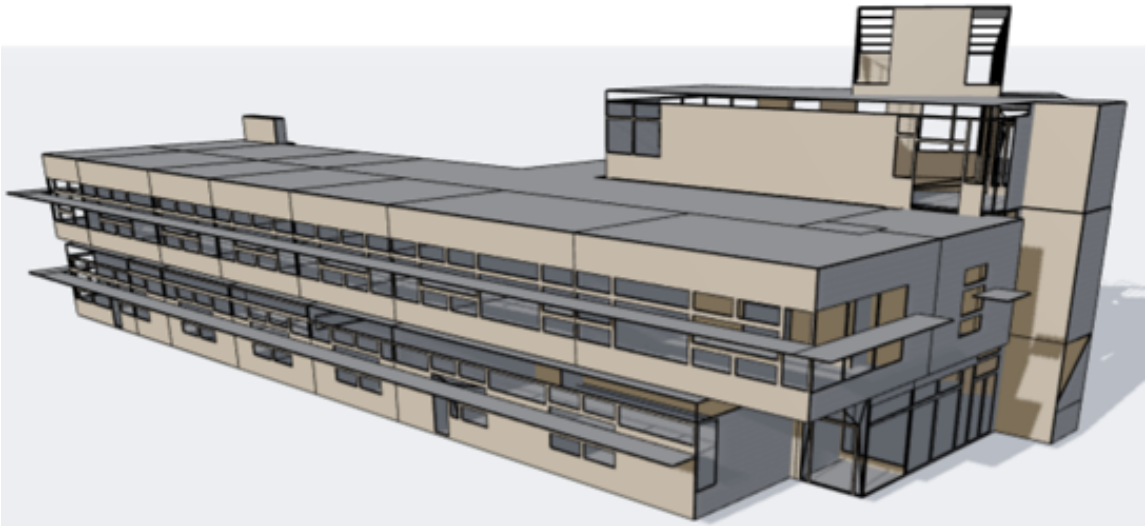

Figure 4.4: 3D Rendering of the Geometry of the Whole Building Energy Model for VAVCooling (rendered by BuildSimHub, Inc. (2018))

Table 4.1: Basic Simulation Settings of the Whole Building Energy Models for the VAVCooling Scenarios

Climate	**Thermal Mass**	**Simulation Period**	**Simulation Time Step**
Pittsburgh	Light / Heavy	June 1st-Aug 31st	10-min
Beijing	Light / Heavy	June 1st-Aug 31st	10-min
Shanghai	Light / Heavy	June 1st-Aug 31st	10-min
Singapore	Light / Heavy	Sept 1st-Nov 30th	10-min

Table 4.2: Comparison of the Training and Perturbed Simulators for the VAVCooling Scenarios

	Training	**Perturbed**			
		1	**2**	**3**	**4**
Weather	TMY3 for Pittsburgh, IWEC for other locations	AMY 2017	TMY/IWEC with additive white Gaussian noise	AMY 2017	TMY3/IWEC with additive white Gaussian noise
Occupancy Schedule	Deterministic with additive white Gaussian noise	Stochastic			
Plug-load Schedule	Deterministic with additive white Gaussian noise	Stochastic			
IAT Setpoint	Deterministic with additive white Gaussian noise	PMV-based		Deterministic	

4.2 Training and Perturbed Simulators

As specified in section 3.4.4, the control performance will be evaluated in the training simulator and perturbed simulators. The perturbed simulators are varied from the training simulator in weather conditions, occupancy schedules, plug-load schedules and indoor air temperature (IAT) setpoint schedules. The perturbed simulators are used to evaluate the versatility of the trained RL control policies. The configurations of the training simulator and the perturbed simulators are shown in Table 4.2.

Table 4.3: Observation Vector in the State for VAVCooling

No.	Item
1	Is weekday or not
2	Hour of the day
3	Outdoor air temperature (°C)
4	Outdoor air relative humidity (%)
5	Diffuse solar radiation (W/m^2)
6	Direct solar radiation (W/m^2)
7-28	Zone air temperature (of 22 zones, °C)
29-50	Zone cooling setpoint temperature (of 22 zones, °C)
51-72	Zone heating setpoint temperature (of 22 zones, °C)
73	Total HVAC Electric Demand (kW)

4.3 Reinforcement Learning Setup

4.3.1 State Design

As specified in section 2.4.1, the state is a stack of the current and historical observations. Two variables should be determined, including the observation vector and the length of the history in the state.

The observation vector (*ob* in Equation (2.19)) includes the sensor data of the VAV system, as shown in Table 4.3.

The length of the history in the state is determined using the method described in section 2.4.1. The process is repeated for all the experiment scenarios based on the data from the training simulators. Control time step length is 10-minute. Figure 4.5 shows the relationships between the time interval n and the distance correlation $dcor_n$. A large $dcor_n$ means there is a strong dependence between the observations at $t+1$ (i.e., $\mathbf{ob_{t+1}}$) and the observations at $t-n$ (i.e., $\mathbf{ob_{t-n}}$), and vice-versa. It can be seen in the figure that $dcor_n$ decays slower in the heavyweight scenarios of Pittsburgh, Beijing, and Shanghai. In the Singapore scenarios, $dcor_n$ has similar behaviors in both lightweight and heavyweight simulators. This may be attributed to the stable weather conditions of Singapore so thermal mass has less effect on building thermal behaviors. In addition, $dcor_n$ repeats its pattern

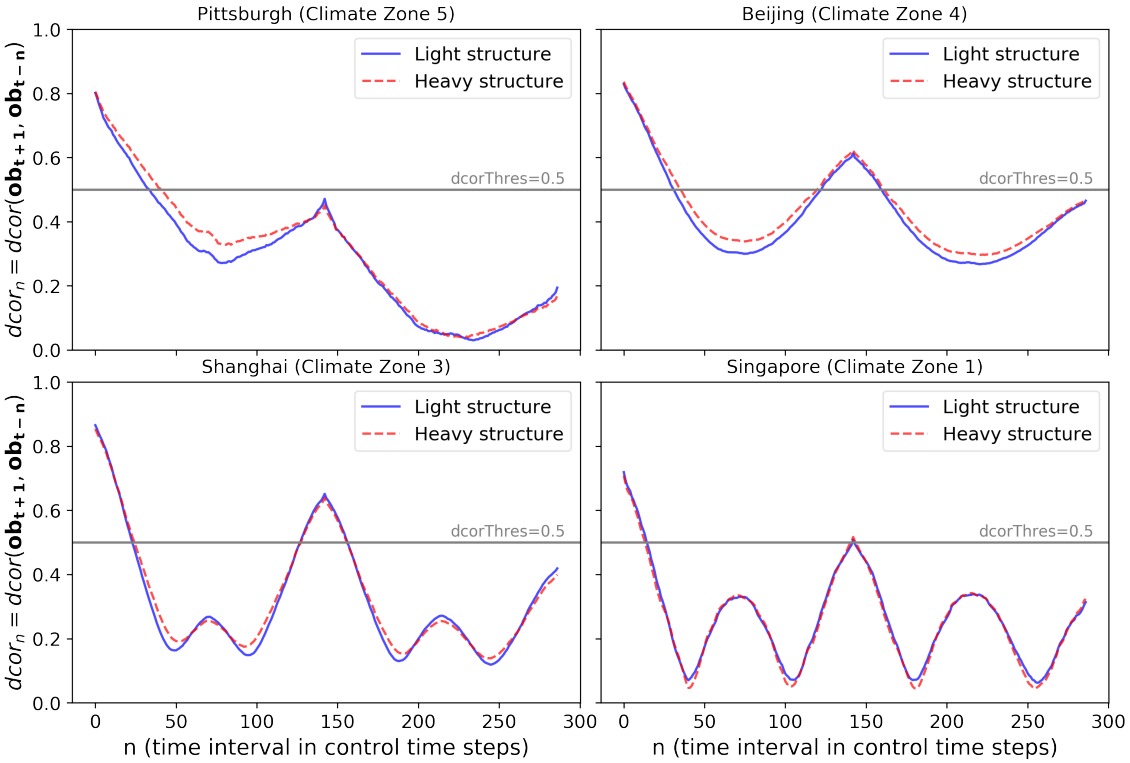

Figure 4.5: Relationship Between the Time Interval n and $dcor_n$ (specified in Equation (2.23)) for All the VAVCooling Scenarios

after $n = 144 = 1$ day. This is because both weather conditions and building operations have daily cyclic patterns.

By using Algorithm 1 with dcorThres as 0.5, the length of the history in the state is determined, as shown in Table 4.4.

4.3.2 Action Design

The discrete action space for the supply air temperature setpoint is:

$$A_{vavcooling} = \{12°C, 12.5°C, ..., 23.5°C, 24°C\} \tag{4.2}$$

Table 4.4: Length of the History in the State for VAVCooling Scenarios

Climate	Thermal Mass	Length of History (in control time steps)
Pittsburgh	Light	34 (5.7 hr)
	Heavy	41 (6.8 hr)
Beijing	Light	31 (5.2 hr)
	Heavy	36 (6.0 hr)
Shanghai	Light	23 (3.8 hr)
	Heavy	25 (4.2 hr)
Singapore	Light	15 (2.5 hr)
	Heavy	14 (2.3 hr)

4.3.3 Reward Design

The reward determines the control objective of reinforcement learning. In the VAVCooling scenarios, the control framework aims to minimize the total HVAC electric energy consumption and minimize the indoor air temperature setpoint notmet time[1]. As a result, the reward function is:

$$
\begin{aligned}
R_{vavcooling,t} &= 1.0 - [P_{energy,t} + P_{comfort,t}]_0^1, \\
&\text{where,} \\
P_{energy,t} &= \beta * E_{hvac,t}, \\
P_{comfort,t} &= \tau * \left[\max\left([\mathbf{T_{ia,t}} - \mathbf{T_{clgstpt,t}}]^+\right) + \max\left([\mathbf{T_{htgstpt,t}} - \mathbf{T_{ia,t}}]^+\right)\right].
\end{aligned}
\tag{4.3}
$$

In the above function, subscript t is one control time step, β and τ are tunable hyperparameters controlling the weights for the energy penalty and setpoint notmet penalty, E_{hvac} is the normalized total HVAC electric demand, and $\mathbf{T_{ia}}, \mathbf{T_{clgstpt}}, \mathbf{T_{htgstpt}}$ are the normalized indoor air temperature, cooling setpoint and heating setpoint of all the 22 zones

4.3.4 Hyperparameters

The reinforcement learning agents are trained using the training simulators with the hyperparameters shown in Table 4.5. For the training of each experiment scenario, 7 neural network models and 6

[1]setpoint notmet time means the cumulative time that the indoor air temperature of all the 22 zones is either above the cooling setpoints or below the heating setpoints

Table 4.5: Hyperparameters for the RL Training for the VAVCooling Scenarios

Item	**Value**	**Item**	**Value**
Simulation time step	10-min	A3C local agent number	16
Control time step	10-min	Reward discount factor	0.99
Nonlinear activation*	ReLu	RL interaction steps	5M
Optimizer	RMSProp	Learning batch size	5
RMSProp decay rate	0.99	Value loss weight	0.5
RMSProp momentum	0.0	τ in the reward^{+}	1.0
RMSPorp epsilon	$1e^{-10}$	β in the reward	1.2
Gradient clip method	L2-norm	Gradient clip threshold	5.0

Note: * nonlinear activation applies to the shared layers of the neural networks (except the one with the linear shared layers); $^{+}$ $\tau = 1$ means that when the setpoint notmet of any zones is larger than 1°C, the reward is zero.

learning rates will be tuned, as shown in Figure 3.5.

4.4 Results

4.4.1 Convergence Results

This section shows the results related to the convergence performance of the reinforcement learning training.

Convergence Robustness to Different Learning Rates

Reinforcement learning problems are sensitive to the learning rate. Inappropriate choices of the learning rate may lead to training divergence. However, there is not a well-established theory to pre-determine the best learning rate choice. Its number must be tuned.

This study tunes six learning rates, from 1e-3 to 5e-6, for all the neural network models. Figure 4.6 shows the count of convergence out of the six learning rates vs. the neural network models. A larger count means a neural network model is more robust to different learning rates for convergence. As a general trend, the shallower neural network models (the models with fewer layers) are more robust for convergence than the deeper neural network models (the models with more layers). In all the scenarios, the linear model is easier to converge than the 4-layer and 8-layer models. The width of a neural network model (the number of neurons per layer) does not affect the convergence robustness to the learning rate. The results align with the expectation because deep neural networks suffer vanishing or exploding gradients during gradient descent optimization (Goodfellow et al., 2016).

Figure 4.7 shows the relationship between the convergence count out of the seven neural network models vs. the six learning rates. A larger count means the corresponding learning rate is more favorable for convergence. It is clear that a small learning rate is more favorable than a large learning rate for all the experiment scenarios.

Convergence History

To further analyze the convergence behaviors of the neural network models, Figures 4.8, 4.9, 4.10 and 4.11 show the training evaluation history of the seven neural network models at the same learning rate (1e-5). The training evaluation history is the total evaluation reward in the training simulator for one simulation episode at different RL interaction steps. For example, when an RL

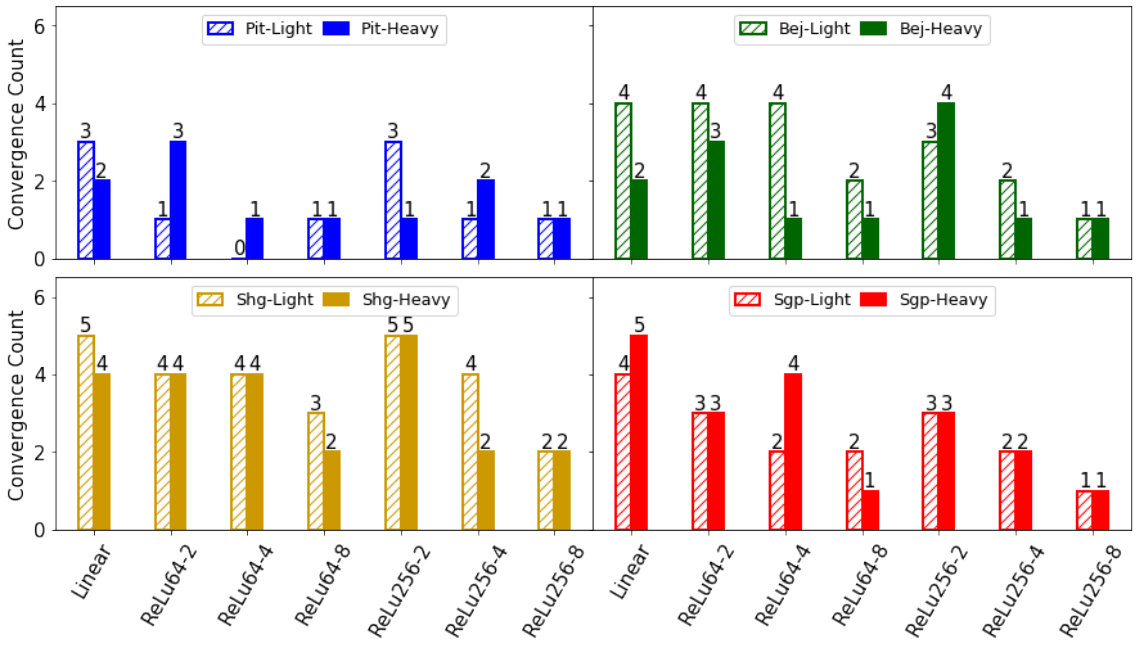

Figure 4.6: VAVCooling: Convergence Robustness to the Learning Rate (count of convergence out of the six learning rates) vs. Neural Network Models for All the Experiment Scenarios

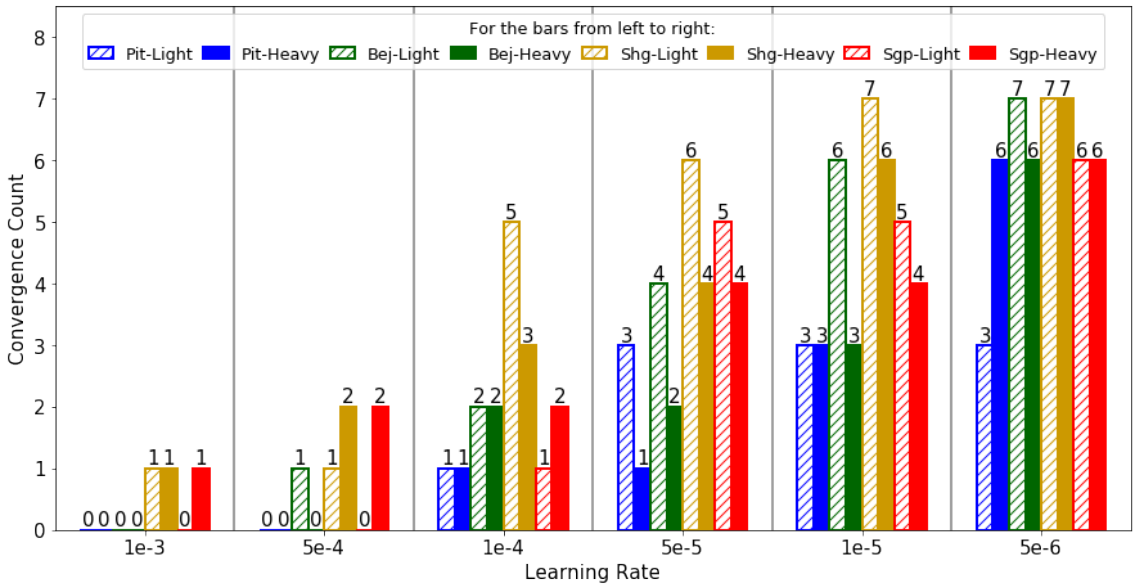

Figure 4.7: VAVCooling: Convergence Count out of the Seven Neural Network Models vs. the Learning Rate for All the Experiment Scenarios

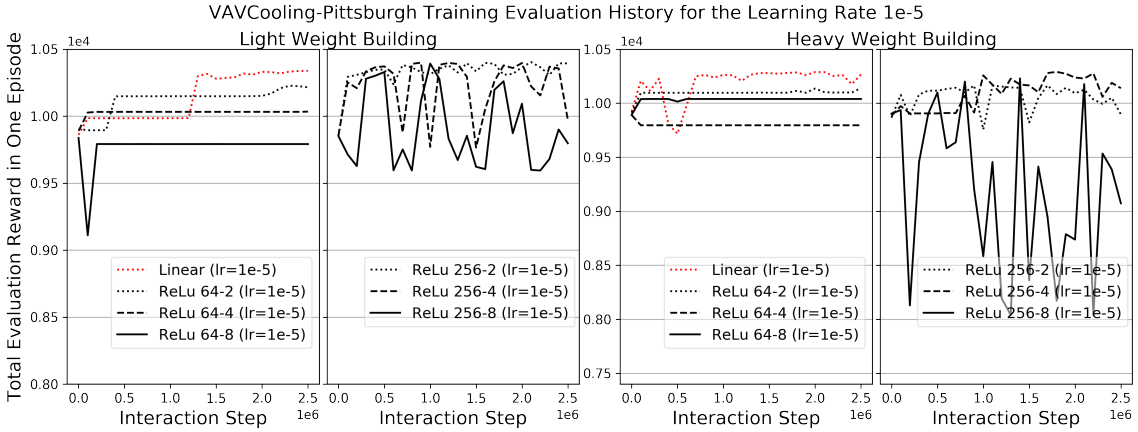

Figure 4.8: VAVCooling: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Pittsburgh Climate)

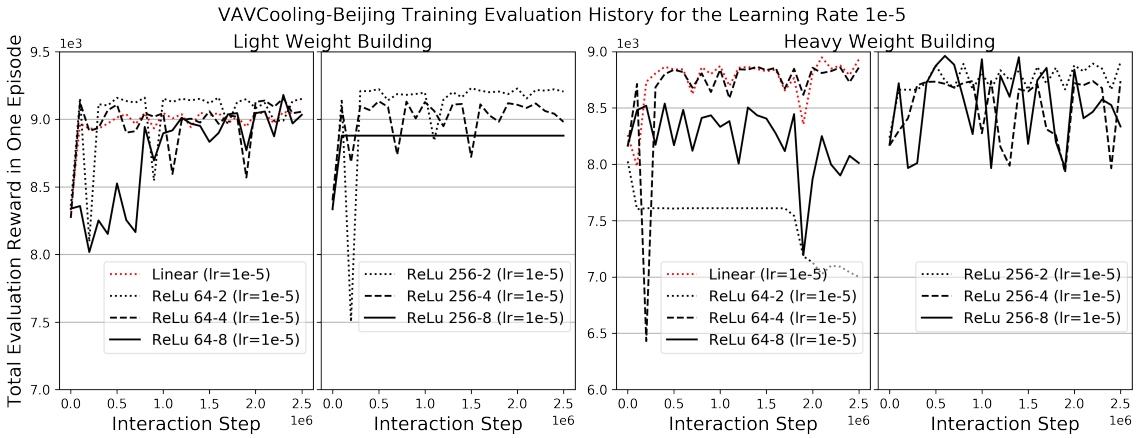

Figure 4.9: VAVCooling: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Beijing Climate)

agent interacts with its environment after 1-million times (in this study we use A3C, so the 1-million times are jointly from 16 local RL agents), it pauses the training and uses the current trained control policy to control the training simulator for one episode. The cumulative reward obtained in this one-episode simulation is the total evaluation reward at the interaction step 1-million.

As shown in the Figures of the training evaluation history, a deep neural network has more fluctuations and is easier to diverge and saturate (i.e., when the total evaluation reward remains constant) than a shallow neural network. The linear model has smooth convergence histories for all the experiment scenarios. The neural network width and building thermal mass level have no obvious effects on the training evaluation history.

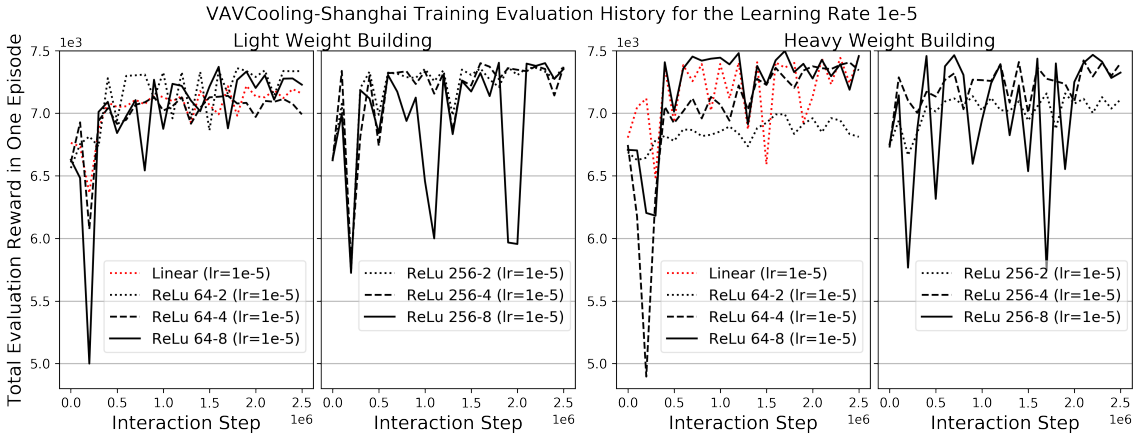

Figure 4.10: VAVCooling: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Shanghai Climate)

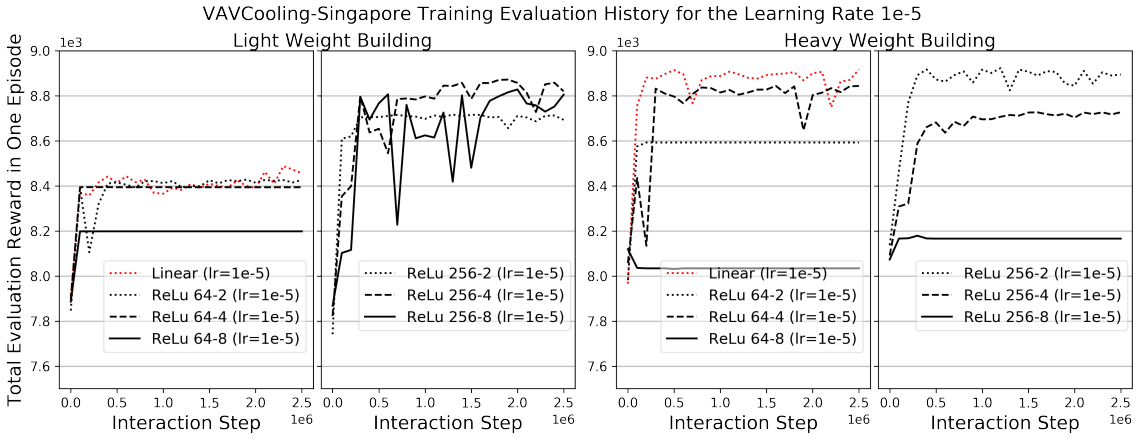

Figure 4.11: VAVCooling: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Singapore Climate)

4.4.2 Control Performance

The control performance of the reinforcement learning agents is evaluated by the percentage saving of the total HVAC electricity consumption (E_{saving}) and setpoint notmet time (Ti_{nmt}). E_{saving} is

$$E_{saving} = \frac{E_{baseline} - E_{rl}}{E_{baseline}} * 100, \tag{4.4}$$

where E_{rl} is the total HVAC electricity consumption using a trained RL agent, $E_{baseline}$ is the total HVAC electricity consumption using the baseline control strategy. Ti_{nmt} is

$$\begin{aligned} Ti_{nmt} = Ti_{simstep} * \sum_{t=0}^{T_{simend}} \Big(& \Big[\sum_{i \in \text{all zones}} \big((T_{ia,t,i} - T_{clgstpt,t,i}) > 0.5\big) \Big]_0^1 \\ & + \Big[\sum_{i \in \text{all zones}} \big((T_{htgstpt,t,i} - T_{ia,t,i}) > 0.5\big) \Big]_0^1 \Big), \end{aligned} \tag{4.5}$$

where $Ti_{simstep}$ is the time step length of the simulation (10-min in this case), t is one time step in the simulation, T_{simend} is the number of time steps in one simulation episode, i is one conditioned zone, T_{ia} is zone air temperature (°C), $T_{clgstpt}$ is zone cooling setpoint (°C) and $T_{htgstpt}$ is zone heating setpoint (°C). Intuitively, Ti_{nmt} shows the cumulative time that at least one zone cannot meet its cooling or heating setpoint. ASHRAE 90.1-2016 requires that Ti_{nmt} for one year is less than 300 hours.

Figures 4.12, 4.13, 4.14 and 4.15 show the control performance of the reinforcement learning method in the four climates. The configurations of the different simulators are described in Table 4.2. In the figures, it can be seen that:

- In most experiment scenarios (except Perturbed1 and Perturbed2 in the Pittsburgh-Lightweight-Building and Pittsburgh-Heavyweight-Building scenarios as shown in Figure 4.12), the reinforcement learning method can achieve significant HVAC energy savings and smaller setpoint notmet time compared to the baseline control strategy. The specific control performance is different in different climates and different thermal mass levels.

- There is no obvious relationship between a neural network model and the control performance. The linear model can achieve similar control performance with the other more complex neural network models, except in the Beijing-Lightweight-Building scenario (as shown in Figure 4.13). In this scenario, the linear model leads to smaller energy savings in the Perturbed1 and

Perturbed2 simulators than the other neural network models.

- There is no significant difference in the control performance between the lightweight-building scenarios and heavyweight-building scenarios. This means the reinforcement learning method is robust for different thermal mass levels in VAVCooling.

- In the Pittsburgh scenarios (Figure 4.12) and Beijing-Lightweight-Building scenario (Figure 4.13), the control performance in the Perturbed1 and Perturbed2 simulators is worse than that in the training simulator. This phenomenon does not occur in the Perturbed3 and Perturbed4 simulators. Compared to the Perturbed3 and Perturbed4 simulators, the Perturbed1 and Perturbed2 simulators have PMV-based indoor air temperature setpoint, which may lead to a significant different setpoint distribution from that in the training simulator. This means that the reinforcement learning agents do not tolerate the variations of the indoor air temperature setpoint.

- The control performance in the Perturbed3 and Perturbed4 simulators is better than that in the training simulator for all the experiment scenarios. The Perturbed3 and Perturbed4 simulators have the deterministic indoor air temperature setpoint, which has a similar distribution as the setpoint in the training simulator. However, the two perturbed simulators have different weather data and occupancy/plug-load schedules from the training simulator. This means the RL agents are tolerant of the weather change and occupancy/plug-load schedule change.

4.4.3 Effects of the Indoor Air Temperature Setpoint Strategy

The previous section finds that the control performance in the Perturbed1 and Perturbed2 simulators is worse than the training control performance in some experiment scenarios. In the Pittsburgh scenarios, the trained control policy even leads to negative HVAC energy savings in the Perturbed1 and Perturbed2 simulators. This is mainly because the two perturbed simulators have the PMV-based indoor air temperature setpoint, which is significantly different from the one in the training simulator.

Figure 4.16 shows the comparisons of the cooling setpoint in different simulators. It can be seen that, in the Pittsburgh scenarios and Beijing scenarios, the cooling setpoint in the perturbed simulators is significantly higher than that in the training simulator. Correspondingly, as shown in Figures 4.12 and 4.13, the energy efficiency performance in the perturbed simulators is significantly worse than that in the training simulator (with one exception in the Beijing-Heavyweight

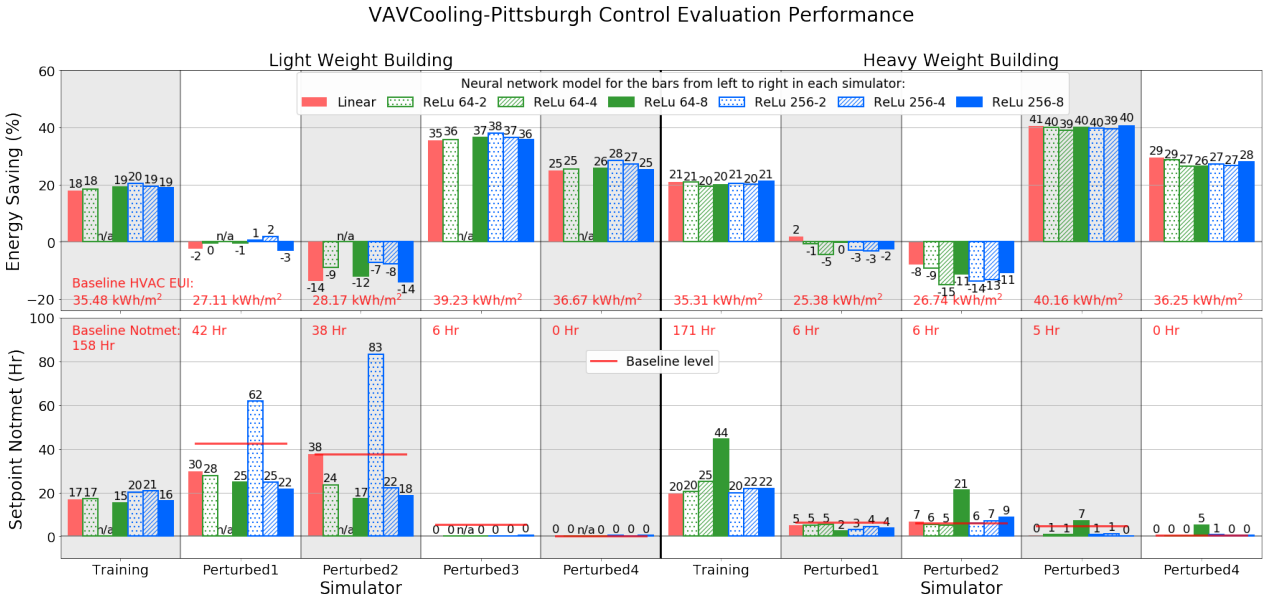

Figure 4.12: VAVCooling: Control Performance in Pittsburgh Climate (n/a means the reinforcement learning does not converge; the results of each neural network model are from the best-performing learning rate; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

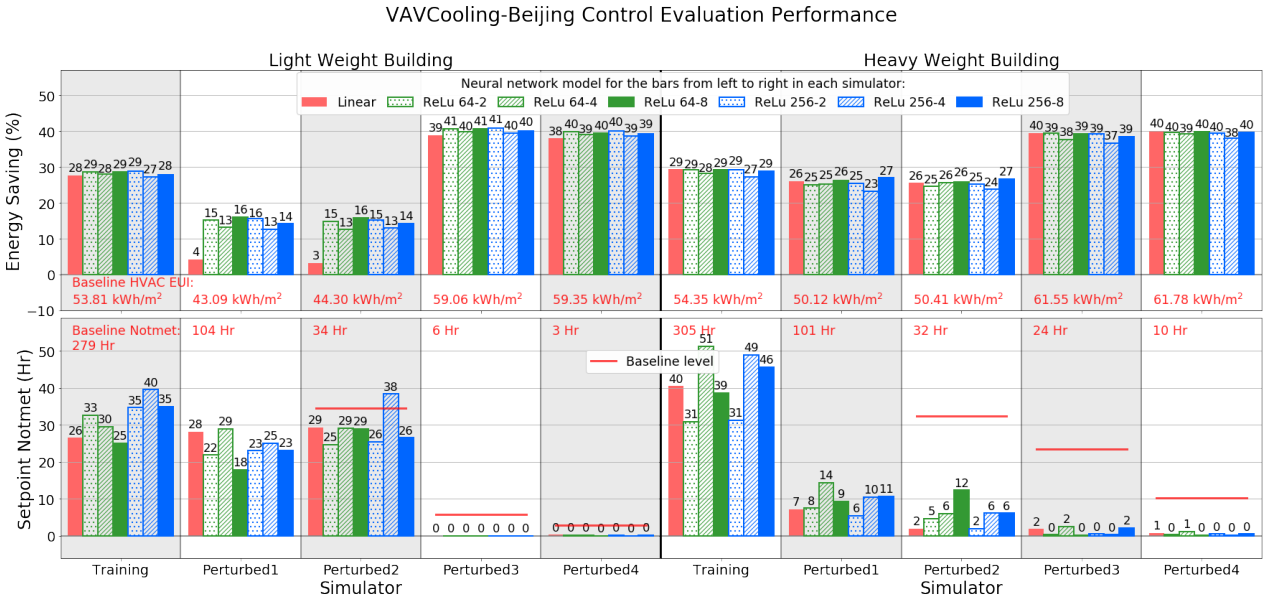

Figure 4.13: VAVCooling: Control Performance in Beijing Climate (the results of each neural network model are from the best-performing learning rate; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

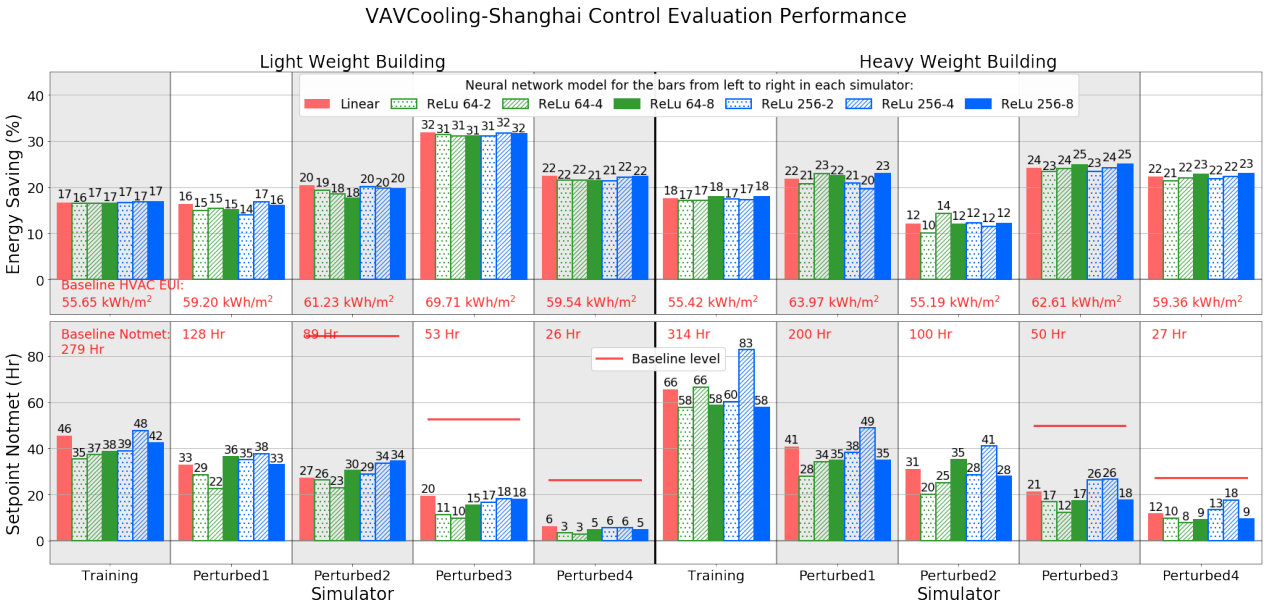

Figure 4.14: VAVCooling: Control Performance in Shanghai Climate (the results of each neural network model are from the best-performing learning rate; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

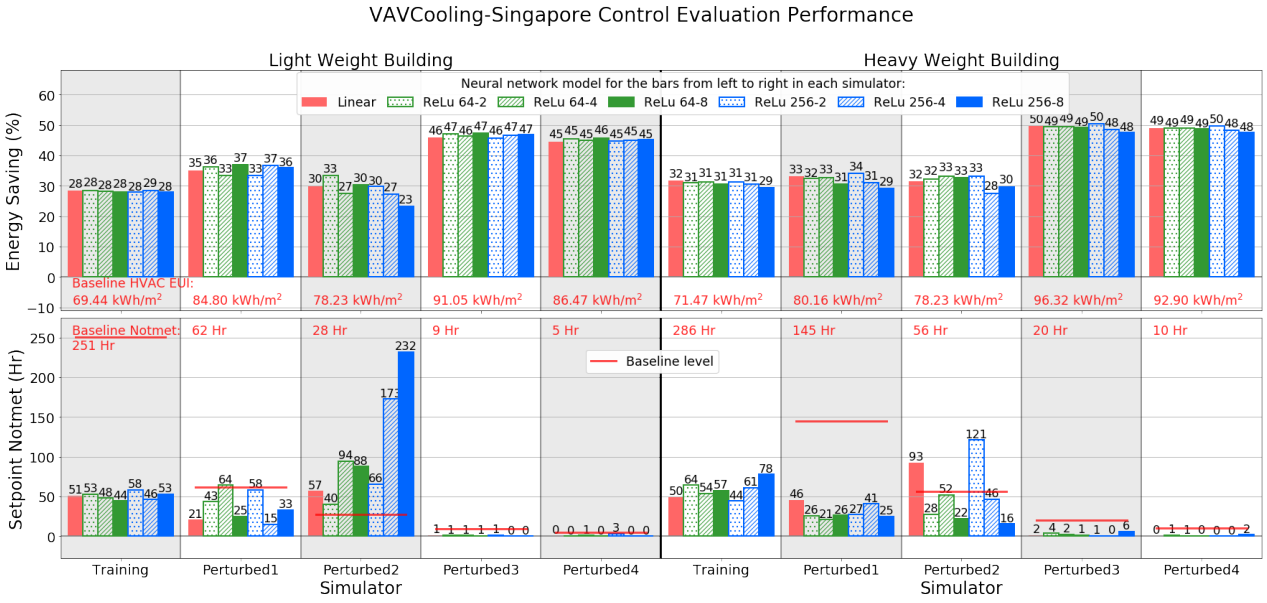

Figure 4.15: VAVCooling: Control Performance in Singapore Climate (the results of each neural network model are from the best-performing learning rate; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

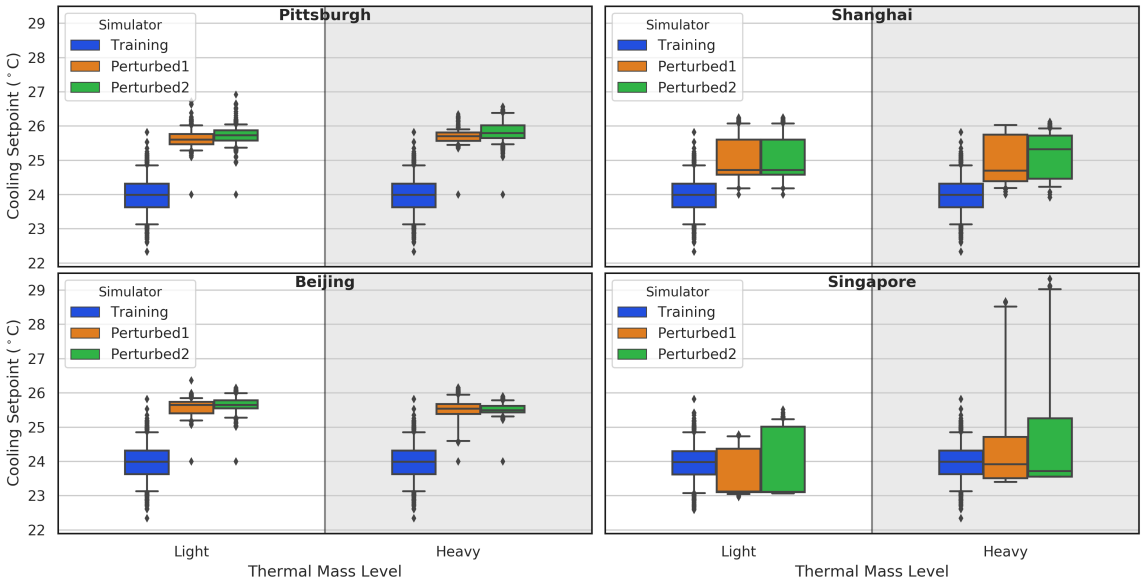

Figure 4.16: VAVCooling: Box-plots of the Average Cooling Setpoint in All Conference and Office Zones in Working Hours of the Original Training Simulator, Perturbed1 Simulator and Perturbed2 Simulator of the Selected Control Policies (the control policy of the ReLu64-2 model is used for the Beijing-Lightweight scenario, the control policies of the linear model are used for all the other scenarios)

scenario). For Shanghai climate, the perturbed cooling setpoint is higher than the training cooling setpoint, but the amount of difference is smaller than the Pittsburgh and Beijing scenarios. For Singapore climate, the median of the perturbed cooling setpoint is lower than the median of the training setpoint, which means the PMV-based strategy tends to generate low cooling setpoint values. Correspondingly, as shown in Figures 4.14 and 4.15, the Shanghai and Singapore scenarios have similar-to-the-training energy efficiency performance in the perturbed simulators. The results indicate that the trained control policies are more tolerant of the decreased cooling setpoint than the increased cooling setpoint.

To further analyze the effects of the indoor air temperature setpoint, a new RL agent is trained for the Pittsburgh-Lightweight scenario using a new training simulator. The new training simulator is exactly the same as the original one except that the indoor air temperature setpoint is PMV-based (same as the Perturbed1 and Perturbed2 simulators). Figure 4.17 shows the control performance comparison between the new training setting and the original setting. The best-performing control policy is selected from both settings. It can be seen that the control performance is significantly improved by using the new training simulator. The new trained RL agent has achieved positive energy-savings and better-than-baseline setpoint notmet time in the Perturbed1 and Perturbed2

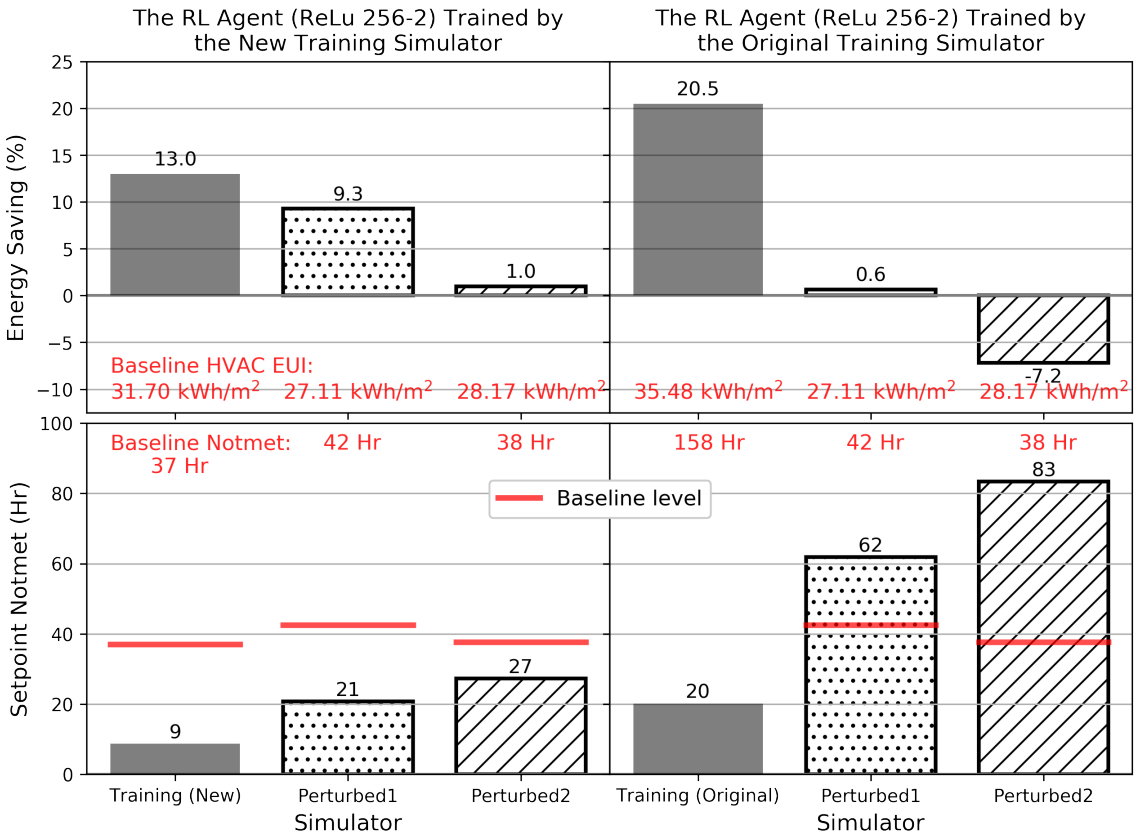

Figure 4.17: VAVCooling: Comparison of the Control Performance of the Best-performing RL Control Policies Trained in the New and Original Training Simulator for the Pittsburgh-Lightweight-Building Scenario

simulators. The improvements are attributed to the decreased difference between the training and perturbed cooling setpoint. As shown in Table 4.6, the KL divergence between the two cooling setpoint distributions has been significantly reduced. However, the control performance of the new control policy in the perturbed simulators is still worse than that in the new training simulator. This is because the PMV-based setpoint uses PMV to determine the setpoint value at each control time step, so the distribution of the setpoint is different for different conditions (such as different weather conditions). Thus, as shown in Figure 4.18 and Table 4.6, even though the new training and perturbed simulators use the same PMV-based setpoint strategy, the cooling setpoint distributions are still different.

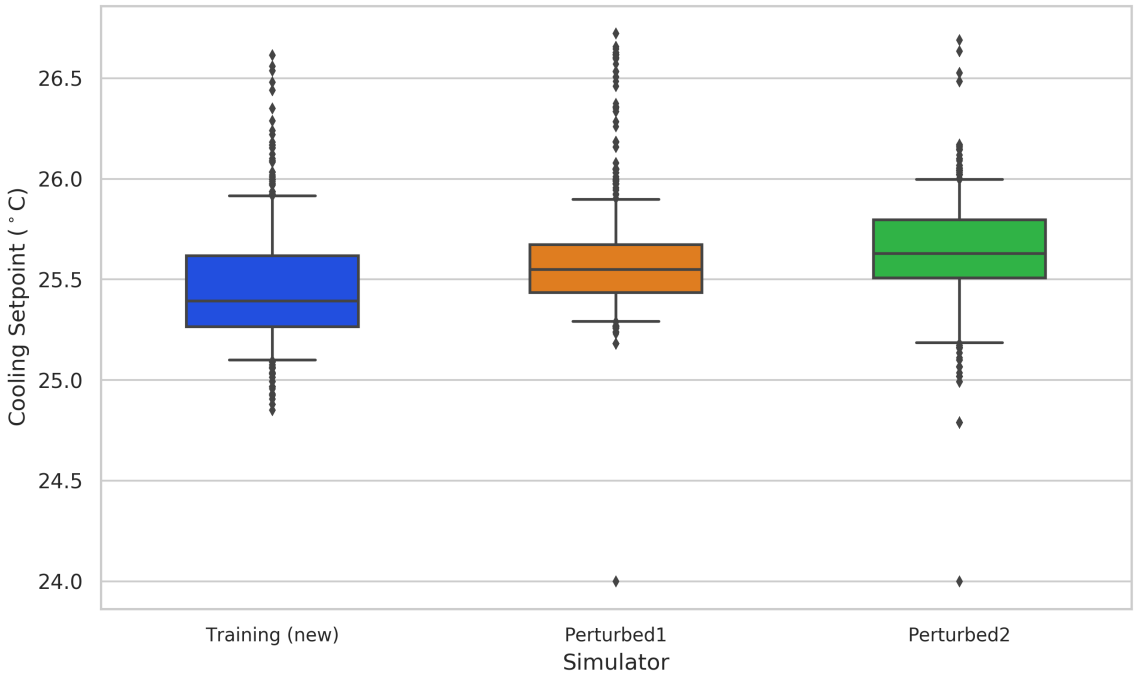

Figure 4.18: VAVCooling: Box-plots of the Average Cooling Setpoint of All Conference and Office Zones in Working Hours of the New Training Simulator, Perturbed1 Simulator and Perturbed2 Simulator of the Pittsburgh-Lightweight-Building Scenario using the Control Policies Trained by the New Training Simulator

Table 4.6: VAVCooling: Kullback–Leibler (KL) Divergence between the Training Cooling Setpoint and the Perturbed Cooling Setpoint of the Pittsburgh-Lightweight-Building Scenario in the New and Original Training Settings (all results are based on the best-performing control policies)

	Training vs. Perturbed1	**Training vs. Perturbed2**
Original Training	9.691	9.689
New Training	3.898	2.657

4.5 Summary and Discussion

This chapter presents the simulation experiments of VAVCooling, a variable-air-volume system serving 22 thermal zones for cooling. The system has complex dynamics because it is an integration of multiple HVAC components, including a heat pump, a variable-speed fan, a solid-desiccant dehumidification wheel, an air-to-air heat exchange wheel, an air-side economizer, multiple terminal reheat electric coils, etc. Also, the operation of all the HVAC components is highly correlative, which makes the control problem even more complicated. This chapter applies the proposed control framework to develop energy-efficient control strategies for this complex system. Eight experiment scenarios are considered including four different climates and two different thermal mass levels mixed together. The effects of the neural network model complexity and the learning rate are studied.

The state/action/reward are designed following the requirements of the control framework. The observation vector in the state includes the information of time, weather, operational state, and energy consumption. The action space is the discrete AHU supply air temperature setpoints. Based on the literature review, this control variable is the most influential variable for a VAV system. Different values of the setpoint significantly affect the operations of all HVAC components. The reward function is design as such that it is negatively proportional to the current-time-step HVAC energy consumption if the indoor air temperature setpoints are met. This reward function contains the minimum prior-knowledge for the HVAC system dynamics, so the exploration space of an RL agent is not constrained.

After the RL training, the convergence results are firstly presented in this chapter. It is found that the reinforcement learning agents with the shallow neural network models are easier to converge than those with the deep neural network models. Such results are aligned with the hypotheses of the thesis. It is also found that the small learning rates are obviously more favorable than the large learning rates for convergence. This is probably because a too aggressive learning rate causes a neural network to saturate easily, i.e., the neurons of a neural network output the same value for all different inputs. A neural network cannot "learn" anymore after it saturates.

The control performance of each trained RL policy is evaluated in the training simulator and four different perturbed simulators. The perturbed simulators are used to evaluate a trained control policy's versatility under perturbed HVAC operational conditions. For the training simulator evaluation, it is found that the RL control polices have achieved significant energy efficiency improvements compared the rule-based control for all the scenarios. The energy saving percentages are large than

20% and the setpoint notmet hours are significantly lower than the baseline. For the evaluation in the perturbed simulators, the control performance is not consistent. The trained control policies can well tolerate the perturbations in weather conditions and internal load schedules. The major problem is located in the indoor air temperature setpoint perturbations which are reflected in the Perturbed1 and Perturbed2 simulators. These two simulators have a different indoor air temperature setpoint strategy which is "PMV-based" (i.e., set the setpoint based on the simulated thermal comfort responses). For the Pittsburgh-Lightweight, Pittsburgh-Heavyweight and Beijing-Lightweight scenarios, the control performance in the Perturbed1 and Perturbed2 simulators is much worse than the training control performance. For the other scenarios, the Perturbed1 and Perturbed2 simulator control performance is comparable or even better than the training control performance.

Further analysis is conducted for the effects of the indoor air temperature setpoint on the RL control performance. It is found that, the trained control policies show more tolerance for reduced indoor air temperature setpoint than for increased indoor air temperature setpoint. An additional experiment is also conducted, in which a new training simulator is created with the PMV-based strategy for the indoor air temperature setpoint. The results show that this change in the training simulator can improve the control performance in the Perturbed1 and Perturbed2 simulators. However, since the PMV-based strategy generates different setpoint distributions for different conditions, the control performance in the Perturbed1 and Perturbed2 simulators still cannot match that in the new training simulator. The results from the perturbed simulators give two indications for the actual deployment of the control framework in VAV systems. Firstly, typical weather data can be used for the offline RL training since the trained control policies can well tolerate weather condition changes; secondly, training simulators must be calibrated to match the behaviors of actual systems, especially for indoor air temperature setpoint.

The results also show the effects of the neural network model complexity on the control performance. It is found that the different neural network models have not shown obvious effects on the control performance. More interestingly, the linear neural network model has achieved similar control performance with the other more complex nonlinear neural network models in most experiment scenarios (except the Beijing-Lightweight scenario). This does not support the thesis's hypotheses. However, the conclusion is derived from the limited number of experiments shown in this chapter. Its statistical significance is not tested, and the results cannot be generalized to other scenarios.

Chapter 5

Experiment 2: VAVHeating

This chapter presents the experiments related to VAVHeating. There are 6 scenarios related to this system, for two thermal mass levels and three different climates, as shown in Figure 5.1. Seven neural network models and six learning rates are tuned for each scenario.

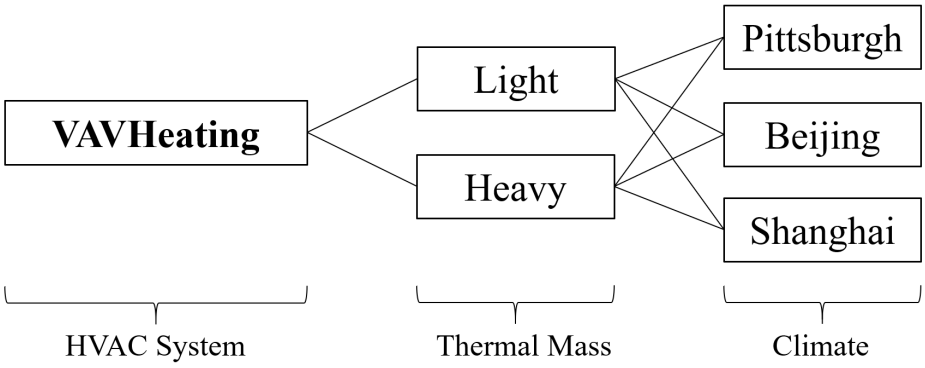

Figure 5.1: VAVHeating Experiment Scenarios

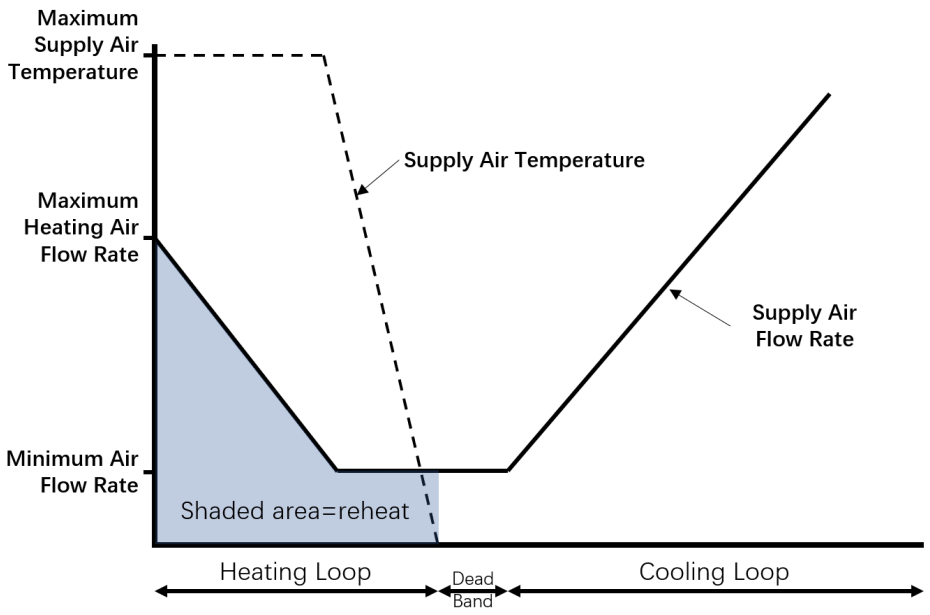

Figure 5.2: Terminal Air Flow Rate and Temperature Control Logic of VAV Systems with Terminal Reheat (re-generated based on (EnergyPlus, 2019))

5.1 HVAC System Description

5.1.1 System Layout, Thermal Zones and Envelopes

VAVHeating is a variable-air-volume (VAV) system with terminal reheat for heating. The system serves 22 zones. The system layout, thermal zones and envelopes are the same as VAVCooling (see sections 4.1.1 and 4.1.2).

The overall operation strategy of VAVHeating is slightly different from VAVCooling. In VAV-Cooling, the supply air flow rate of a zone is proportional to the zone cooling load. In VAVHeating, the zone supply air flow rate is firstly kept at the minimum and a terminal electric heater "reheats" the zone supply air to meet the zone heating load; if the temperature of the reheated supply air reaches its high limit (35 °C in this study) but the zone heating load cannot be met, the zone supply air flow rate starts to increase. This strategy is shown in Figure 5.2.

5.1.2 Target Control Variable and Baseline Control Strategy

The system's air-handling-unit supply air temperature setpoint is the target control variable of VAVHeating. It is the same as the experiments of VAVCooling. This setpoint can affect the operations of almost all the components of VAVHeating, such as fan, heat pump, terminal electric heaters, etc. For example, if this setpoint is low (the air-handling-unit supplies cold air), the power of the heat pump will be reduced, but the power of the fan and the terminal electric heaters will be increased; if this setpoint is high (the air-handling-unit supplies warm air), the power of the fan and the terminal electric heaters will be decreased, but more power will be consumed by the heat pump.

The baseline control strategy for this setpoint (T_{sa}) is called "coldest", which is a built-in function in EnergyPlus, as shown below:

$$T_{sa,t} = \left[\max_{i \in \text{all zones}} T_{ia,i,t-1} + \frac{\dot{Q}_{heating,i,t-1}}{C_{p,air} \dot{m}_{max,i}} \right]_{T_{sa}}^{T_{sa}}, \tag{5.1}$$

where subscript t is one control time step, i is one zone, T_{ia} is zone indoor air temperature, $\dot{Q}_{heating}$ is zone heating load, $C_{p,air}$ is the specific heat capacity of air, $\dot{m}_{max}$ is zone maximum supply air mass flow rate, T_{sa} and T_{sa} are the high and low limit of the supply air temperature setpoint (in this case, 30°C and 18°C). Intuitively, the baseline control strategy provides the lowest possible air-handling-unit supply air temperature setpoint that could meet the heating loads of all zones.

Whole Building Energy Model

Same as VAVCooling, the whole building energy model of VAVHeating is created in EnergyPlus version 8.3. The geometry 3D rendering of the model is shown in the previous section in Figure 4.4. The capacities of the components in the system are autosized by EnergyPlus using the design conditions of each climate. The length of one simulation episode and simulation time step are shown in Table 5.1.

Table 5.1: Basic Simulation Settings of the Whole Building Energy Models for the VAVHeating Scenarios

Climate	**Thermal Mass**	**Simulation Period**	**Simulation Time Step**
Pittsburgh	Light / Heavy	Jan 1st-Mar 31st	10-min
Beijing	Light / Heavy	Jan 1st-Mar 31st	10-min
Shanghai	Light / Heavy	Jan 1st-Mar 31st	10-min

5.2 Training and Perturbed Simulators

The configurations of the training and perturbed simulators are the same as that in the VAVCooling scenarios. The four perturbed simulators are different from the training simulator in weather conditions, occupancy schedules, plug-load schedules and indoor air temperature setpoint schedules. The detailed configurations are shown in Table 5.2.

Table 5.2: Comparison of the Training and Perturbed Simulators for VAVHeating Scenarios

	Training	**Perturbed**			
		1	**2**	**3**	**4**
Weather	TMY3 for Pittsburgh, IWEC for other locations	AMY 2017	TMY/IWEC with additive white Gaussian noise	AMY 2017	TMY3/IWEC with additive white Gaussian noise
Occupancy Schedule	Deterministic with additive white Gaussian noise	Stochastic			
Plug-load Schedule	Deterministic with additive white Gaussian noise	Stochastic			
IAT Setpoint	Deterministic with additive white Gaussian noise	PMV-based		Deterministic	

5.3 Reinforcement Learning Setup

5.3.1 State Design

The state design is the same as that in VAVCooling scenarios, as shown in Table 5.3. The items are normalized using min-max normalization (Equation (2.20)).

The length of the history in the state is determined based on the method in section 2.4.1. The control time step length is 10-min. The relationships between the time interval n and the distance correlation $dcor_n$ are shown in Figure 5.3. It can be seen that the $dcor_n$ decays slower in the heavyweight scenarios, and the $dcor_n$ pattern is repeated after $n = 144$ = 1-day. These results are as expected. Based on algorithm 1, the length of the history in the state is the largest time interval that makes the $dcor_n$ larger or equal to 0.5, as summarized in Table 5.4.

Table 5.3: Observation Vector in the State for VAVHeating

No.	Item
1	Is weekday or not
2	Hour of the day
3	Outdoor air temperature (°C)
4	Outdoor air relative humidity (%)
5	Diffuse solar radiation (W/m^2)
6	Direct solar radiation (W/m^2)
7-28	Zone air temperature (of 22 zones, °C)
29-50	Zone cooling setpoint temperature (of 22 zones, °C)
51-72	Zone heating setpoint temperature (of 22 zones, °C)
73	Total HVAC Electric Demand (kW)

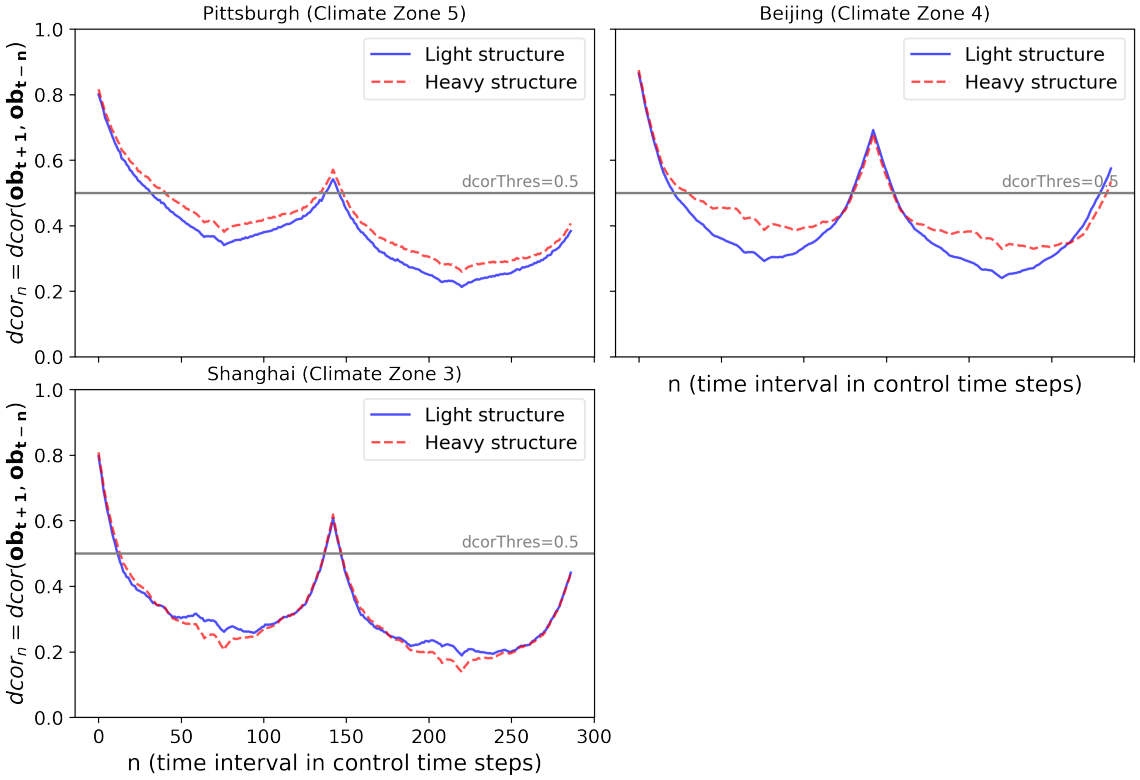

Figure 5.3: Relationship Between Time Interval n and $dcor_n$ (specified in Equation (2.23)) for All VAVHeating Scenarios

Table 5.4: Length of the History in the State for VAVHeating Scenarios

Climate	**Thermal Mass**	**Length of History (control time steps)**
Pittsburgh	Light	32 (5.3 hr)
	Heavy	41 (6.8 hr)
Beijing	Light	22 (3.7 hr)
	Heavy	30 (5.0 hr)
Shanghai	Light	12 (2.0 hr)
	Heavy	13 (2.2 hr)

5.3.2 Action Design

The discrete action space for the air-handling-unit supply air temperature setpoint is:

$$A_{vavheating} = \{18°C, 18.5°C, ..., 29.5°C, 30°C\} \tag{5.2}$$

5.3.3 Reward Design

The reward function is the same as the VAVCooling scenarios, as shown in Equation (4.3). The reward function penalizes large HVAC energy consumption and indoor air temperature setpoint notmet.

5.3.4 Hyperparameters

The choice of the hyperparameters is the same as the VAVCooling scenarios, as shown in Table 4.5. Seven neural network models and six learning rates will be studied for each experiment scenario, as shown in Figure 3.5.

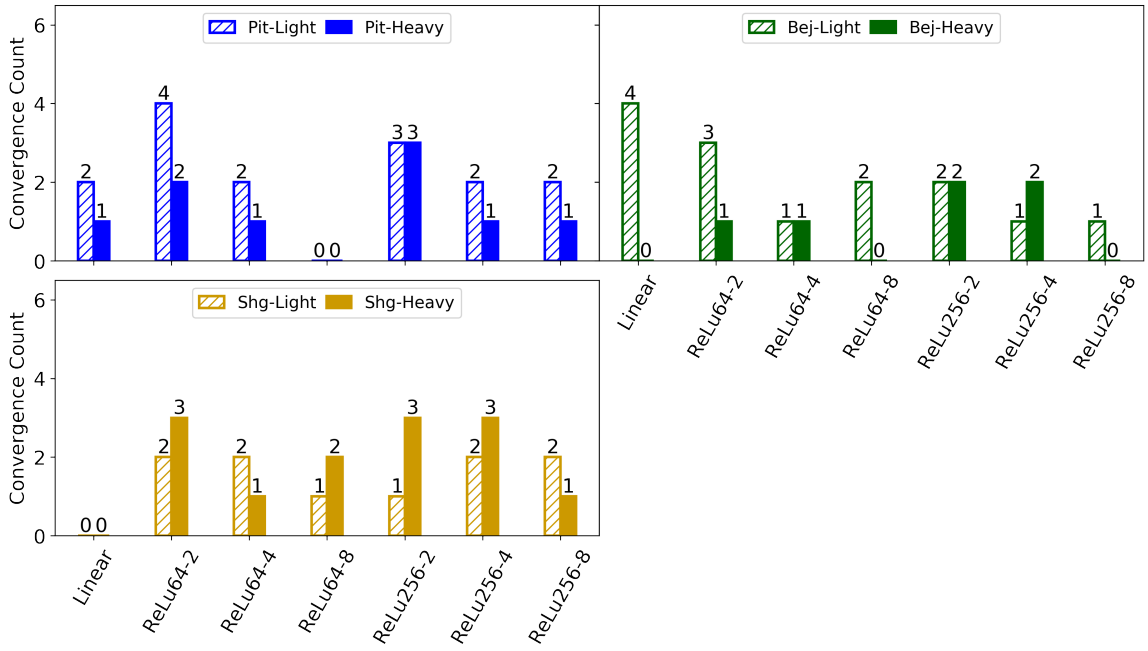

Figure 5.4: VAVHeating: Convergence Robustness to Learning Rate (the count of convergence out of the six learning rates) vs. Neural Network Models

5.4 Results

5.4.1 Convergence Results

Six learning rates, from 1e-3 to 5e-6, are tuned for each experiment scenario/neural network model combination. Figure 5.4 shows the count of convergence out of the six learning rates for all the combinations. A larger convergence count means the corresponding neural network model is more robust to different learning rates. It is found that the width of a neural network model does not have obvious effects on the convergence robustness, but the depth does. In general, a shallow neural network has larger convergence count than a deep neural network. However, exceptions exist, such as the ReLu256-x neural network models in the Shanghai-Light scenario. Besides, the convergence count of the linear model varies across different scenarios. For example, in the Beijing-Light scenario, the linear model has the largest convergence count; but in the Shanghai-Light and Shanghai-Heavy scenarios, its convergence count is zero.

Figure 5.5 shows which learning rate is the most favorable for the reinforcement learning convergence. It is clear that, in general, a smaller learning rate leads to a larger convergence count.

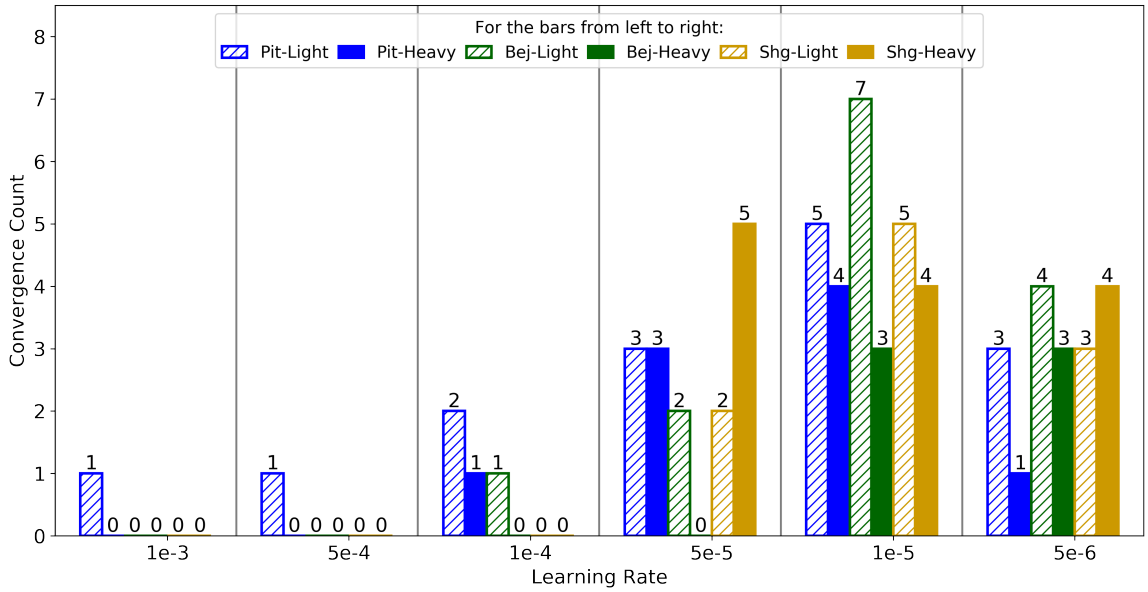

Figure 5.5: VAVHeating: Convergence Count of the Seven Neural Network Models vs. the Six Learning Rates

The learning rate 1e-5 has the largest convergence count for all the scenarios. This is different from VAVCooling where 5e-6 is the most favorable learning rate for convergence.

Figures 5.6, 5.7 and 5.8 show the training evaluation histories of the seven neural network models at the learning rate 1e-5. In the figures, it can be seen that there is no obvious relationship between the neural network model complexity and the profile of the training evaluation history. Besides, the linear model has fluctuated or diverged training evaluation histories in all the scenarios. This is different from VAVCooling where the shallow neural network models and the linear model have more smooth training evaluation histories than the deep neural network models.

5.4.2 Control Performance

The control performance of the reinforcement learning agents is evaluated by the percentage saving of the total HVAC electricity consumption (E_{saving}) and the setpoint notmet time (Ti_{nmt}). The equations to calculate E_{saving} and Ti_{nmt} are shown in Equation (4.4) and (4.5).

The control performance is shown in Figures 5.9 (Pittsburgh climate), 5.10 (Beijing climate) and 5.11 (Shanghai climate). In the figures it can be seen that:

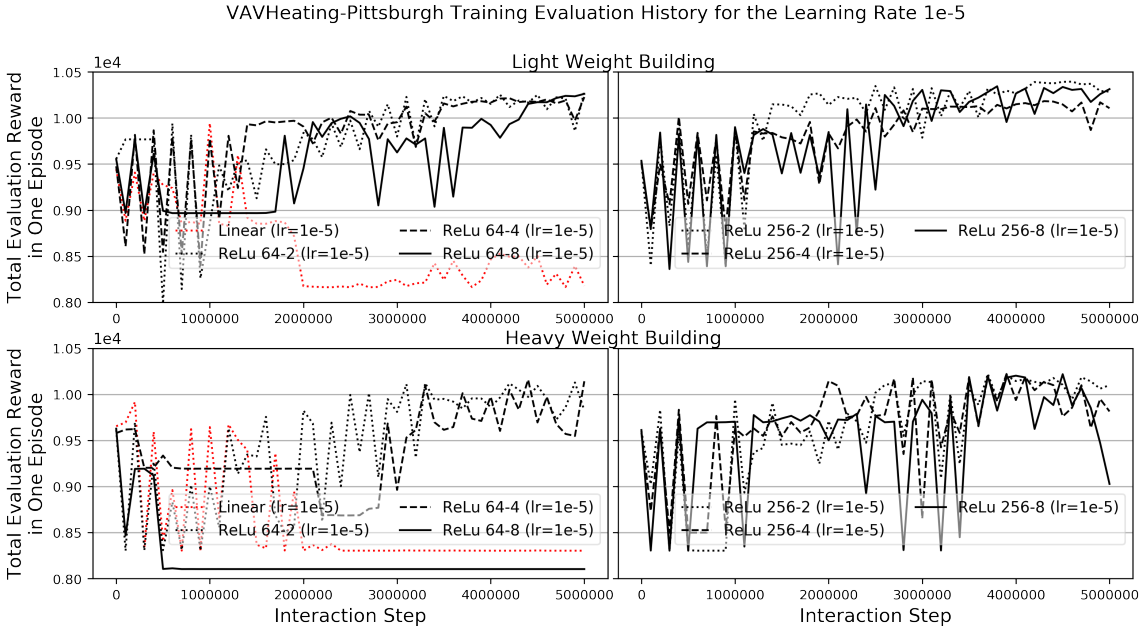

Figure 5.6: VAVHeating: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Pittsburgh Climate)

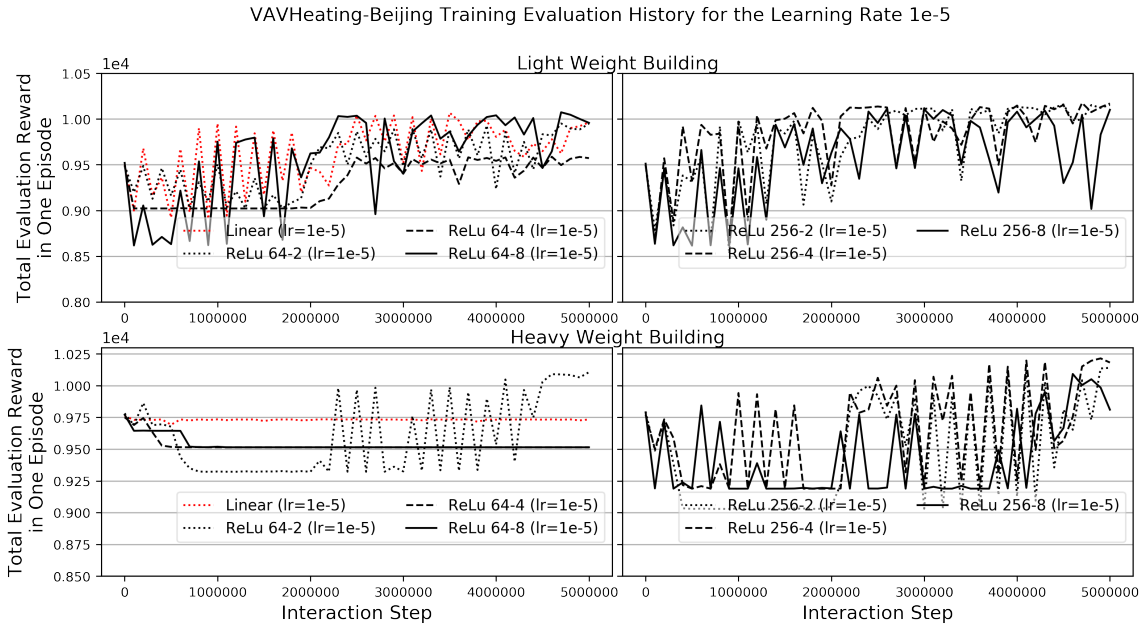

Figure 5.7: VAVHeating: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Beijing Climate)

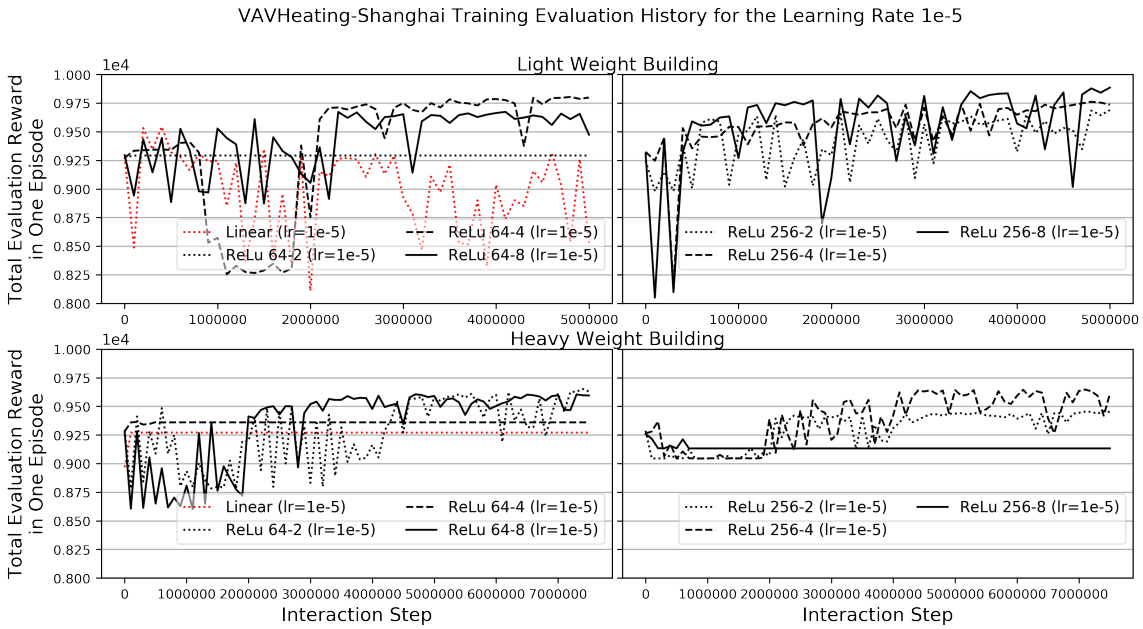

Figure 5.8: VAVHeating: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Shanghai Climate)

- Compared to the VAVCooling scenarios, the reinforcement learning control can still achieve less or similar setpoint notmet time than the baseline control, but the magnitude of HVAC energy savings is reduced. This is as expected because cooling usually has a larger potential for energy saving than heating.

- Different neural network models lead to different control performance. However, there is no obvious relationship between the control performance and the width or depth of a neural network model. The linear model has poor control performance in most experiment scenarios, except the Beijing-Lightweight-Building scenario (Figure 5.10). The ReLu 64-2 model has the most stable performance across all the scenarios. This is different from VAVCooling where all the neural network models, including the linear one, have similar control performance.

- There is no significant difference between the control performance in the lightweight-building scenarios and heavyweight-building scenarios. This indicates that the thermal mass level has little influence on the control performance of this system.

- The control performance in the Perturbed1 and Perturbed2 simulators is worse than that in the Perturbed3 and Perturbed4 simulators. This result is similar to the VAVCooling scenarios. This is because the reinforcement learning agents cannot tolerate the changed indoor air temperature setpoint, which has different distributions from the setpoint in the training

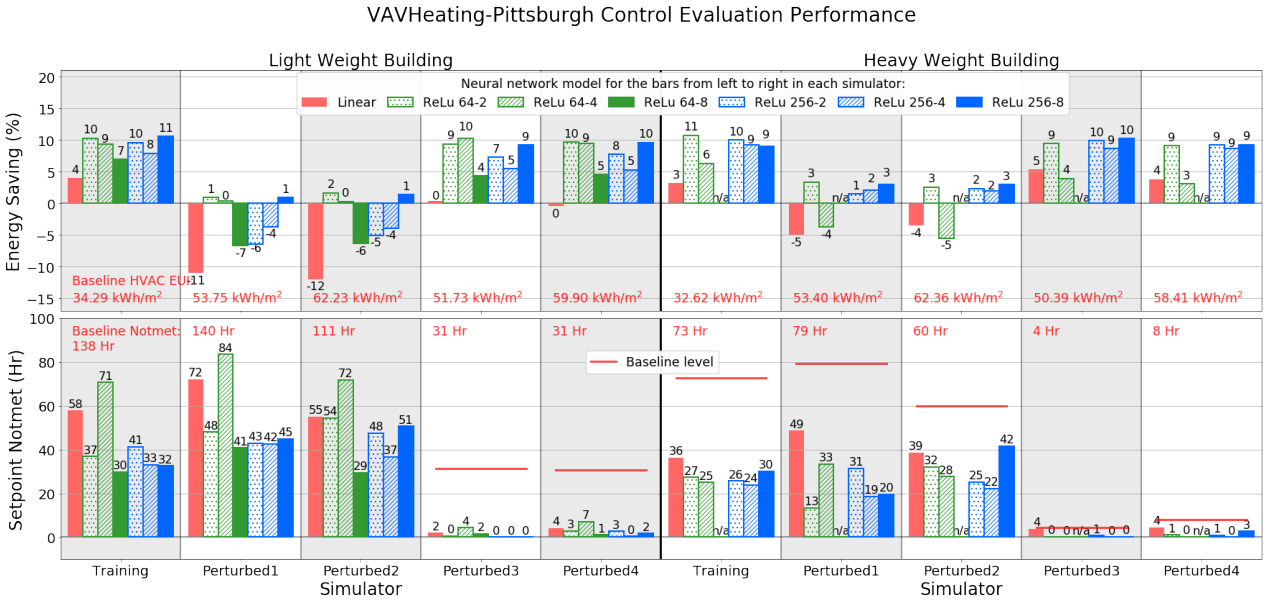

Figure 5.9: VAVHeating: Control Performance in Pittsburgh Climate (the results of each neural network model are from the best-performing learning rate; n/a means none of the learning rates lead to convergence; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

simulator.

- The control performance in the Perturbed3 and Perturbed4 simulators is similar and is comparable to that in the training simulator. This indicates that the RL agents are tolerant of the changes in weather conditions and occupancy/plug-load schedules.

5.4.3 Effects of the Indoor Air Temperature Setpoint Strategy

The previous section finds that the control performance in the Perturbed1 and Perturbed2 simulators is worse than that in the Perturbed3 and Perturbed4 simulators. This is because the Perturbed1 and Perturbed2 simulators use the PMV-based indoor air temperature setpoint, which has different distributions from the one in the training simulator. Figure 5.12 shows the box-plots of the heating setpoint in different simulators. It can be seen that the perturbed heating setpoint is significantly higher than the training heating setpoint in all the scenarios.

This section creates a new training simulator that has the same PMV-based setpoint strategy as the Perturbed1 and Perturbed2 simulators. A new RL agent is trained using the new training simulator for the Pittsburgh-Lightweight-Building scenario. Figure 5.13 shows the control perfor-

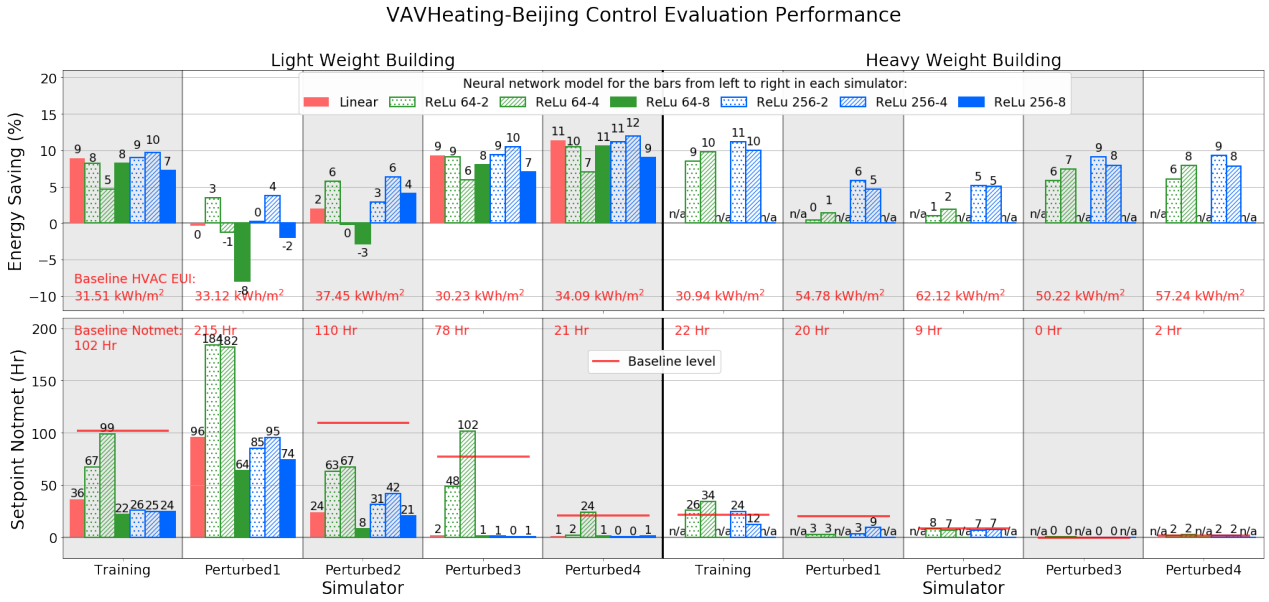

Figure 5.10: VAVHeating: Control Performance in Beijing Climate (the results of each neural network model are from the best-performing learning rate; n/a means none of the learning rates lead to convergence; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

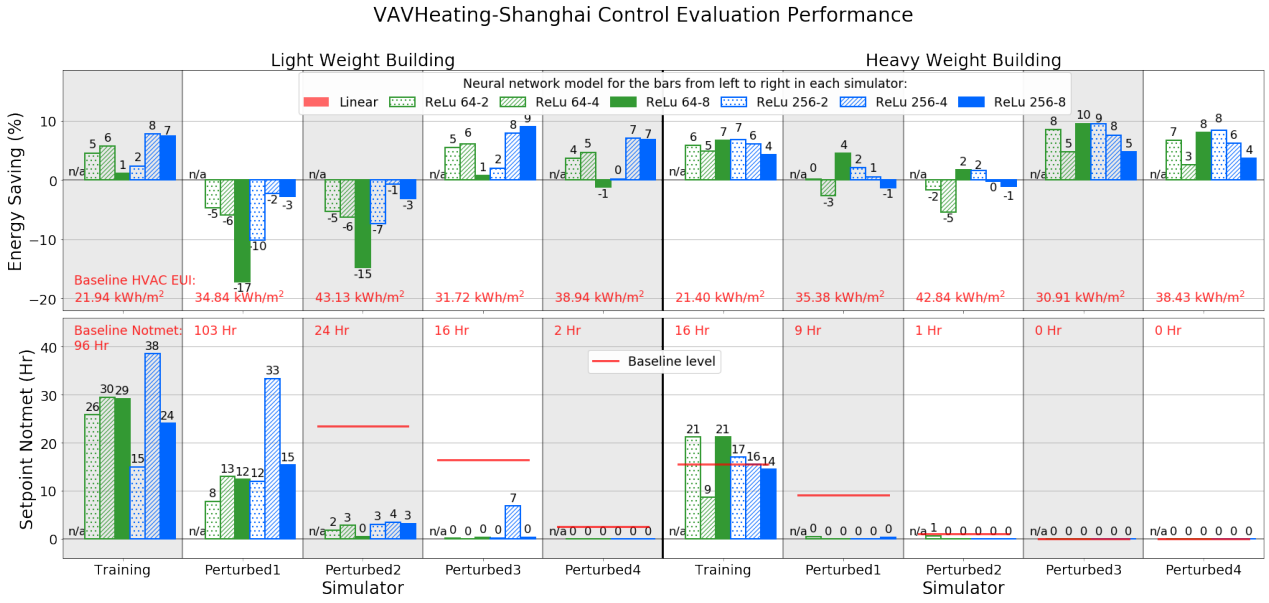

Figure 5.11: VAVHeating: Control Performance in Shanghai Climate (the results of each neural network model are from the best-performing learning rate; n/a means none of the learning rates lead to convergence; baseline HVAC EUI means the total HVAC electricity consumption per building floor area using the baseline control strategy)

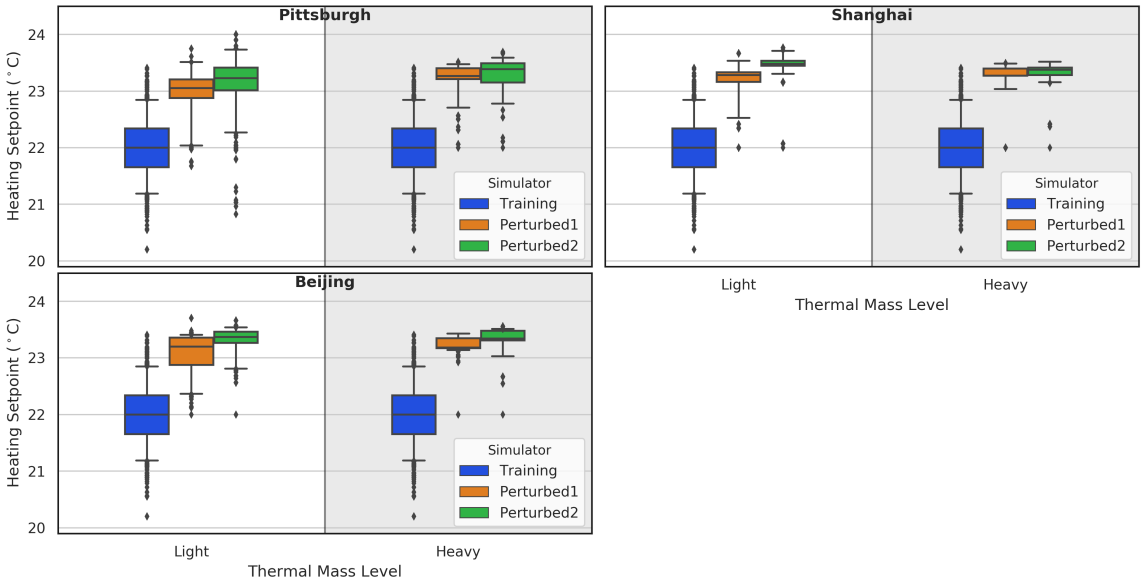

Figure 5.12: VAVHeating: Box-plots of the Average Heating Setpoint of All Conference and Office Zones in Working Hours of the Original Training Simulator, Perturbed1 Simulator and Perturbed2 Simulator of the Selected Control Policy (Pittsburgh-Light: ReLu256-8, Pittsburgh-Heavy: ReLu64-2, Beijing-Light: ReLu256-4, Beijing-Heavy: ReLu256-2, Shanghai-Light: ReLu256-8, Shanghai-Heavy: ReLu256-2)

mance comparison of the best-performing control policies using the original training simulator and the new training simulator. It can be seen that, compared to the control policy from the original training simulator, the new control policy leads to the significantly reduced setpoint notmet time but slightly increased energy consumption. Table 5.5 shows that the KL divergence between the training and perturbed heating setpoint is slightly reduced by the new training simulator. However, the new control policy still has worse-than-training control performance in the perturbed simulators. This is because, as shown in Figure 5.14 and Table 5.5, even though both new training and the perturbed simulators use the same PMV-based setpoint strategy, the heating setpoint still has different distributions in different simulators. This indicates that the new control policy still cannot tolerate the perturbations in the indoor air temperature setpoint.

Table 5.5: VAVHeating: Kullback–Leibler (KL) Divergence between the Training Heating Setpoint and the Perturbed Heating Setpoint of the Pittsburgh-Lightweight-Building Scenario in the New and Original Training Settings (all results are based on the best-performing control policies)

	Training vs. Perturbed1	**Training vs. Perturbed2**
Original Training	8.312	8.608
New Training	7.933	7.089

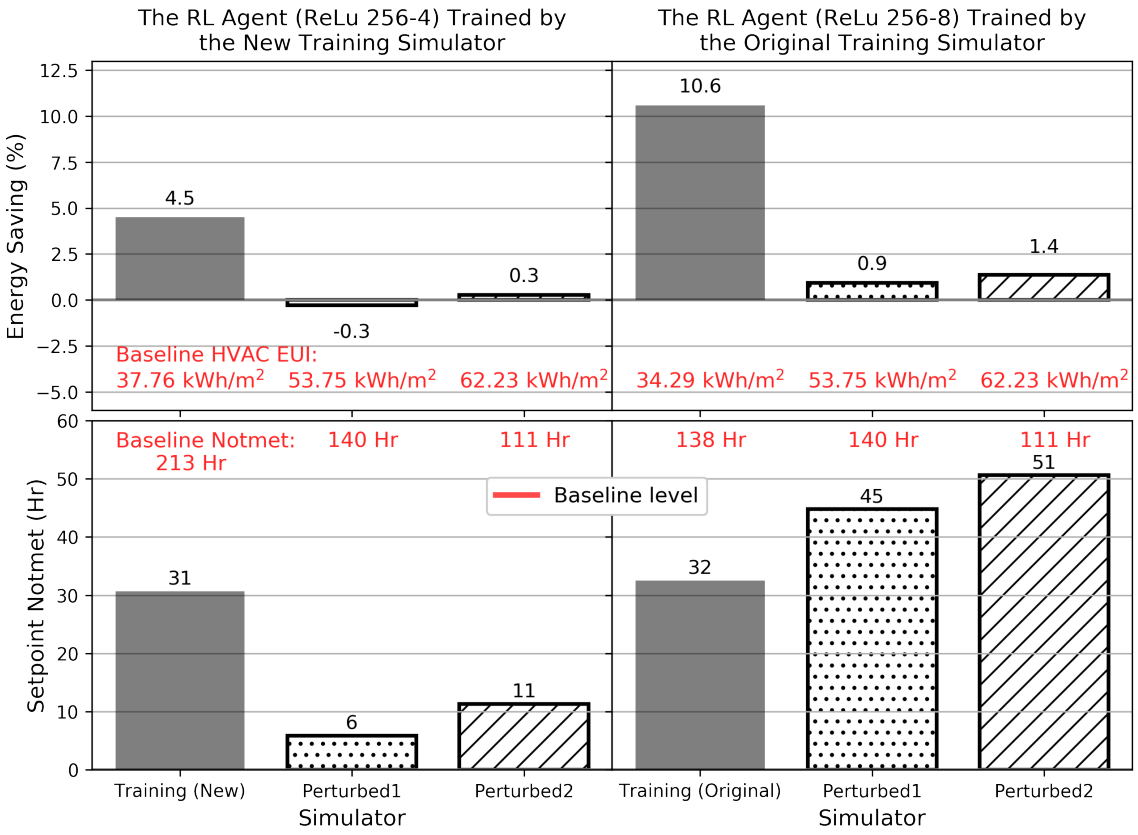

Figure 5.13: VAVHeating: Comparison of the Control Performance of the Best-performing RL Control Policies Trained Using the New and Old Training Simulator for the Pittsburgh-Lightweight-Building Scenario

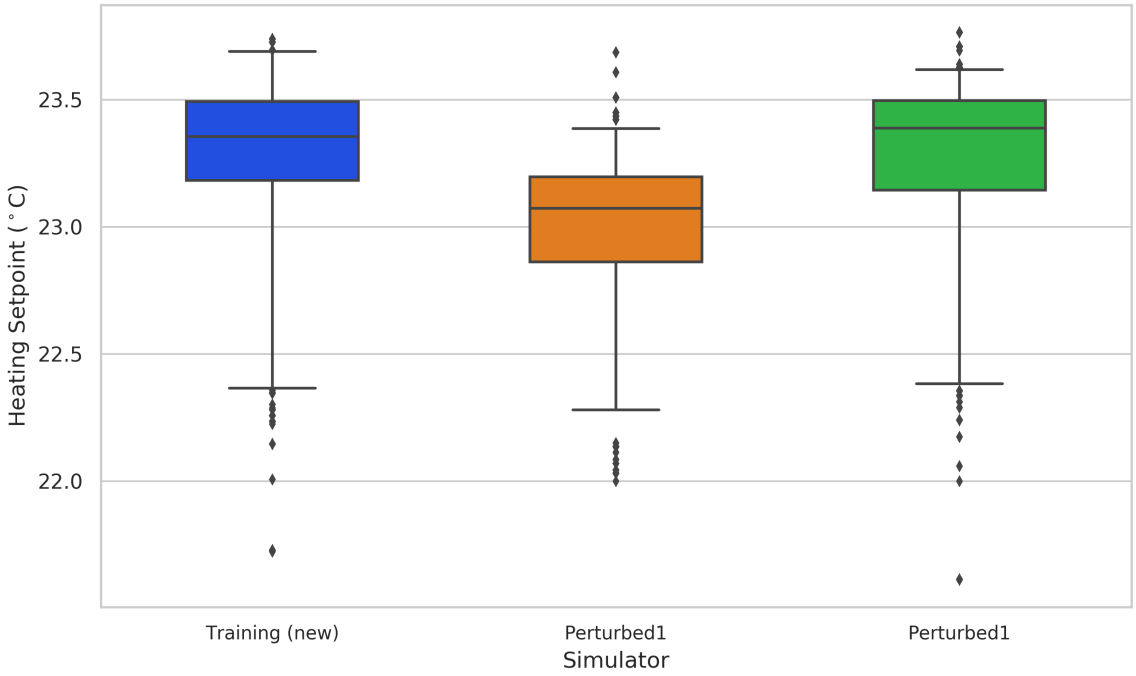

Figure 5.14: VAVHeating: Box-plots of the Average Heating Setpoint of All Conference and Office Zones in Working Hours of the New Training Simulator, Perturbed1 Simulator and Perturbed2 Simulator of the Pittsburgh-Lightweight-Building Scenario using the Control Policy Trained by the New Training Simulator

5.5 Summary and Discussion

This chapter presents the experiments related to VAVHeating, a variable air volume system in heating season. The configuration of VAVHeating is the same as that in VAVCooling, except the simulation period is in heating seasons. The operation strategy and the resulting dynamics of VAVHeating are dramatically different from VAVCooling. Thus, the experiments are separately presented in two chapters. The experiment scenarios for VAVHeating include three climates (Pittsburgh, Beijing, and Shanghai) and two thermal mass levels. For each experiment scenario, seven neural network models and six learning rates are tuned. Since VAVHeating has the same configuration with VAVCooling, the state/action/reward design is also similar.

The convergence results are firstly presented after the RL training. In general, the shallow neural network models are more robust to the different learning rates in terms of convergence. However, the linear model has the poor convergence performance (and also the poor control performance), which is different from the results in VAVCooling. This may be caused by the insufficient representational capacity of the linear model. The results slightly deviate from the hypotheses of this thesis, which states that reinforcement learning with a simple neural network model is easier to converge than a complex neural network model. It is also found that the small learning rates have a larger convergence count than the large learning rates. Out of the six tuned learning rate, 1e-5 has the largest convergence count. This is different from the VAVCooling experiments where 5e-6 is the best choice. This indicates that for different experiment scenarios, the most favorable learning rate is different. Hence, the learning rate needs to be tuned for different scenarios.

The control performance evaluation settings are the same as the VAVCooling experiments. The trained control policies are evaluated in the training simulator and four different perturbed simulators. For the control performance in the training simulator, the magnitude of the HVAC energy savings is less than that in the VAVCooling scenarios. This is as expected since cooling usually has a larger energy efficiency improvement potential than heating. Nevertheless, the trained control polices can still achieve around 10% energy savings and better-than-baseline setpoint notmet time in all the experiment scenarios. For the control performance in the perturbed simulators, the general trend of the results is also similar to VAVCooling. The trained control policies are tolerant of the variations in weather conditions and occupancy/plug-load schedules, but are not tolerant of the PMV-based indoor air temperature setpoints in the Perturbed1 and Perturbed2 simulators. An additional experiment is also conducted with a new training simulator with the PMV-based setpoint

strategy. The new trained control policy has improved control performance compared to the original one, but the control performance in the Pertrubed1 and Perturbed2 simulators is still worse than that in the new training simulator. This is because the dynamic PMV-based strategy delivers different distributions of indoor air temperature setpoint for different conditions. The results again indicate that, for the actual deployment of the control framework in a VAV system, the training simulator should be calibrated especially for its indoor air temperature setpoint schedule.

The results for the effects of the neural network model complexity on the control performance are different from the VAVCooling experiments. In the VAVCooling experiments, the different neural network models generate similar control performance for the same experiment scenario. In the VAVHeating experiments, the different neural network models deliver dramatically different control performance for the same experiment scenario. Besides, there is still no obvious relationship between the neural network model complexity and the control performance, and the best-performing neural network model architecture is different in different experiment scenarios. An interesting finding is that the linear model has much worse control performance than the other more complex neural network models in almost all the experiment scenarios (except the Beijing-Lightweight scenario). This may be because the linear model is insufficient to solve the control problems of VAVHeating. This is different from the VAVCooling experiments, where the linear model has achieved similar control performance with the other neural network models for almost all the experiment scenarios. In general, the results indicate that the neural network model architecture is an important hyperparameter that needs to be tuned for different scenarios. Also, the results do not support the hypothesis stating that reinforcement learning with a complex neural network model can deliver better control performance than a simple neural network model. However, it should be noted that the conclusions are derived from the limited number of experiments shown in this chapter, and cannot be generalized for other scenarios.

Chapter 6

Experiment 3: RadiantHeating

This chapter presents the experiments related to RadiantHeating. There are 6 scenarios for three different climates and two different thermal mass levels, as shown in Figure 6.1. Seven different neural network models and six learning rates are tuned for each scenario.

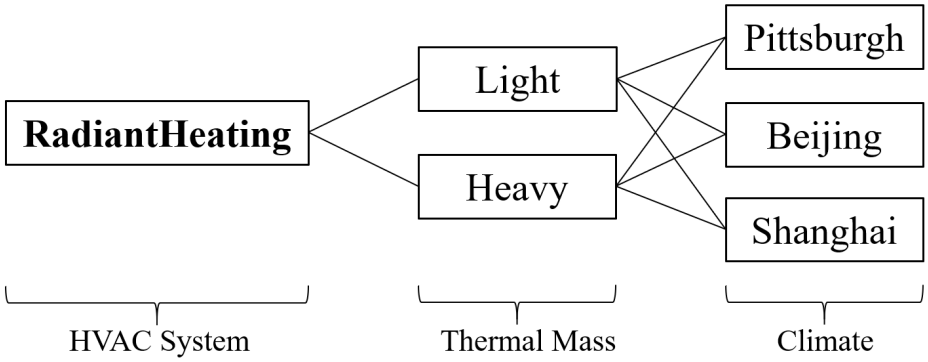

Figure 6.1: RadiantHeating Experiment Scenarios

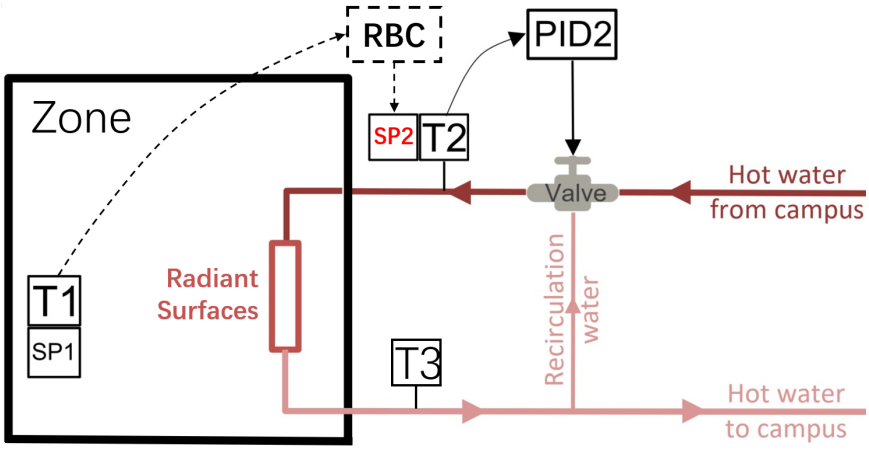

Figure 6.2: System Layout and Control Principles of RadiantHeating

6.1 Heating System Description

6.1.1 System Layout

RadiantHeating is a constant-water-flow radiant heating system. It has a relatively simpler layout compared to VAV systems. As shown in Figure 6.2, the system consists of a hot water source ("hot water from campus"), a three-way valve, a constant-speed water pump (not shown in the figure), radiant surfaces and a recirculation pipe.

During its operation, the hot water source supplies high-temperature water at a constant temperature, and the three-way valve regulates the mixture ratio between the recirculation water and the hot water from campus to adjust the system supply water temperature (T2 in the figure). A PID controller (PID2 in the figure) determines the mixture ratio of the three-way valve based on the error between the system supply water temperature (T2) and its setpoint (SP2). Since the system has a constant supply water flow rate, SP2 is used to change the amount of heating power supplied by the system.

6.1.2 Mullion Radiant Surface

Even though RadiantSystem has a simple layout and operation strategy, its thermal and energy behaviors are complicated. This is attributed to the design of the radiant surface. The top view of the radiant surface is shown in Figure 6.3. The radiant surface is named "mullion radiant surface"

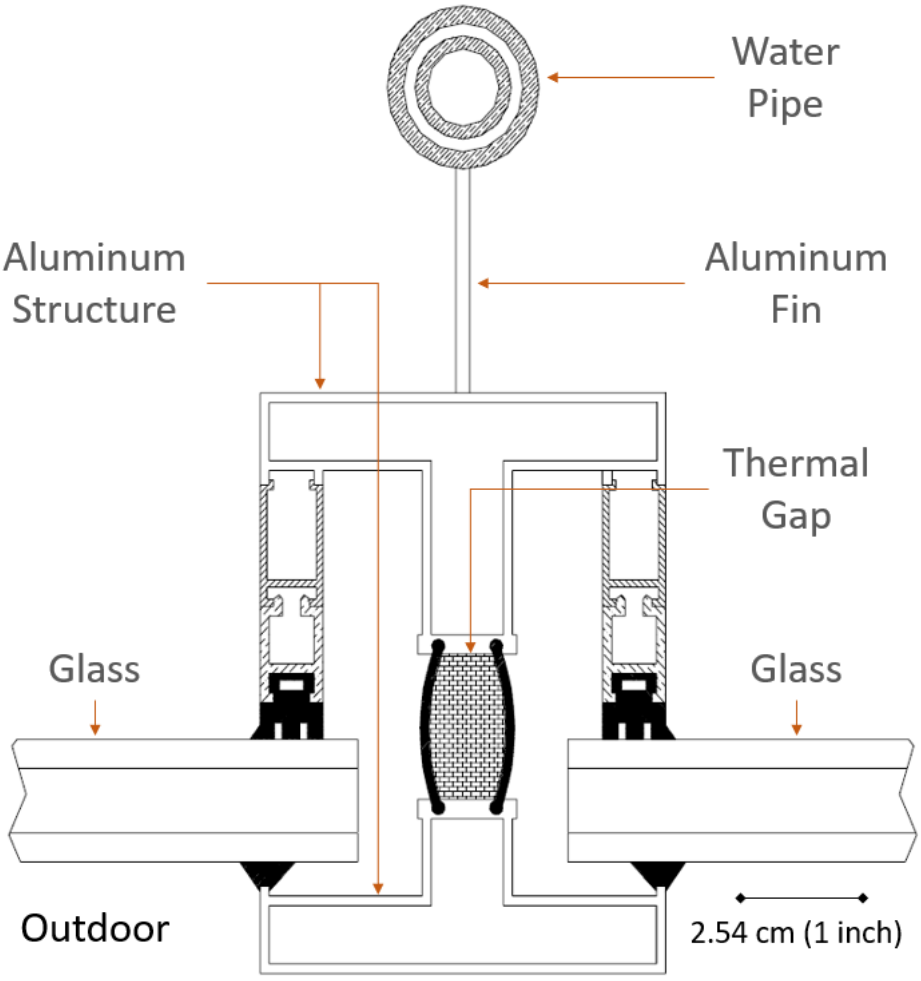

Figure 6.3: Top View of the Mullion Radiant Surface (Gong and Claridge, 2006)

because it uses window mullions as the radiant surface. During heating, a proportion of heat of the hot water in the water pipe is transferred to the aluminum fin and aluminum structures through heat conduction, and then the heat is radiated to the zone objects or lost to the outdoor environment. Hence, the system has a slow thermal response due to the thermal mass of the metal structures and the radiant heating thermodynamics.

6.1.3 Thermal Zones and Envelopes

The system serves a one-level building with 6 conditioned thermal zones, as shown in Figure 6.4. However, the indoor environmental conditions of the 6 zones are averaged for control purposes. As a result, from the control perspective, the system serves a single-zone building.

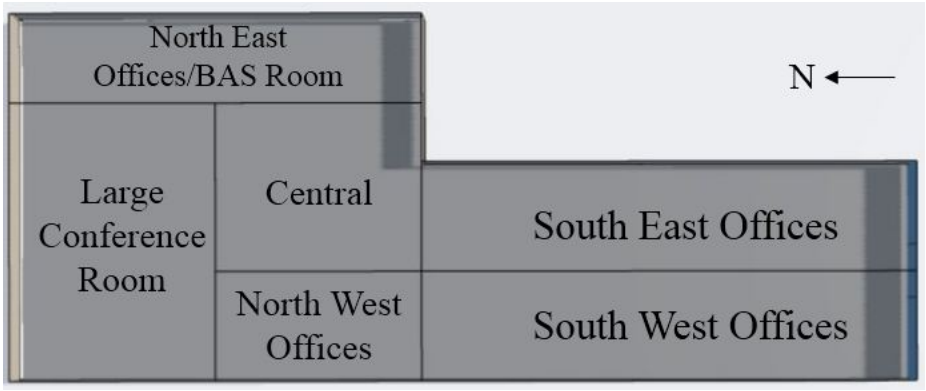

Figure 6.4: Thermal Zones Served by RadiantHeating

The thermal properties of the envelopes follow the requirements of ASHRAE 90.1-2016.

6.1.4 Target Control Variable and Baseline Control Strategy

The target supervisory level control variables are on/off of the whole heating system and SP2: the system supply water temperature setpoint. SP2 directly affects the amount of heating power supplied by the system. Also, this setpoint can affect occupants' thermal comfort by changing the mean radiant temperature of the building.

The baseline control strategy is a rule-based controller (RBC), which is modified based on a real-life control strategy for a radiant heating system. For the system on/off, the whole system is on unless the outdoor air temperature is above 10°C. For SP2, the general control concept is shown in Figure 6.2, which uses the error between the average indoor air temperature (T1) and its setpoint (SP1) to determine its value. The details of the logic are shown in Algorithm 2. It can be seen that the baseline control strategy contains several "magic numbers", which are determined by engineering experience without any scientific reasons.

Figure 6.5 shows an example of the behaviors of the average indoor air temperature under the baseline control strategy. It can be seen that, in the morning, it takes several hours for the average indoor air temperature to be close to its setpoint. The response time of the heavyweight building is significantly longer than the lightweight building. Also, the temperature never settles around the setpoint. This is caused by the slow thermal response of the system and the imperfect control strategy. It is expected that the reinforcement learning control method could find a better control strategy for the system on/off and SP2.

Algorithm 2 Baseline Control Strategy for the System Supply Water Temperature Setpoint (SP2) of RadiantHeating

procedure DETERMINESP2(T1, SP1, OAT, Mode, Kp)
▷ T1, SP1, OAT, Mode represent the conditions at the current control time step
▷ OAT is outdoor air temperature, Mode is regular heating/setback mode
▷ Mode is "regular heating" during 7:00-20:00 of weekdays, otherwise it is "setback"
▷ T1, SP1, OAT, SP2 have the unit degree Celsius
▷ Kp is a tunable hyperparameter larger than 0
    $stptError = SP1 - T1$
    if $stptError < 0.3$ **then**
        $SP2 = T1 - 5$ ▷ Recirculate all return water if SP1 is met
    else
        $propError = stptError * Kp$
        if Mode is "setback" **then** ▷ Enter the setback mode
            $SP2 = propError + 29.5$
            $SP2 = \min(51.5, SP2)$
        else ▷ Enter the regular heating mode
            if $stptError > 10$ **then**
                $SP2 = 65$
            else
                $oatBias = 4.35 - 1.71 * OAT$ ▷ Adjust the setpoint based on OAT
                $oatBias = \min(30, \max(-30, oatBias))$ ▷ Limit oatBias to -30 and 30
                $SP2 = propError + oatBias$
                $SP2 = \min(65, \max(0, SP2))$ ▷ Limit SP2 to 0 and 65
    return SP2

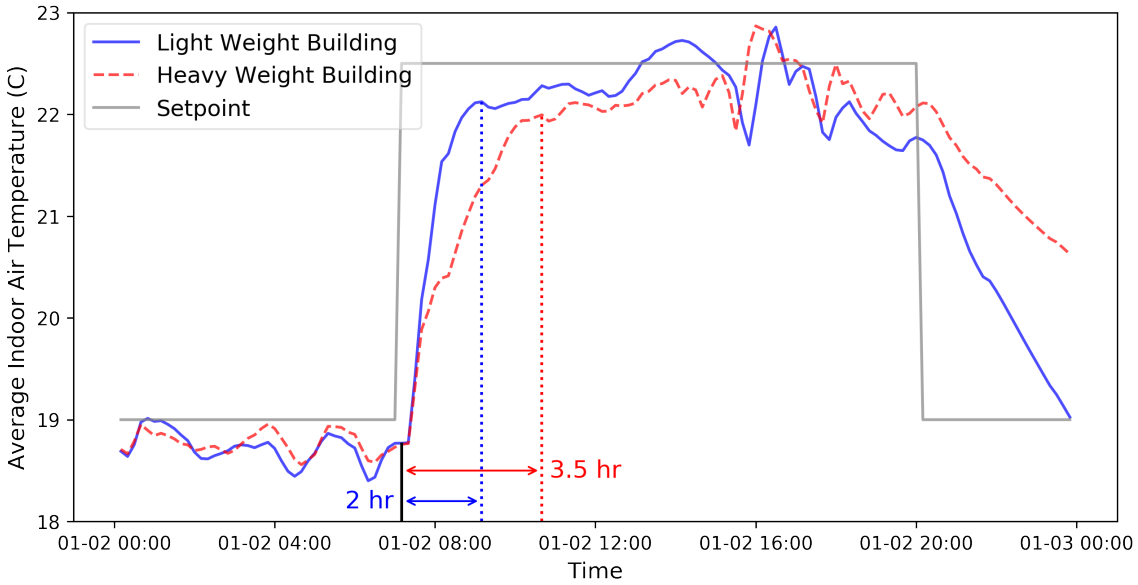

Figure 6.5: Behaviors of the Average Indoor Air Temperature of the Selected Day of RadiantHeating Using the Baseline Control Strategy in Pittsburgh Climate

6.1.5 Whole Building Energy Model

The whole building energy model is built using EnergyPlus (The U.S. Department of Energy, 2019a) version 8.3. The geometry 3D rendering is shown in Figure 6.6. The capacities of the system components are autosized by EnergyPlus using the design conditions of each climate. One simulation episode lasts for 2-month with 10-min as the simulation time step, as shown in Table 6.1.

One challenge to model RadiantHeating is that "Mullion radiant surface" cannot be directly modeled in EnergyPlus. A workaround modeling method is proposed and more details are presented in Zhang et al. (2019).

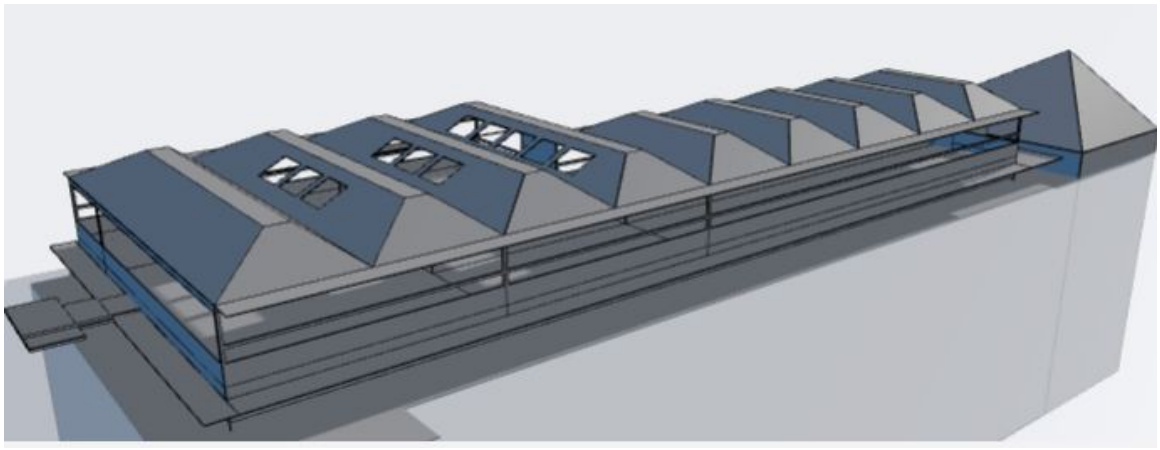

Figure 6.6: Geometry Rendering of the Whole Building Energy Model for RadiantHeating (rendered by BuildSimHub, Inc. (2018))

Table 6.1: Basic Simulation Settings of the Whole Building Energy Models for RadiantHeating Scenarios

Climate	**Thermal Mass**	**Simulation Period**	**Simulation Time Step**
Pittsburgh	Light	Jan 1st-Feb 28th	10-min
	Heavy		
Beijing	Light	Jan 1st-Feb 28th	10-min
	Heavy		
Shanghai	Light	Jan 1st-Feb 28th	10-min
	Heavy		

6.2 Reinforcement Learning Setup

6.2.1 State Design

The state is a stack of current and historical observations. The observation vector is shown in Table 6.2.

The length of the history in the state is determined using the method proposed in section 2.4.1. Figure 6.7 shows the relationship between the time interval n (each control time step is 10-min) and the distance correlation $dcor_n$, which represents the dependence between the observation at $t+1$ and $t-n$. A larger $dcor_n$ means a stronger dependency. It is interesting to find that, the lightweight and heavyweight buildings have almost the same $dcor_n$ curves. This is because only three items in the observation vector (ob) are building-weight related, including average indoor air temperature, average PMV and average system heating demand since the last control time step. The other items (such as weather conditions, time, setpoint schedules) are the same in both thermal-mass scenarios. By using Algorithm 1 with $dcorThres = 0.5$, the length of the history in the state is summarized in Table 6.3.

6.2.2 Action Design

The discrete action space for RadiantHeating is:

$$A_{radiantheating} = \{\text{turn-off}, 20°C, 25°C, ..., 65°C\}, \tag{6.1}$$

where the first action "turn-off" means to turn the heating system off (i.e., the supply water flow rate is zero) and the following actions are the setpoint for the system supply water temperature.

Table 6.2: Observation Vector in the State for RadiantHeating

No.	Item
1	Is weekday or not
2	Hour of the day
3	Outdoor air temperature (°C)
4	Outdoor air relative humidity (%)
5	Diffuse solar radiation (W/m^2)
6	Direct solar radiation (W/m^2)
7	Average indoor air temperature setpoint in setback mode (°C)*
8	Average indoor air temperature (°C)
9	Average predicted mean vote by Fanger's model (PMV)
10	Is building occupied (0 or 1)
11	Heating system on/off (0 or 1)
12	Heating system supply water temperature setpoint (°C)
13	Average system heating demand since last control time step (kW)†

*Setback mode occurs during weekends and 20:00-07:00 of weekdays.
†It is calculated using the equation $Q_{heating} = C_p m(T2 - T3)$ where C_p is the specific heat of water, m is the mass flow rate of hot water. It represents how much heating power is demanded by the building.

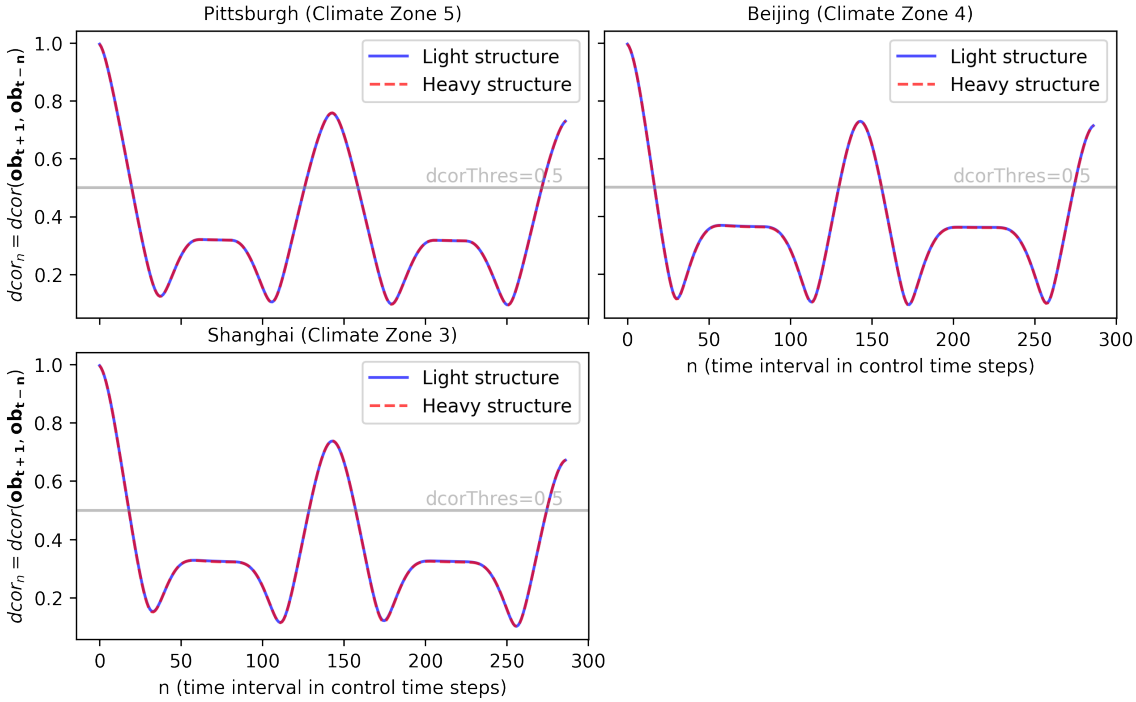

Figure 6.7: Relationship Between the Time Interval n and the Distance Correlation $dcor_n$ (specified in Equation (2.23)) for All the RadiantHeating Scenarios

Table 6.3: Length of History in the State for RadiantHeating Scenarios

Climate	**Thermal Mass**	**Length of History (control time steps)**
Pittsburgh	Light	20 (3.3 hr)
	Heavy	20 (3.3 hr)
Beijing	Light	16 (2.7 hr)
	Heavy	16 (2.7 hr)
Shanghai	Light	18 (3.0 hr)
	Heavy	18 (3.0 hr)

6.2.3 Reward Design

Energy and Comfort Metric

Energy saving and thermal comfort improvement are the control objectives governing the design of the reward function. RadiantHeating uses "heating demand" and "PMV" as the energy and comfort metric.

For the energy metric, RadiantHeating does not have an independent heating source. Thus, the electric demand or gas demand for heating is not available. As a common practice, heating demand ($Q_{heating}$) is used as the energy metric, i.e.,:

$$Q_{heating} = C_p m(T2 - T3), \tag{6.2}$$

where C_p is the specific heat capacity of water, m is the mass flow rate of water, T2 and T3 are the supply and return water temperature of the system (illustrated in Figure 6.2). This method is widely used in practice. However, this method is accurate only for steady-state situations, i.e., when all variables are not changed by time. Thus, this method can give an accurate estimation of the cumulative heating demand of a long period, but may fail to do so for the system transient behaviors. For example, when the supply water temperature suddenly increases, the method may over-estimate the heating demand since the hot water cannot charge all the pipes of the system immediately.

For the comfort metric (operational constraints), RadiantHeating uses predicted mean vote (PMV) based Fanger's model (Fanger, 1970) rather than setpoint notmet. This is because the Mullion radiant surface can significantly affect the thermal comfort feeling of an occupant by changing the zone mean radiant temperature. This effect cannot be considered if setpoint notmet is used as the comfort metric. PMV uses indoor air temperature, relative humidity, mean radiant temperature, air speed, occupants' clothing value and occupants' metabolic rate to calculate a comfort index value ranging from -3 to 3 (-3 is very cold and 3 is very warm). Thus, the effect on the mean radiant temperature is inherently included in PMV.

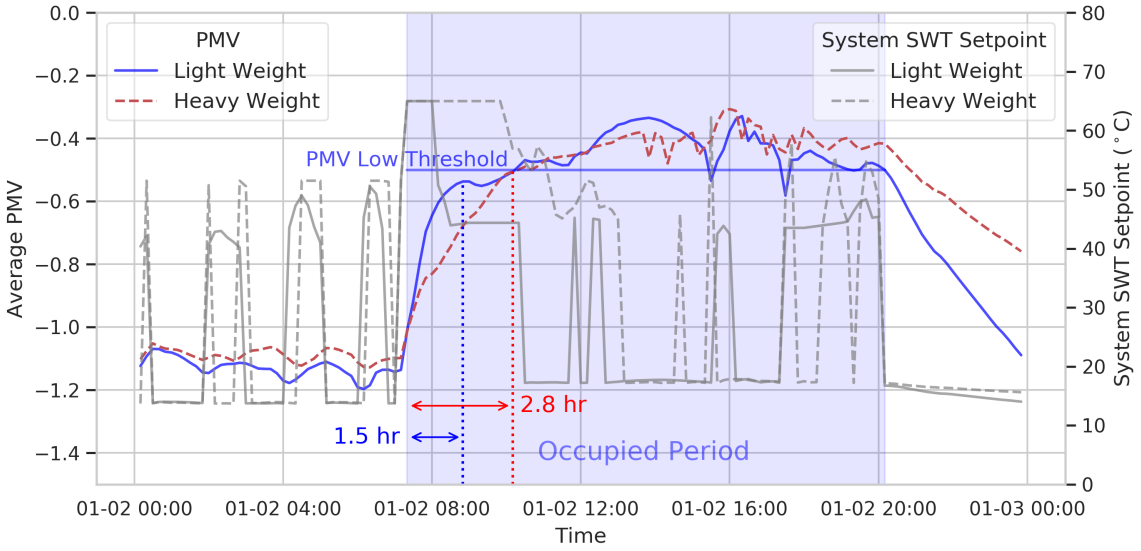

Figure 6.8: "Delayed Reward" Problem of RadiantHeating: Behaviors of the PMV vs. Control Actions (SWT) in a Selected Day of RadiantHeating using the Baseline Control Strategy in Pittsburgh Climate (SWT: supply water temperature)

Reward Design for the Slow Thermal Response

A reinforcement learning agent must obtain a clear reward signal to evaluate its state and action. The reward functions of VAVCooling and VAVHeating only use the state at current control steps to evaluate an agent, i.e., the agent is rewarded if the energy consumption of the current control step is low and the indoor air temperature of the current control step meets its setpoint. Based on the results, this design has achieved better energy efficiency compared to the baselines in VAVCooling and VAVHeating.

However, this design does not fit RadiantHeating because of its slow indoor air temperature response. As shown in Figure 6.8, in the winter morning, it takes several hours for the PMV to reach a comfortable level (larger than -0.5 as required by ASHRAE 55-2017 (ASHRAE, 2017)) even though the system supply water temperature setpoint is at the maximum. This is because the system has a slow air temperature response and air temperature is one of the major factors that influence PMV. This means that, if the reward function depends only on the PMV observation at the current control time step, the reward will remain small for several hours even though the right action is taken. This is called "delayed reward" problem and will lead to reinforcement learning divergence (Zhang et al., 2019).

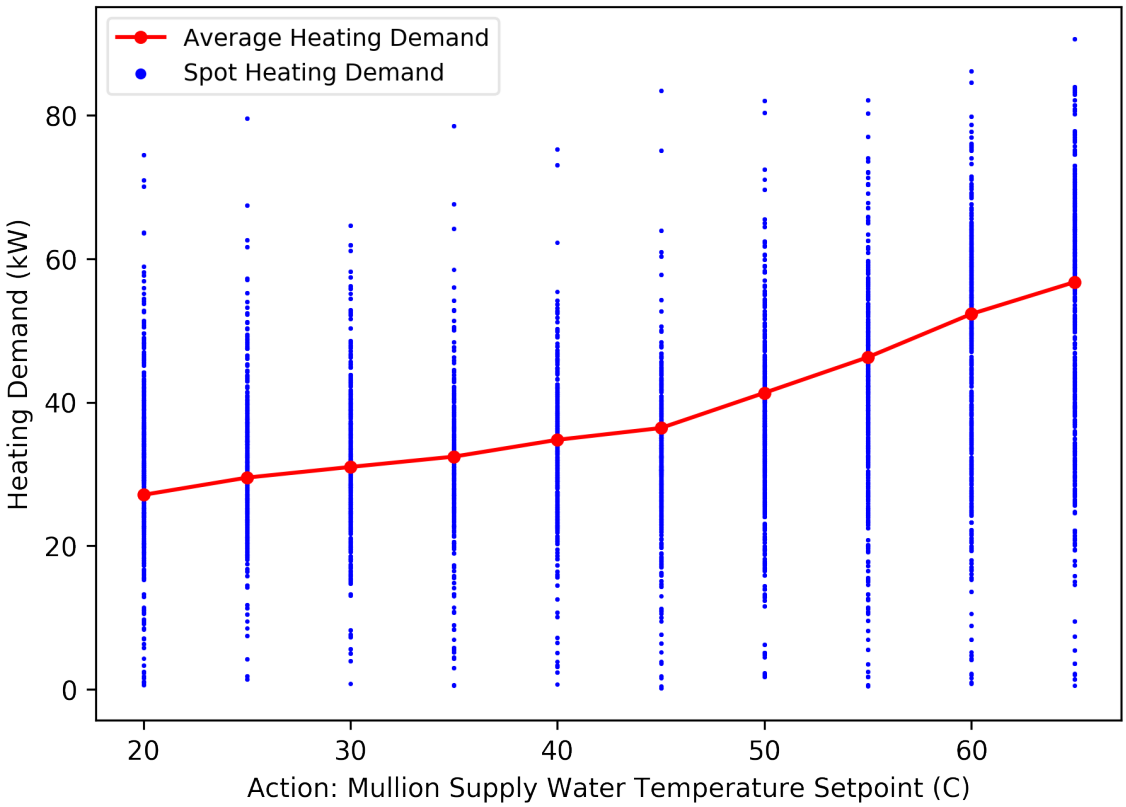

Figure 6.9: Imperfect Energy Metric Problem: Control Action vs. the Heating Demand of RadiantHeating in the Training Simulator of the Pittsburgh-Lightweight-Building Scenario using a Random Control Strategy

An interesting finding in Figure 6.8 is that, even though the PMV is below the threshold in the winter morning, it has an increasing trend when the system supply water temperature is high. This trend should be rewarded because it represents the potential to reach a comfortable indoor environment. Thus, in the reward function of RadiantHeating, the reward value is determined by the change of the PMV in two adjacent control time steps.

Reward Design for the Imperfect Energy Metric

Heating demand, calculated by a steady-state specific heat function (Equation (8.1)), is the energy metric for RadiantHeating. However, this is not an ideal energy metric and may confuse the reinforcement learning agents. This is illustrated in Figure 6.9, which shows the relationship between the system supply water temperature setpoint (the action) and the resulting heating demand. For the averaged data, the higher supply water temperature setpoint leads to the higher heating demand,

which is as expected. However, for the "spot" data, the heating demand ranges from nearly zero to nearly 100 kW for all the possible setpoint values. This is because:

- The heating demand is calculated by a steady-state function, which does not consider the transient behaviors of the system.

- Changing the *setpoint* may not lead to sufficient changes in the *actual* supply water temperature in a short period. For example, when changing the setpoint from a high level (e.g., 65 °C) to a low level (e.g., 30 °C), it may take several time steps for the actual water temperature to drop from the high level. This leads to the high heating demand at the low setpoint because the heating demand is determined by the difference between the *actual* system supply water temperature and the return water temperature. This slow response of the supply water temperature is caused by the system characteristics (more specifically, the working principle of the three-way mixture valve).

- The heating demand is low when the zone is overheated. In this situation, even though the supply water temperature is high, there will be little heat transfer from the Mullion radiant surfaces to the zone, so the heating demand is low.

The blurred relationship between the control actions and the heating demand may make the reinforcement learning difficult because the training of an RL agent can be significantly affected by individual samples. To give an RL agent a clear reward signal, the reward function includes a heuristic measure that gives high energy-related reward for the low-setpoint actions, and vice-versa.

Reward Function

Based on the above discussion, the reward function is formulated as:

$$
\begin{aligned}
R_{radiantheating,t} &= 1.0 - \left[(1 - P_{comfort}) * H_{energy} + P_{comfort} * H_{comfort}\right]_0^1, \\
&\text{where,} \\
H_{energy} &= \left[Q_{heating,normalized,t} + 0.02 * (T_{swstpt,t-1} - 15)\right]_0^1, \\
P_{comfort} &= \begin{cases} \left[2.0 * (19 - T_{ia,t})\right]_0^1 & \text{if not occupied,} \\ \left[5.0 * (-0.5 - PMV_t)\right]_0^1 & \text{if occupied,} \end{cases} \\
H_{comfort} &= \begin{cases} \left[\tau * (T_{ia,t} - T_{ia,t-1})\right]_0^1 & \text{if not occupied,} \\ \left[\beta * (PMV_t - PMV_{t-1})\right]_0^1 & \text{if occupied,} \end{cases}
\end{aligned}
\tag{6.3}
$$

where subscript t is a control time step, $Q_{heating,normalized}$ is the normalized average heating demand since last control time step, T_{swstpt} is the setpoint of the system supply water temperature (not normalized, the value is 15 if the heating system is turned-off), T_{ia} is the average indoor air temperature (not normalized), PMV is predicted mean vote by Fanger's model (not normalized), τ and β are tunable hyperparameters controlling the rewarding level for the trend of the average indoor air temperature or PMV.

Intuitively, this reward function aims to save the heating demand and keeps the PMV above -0.5 (during occupied time):

- As shown in $P_{comfort}$ in Equation (6.3), when the PMV is above -0.5, the reward function provides an output value solely based on H_{energy} in Equation (6.3). H_{energy} is a function combining the heating demand and a heuristic measure. This function penalizes high heating demands and high values of the supply water temperature setpoint.

- As shown in $P_{comfort}$ in Equation (6.3), when the PMV is below -0.5, the reward function will also consider the PMV to provide an output value. The PMV-related reward is determined by $H_{comfort}$ in Equation (6.3). This heuristic function rewards increasing trends in the PMV and penalizes decreasing trends in the PMV.

Table 6.4: Hyperparameters for the RL Training for RadiantHeating Scenarios

Item	Value	Item	Value
Simulation time step	10-min	A3C local agent number	16
Control time step	20-min	Reward discount factor	0.99
Nonlinear activation*	ReLu	RL interaction steps	3M
Optimizer	RMSProp	Learning batch size	25
RMSProp decay rate	0.99	Value loss weight	0.5
RMSProp momentum	0.0	τ in the reward	5.0
RMSPorp epsilon	$1e^{-10}$	β in the reward	10.0
Gradient clip method	L2-norm	Gradient clip threshold	1.0

Note: * nonlinear activation applies to the shared layers of the neural networks (except the one with the linear shared layers).

6.2.4 Action Repeat

Action repeat means repeating the same action for more than one control time step. This is a simple strategy to deal with slow-response simulators. For RadiantHeating, action repeat is set to 2, which means the same control action is executed in the simulators for two control time steps. One control time step is 10-min. The action repeat of 2 means the effective control time step is 20-min.

6.2.5 Hyperparameters

The reinforcement learning agents are trained with the hyperparameters shown in Table 4.5. In the training of each experiment scenario, seven neural network models and six learning rates will be tuned, as shown in Figure 3.5.

Table 6.5: Comparison of the Training and Perturbed Simulators for the RadiantHeating Scenarios

	Training	**Perturbed**	
		1	**2**
Weather	TMY3 for Pittsburgh, IWEC for other locations	AMY 2017	TMY/IWEC with additive white Gaussian noise
Occupancy Schedule	Deterministic with additive white Gaussian noise	Stochastic	
Plug-load Schedule	Deterministic with additive white Gaussian noise	Stochastic	

6.3 Training and Perturbed Simulators

Unlike VAVCooling and VAVHeating, there are only two perturbed simulators for RadiantHeating, as shown in Table 6.5. The perturbation of indoor air temperature setpoint is no longer included in RadiantHeating, because indoor air temperature setpoint is used neither in the state or in the reward function. The perturbed simulators still include the variations of weather conditions and occupancy/plug-load schedules.

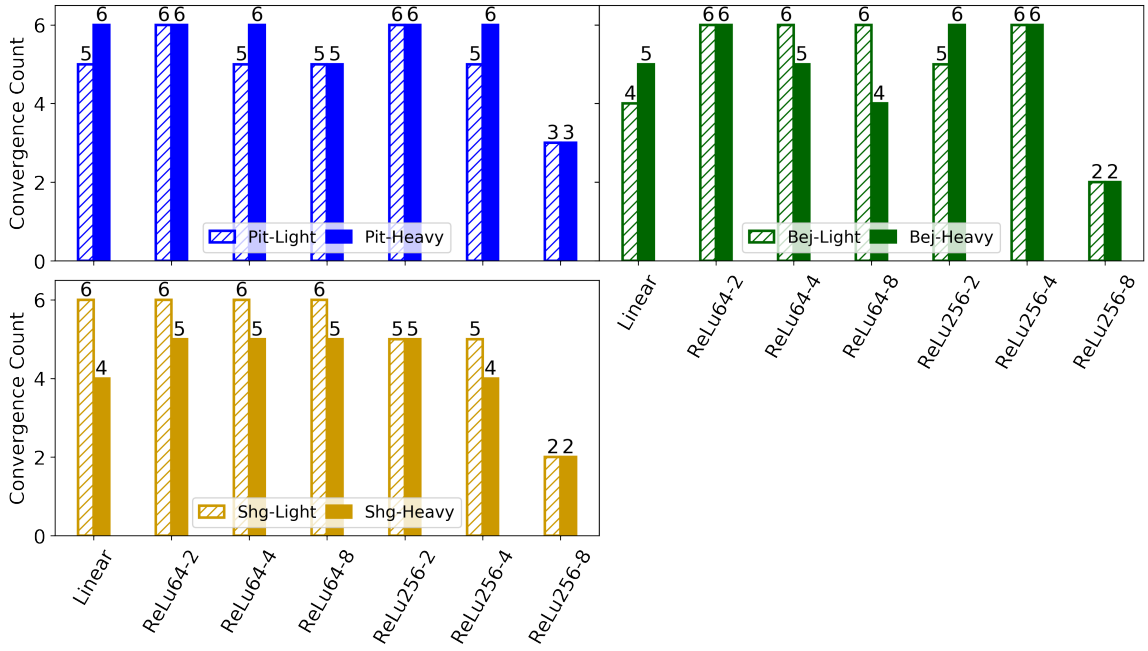

Figure 6.10: RadiantHeating: Convergence Robustness to Learning Rate (the count of convergence out of the six learning rates) vs. Neural Network Models

6.4 Results

6.4.1 Convergence Results

This section shows the results related to the convergence performance of the reinforcement learning training.

Six learning rates, from 1e-3 to 5e-6, are tuned for the seven neural network models. Figure 6.10 shows the convergence count out of the six learning rates vs. the neural network models for each experiment scenario. A larger convergence count means the corresponding neural network model is more robust to the different learning rates. It is found that the complexity of a neural network model has no obvious effects on the convergence count, except the ReLu256-8 model (the deepest and widest neural network model) which has the smallest convergence count in all the experiment scenarios. This result is different from VAVCooling and VAVHeating where the shallower neural network models are more robust to the different learning rates. Besides, the linear model does not have a larger convergence count than the other more complex neural network models, which is different from the result in VAVCooling.

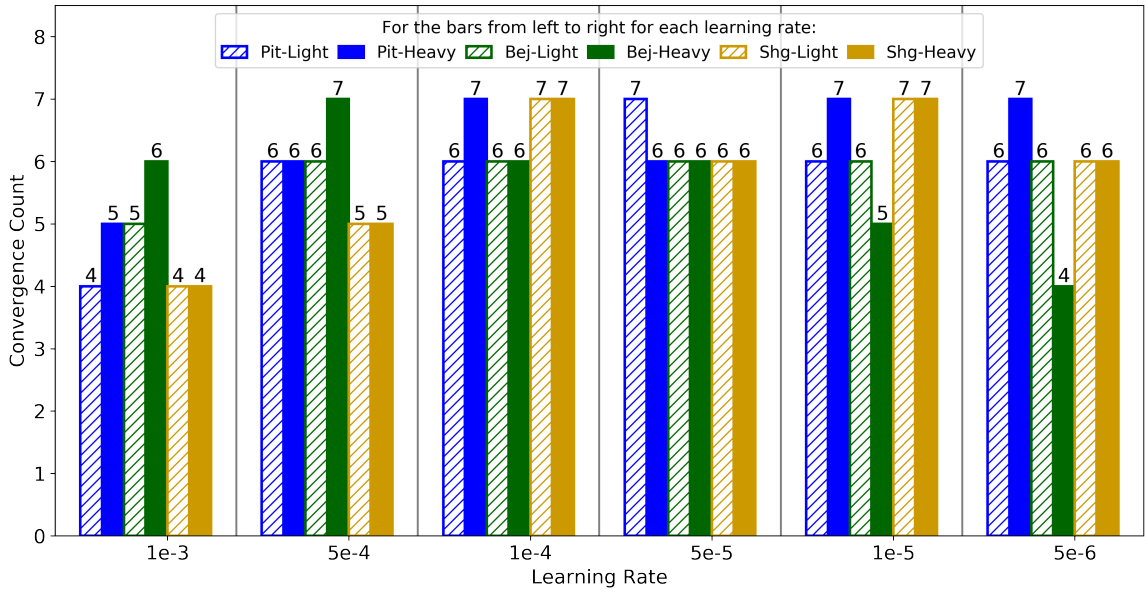

Figure 6.11: RadiantHeating: Convergence Count out of the Seven Neural Network Models vs. Learning Rate

Figure 6.11 shows the convergence count out of the seven neural network models at the different learning rate. There is no obvious trend between the learning rate and the convergence count. All the learning rates have relatively large convergence count for all the scenarios. This is different from VAVCooling and VAVHeating where the smaller learning rates are obviously more favorable than the larger learning rates for convergence.

Figures 6.12, 6.13 and 6.14 show the training evaluation histories of the seven neural network models at the learning rate 5e-4. It clear in the figures that, in all the experiment scenarios, the 8-layer neural network models have more fluctuations in the training evaluation histories than the shallower models. This result aligns with the result in VAVCooling. The linear model has smooth training evaluation histories in all the experiment scenarios. There is no consistent relationship between the width of a neural network model and the smoothness of the training evaluation histories. The building thermal mass level also does not show any obvious effects on the training evaluation histories.

6.4.2 Control Performance

The control performance of the reinforcement learning is evaluated by the percentage savings of the cumulative heating demand (Q_{saving}), and the cumulative time when the zone is occupied and the

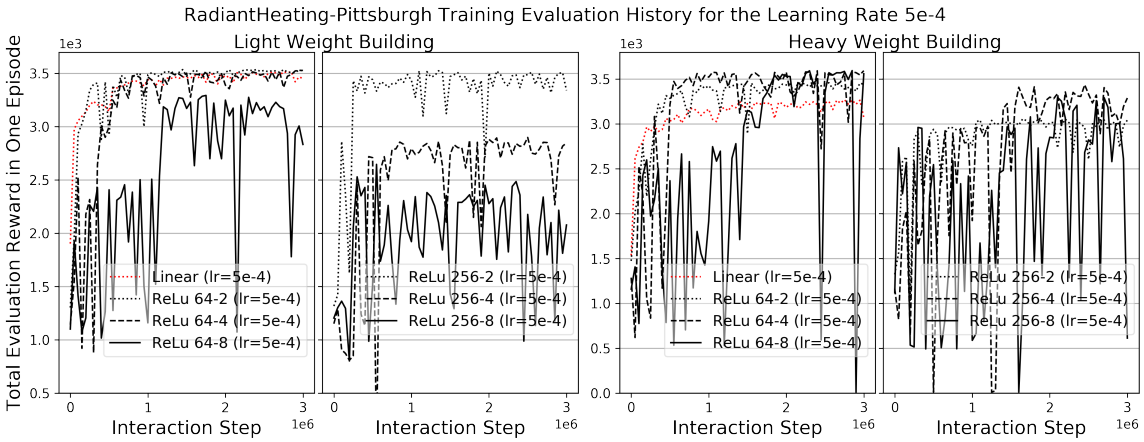

Figure 6.12: RadiantHeating: Training Evaluation History for the Learning Rate 5e-4 vs. Neural Network Models (Pittsburgh Climate)

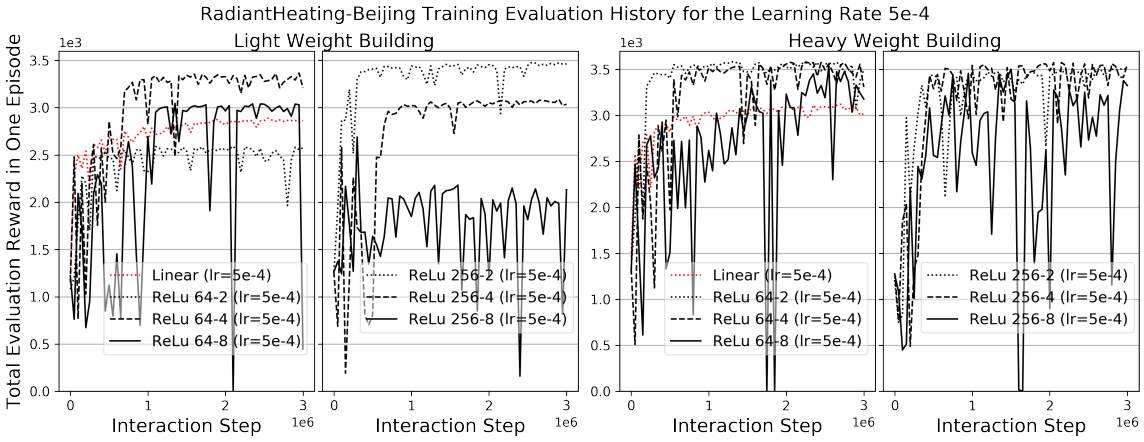

Figure 6.13: RadiantHeating: Training Evaluation History for the Learning Rate 5e-4 vs. Neural Network Models (Beijing Climate)

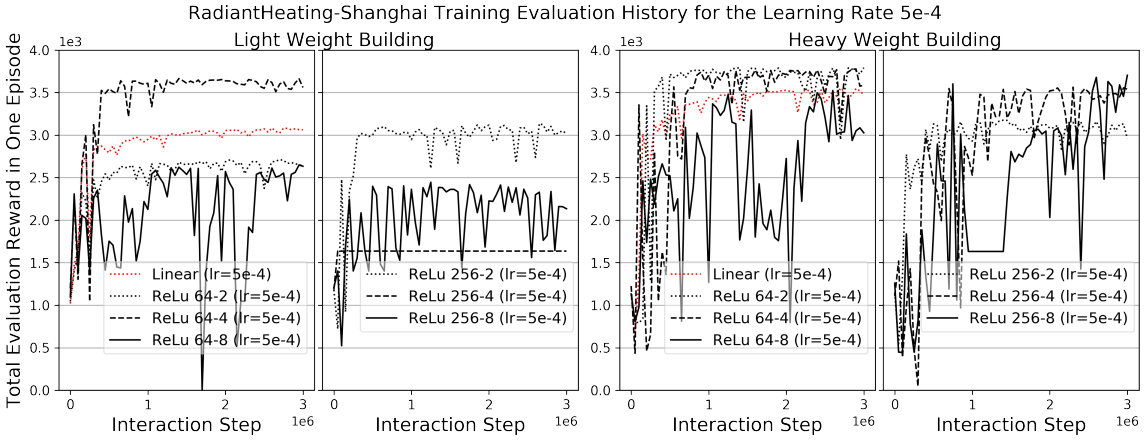

Figure 6.14: RadiantHeating: Training Evaluation History for the Learning Rate 5e-4 vs. Neural Network Models (Shanghai Climate)

PMV is less than -0.5 (named as "PMV notmet" or Ti_{pmvnmt}). Q_{saving} is

$$Q_{saving} = \frac{Q_{baseline} - Q_{rl}}{Q_{baseline}} * 100, \tag{6.4}$$

where Q_{rl} and $Q_{baseline}$ are the cumulative heating demand in one simulation episode under an RL-trained control policy and the baseline control strategy respectively. Ti_{pmvnmt} is

$$Ti_{pmvnmt} = Ti_{simstep} * \sum_{t=0}^{T_{simend}} \left(PMV_t < -0.5 \,\&\, OCCP_t == 1\right), \tag{6.5}$$

where $Ti_{simstep}$ is the time step length of the simulation (10-min in this case), t is one time step in the simulation, T_{simend} is the number of time steps in one simulation episode, and $OCCP$ is the occupancy flag. The PMV threshold "-0.5" is determined based on the requirement of ASHRAE 55-2017 (ASHRAE, 2017).

Figures 6.15, 6.16 and 6.17 show the control performance of the reinforcement learning method in the three climates. For each scenario, the performance in the training simulator and two perturbed simulators is shown. The two perturbed simulators have stochastic occupancy and plug-load schedules, and the Perturbed1 simulator uses AMY2017 weather data and the Perturbed2 simulator uses typical weather data with additive white noises. It is seen in the figures that:

- There is no consistent relationship between the control performance and the neural network model complexity across all the experiment scenarios.

- For the lightweight-building scenarios, almost all the neural network models have achieved obvious heating demand savings with less or similar PMV notmet time compared to the baseline.

- For the heavyweight-building scenarios, all the neural network models have achieved obvious heating demand savings, and the amount of the savings is higher than that in the lightweight-building scenarios. However, in the Pittsburgh and Shanghai scenarios, the PMV notmet time is higher than the baseline. This problem is caused by the imperfect design of the reward function.

  During the reinforcement learning training, an RL agent continues evolving itself to maximize the cumulative reward. For RadiantHeating, an RL agent is evaluated after every 50K interaction steps and the one with the maximum cumulative reward is selected as the final control policy. However, the max-reward control policy may not achieve the best balance between the heating demand saving and the PMV notmet. For example, in Figure 6.15 "Heavy Weight Building" section, the ReLu64-2 model has achieved 21% heating demand saving but the PMV notmet time is increased compared to the baseline. For the same scenario and the same neural network model, Figure 6.18 shows the training-simulator control performance of the other "checkpointed" control policies obtained during the reinforcement learning training. It can be seen that the max-reward choice is dominated by two other choices (i.e., the choices with both lower heating demand and lower PMV notmet time), even though their reward values are lower. This means the reward function is not fully consistent with the control performance evaluation metrics. If one would like a control policy with comparable or better thermal comfort than the baseline, a "better choice" is indicated in the figure. This control policy has much less heating demand and slightly less PMV notmet time than the baseline. Table 6.6 shows the control performance of the "better choice" RL control policy under the different simulators. It can be seen that, compared to the "max-reward" control policy, the "better choice" has less energy saving but also less PMV notmet time. Compared to the baselines, the "better choice" has achieved comparable or less PMV notmet time with obvious heating demand savings. This indicates that the reward function should be further studied to achieve a better balance between multiple control objectives.

- The control performance in the perturbed simulators is comparable or better than that in the training simulator. This indicates that the reinforcement learning control policies are tolerant of the variations in weather conditions and occupancy/plug-load schedules.

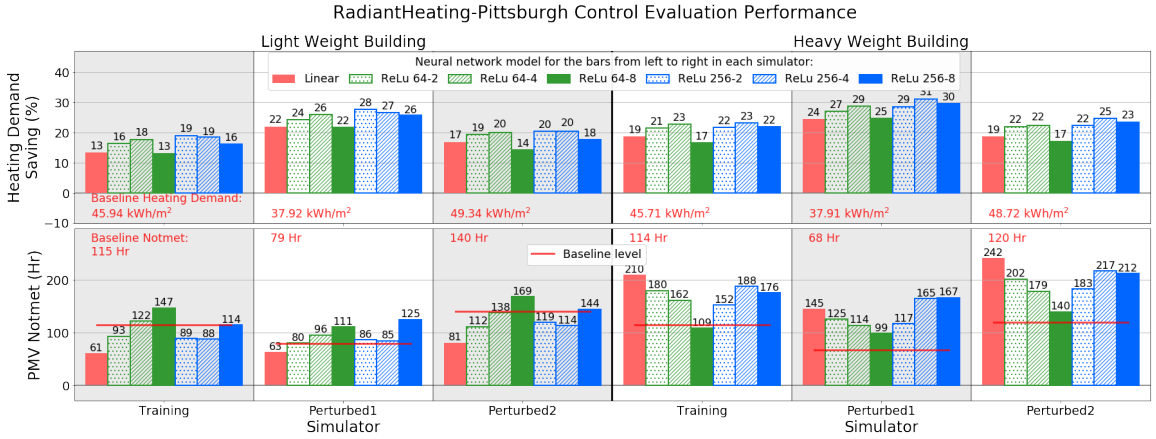

Figure 6.15: RadiantHeating: Control Performance in Pittsburgh Climate (the results of each neural network model are from the best-performing learning rate; baseline heating demand means the cumulative heating demand per building floor area using the baseline control strategy)

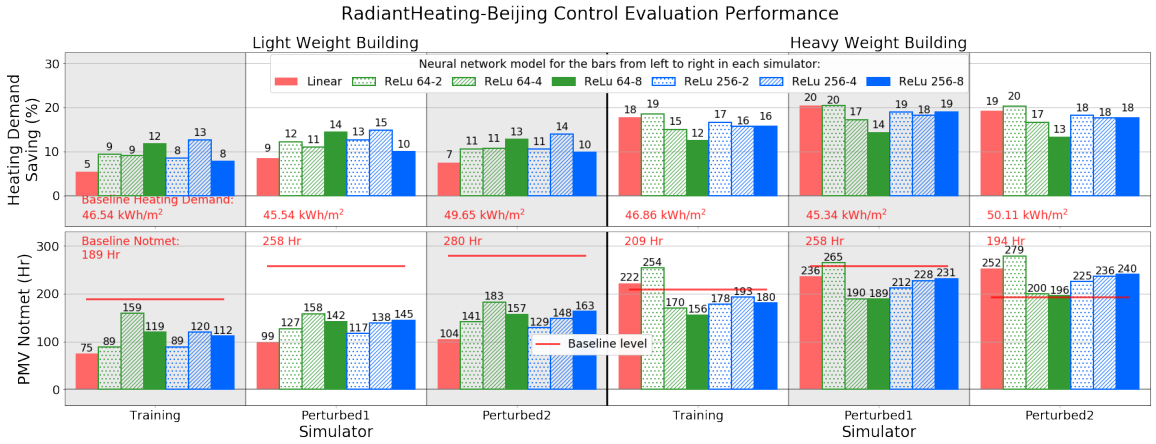

Figure 6.16: RadiantHeating: Control Performance in Beijing Climate (the results of each neural network model are from the best-performing learning rate; baseline heating demand means the cumulative heating demand per building floor area using the baseline control strategy)

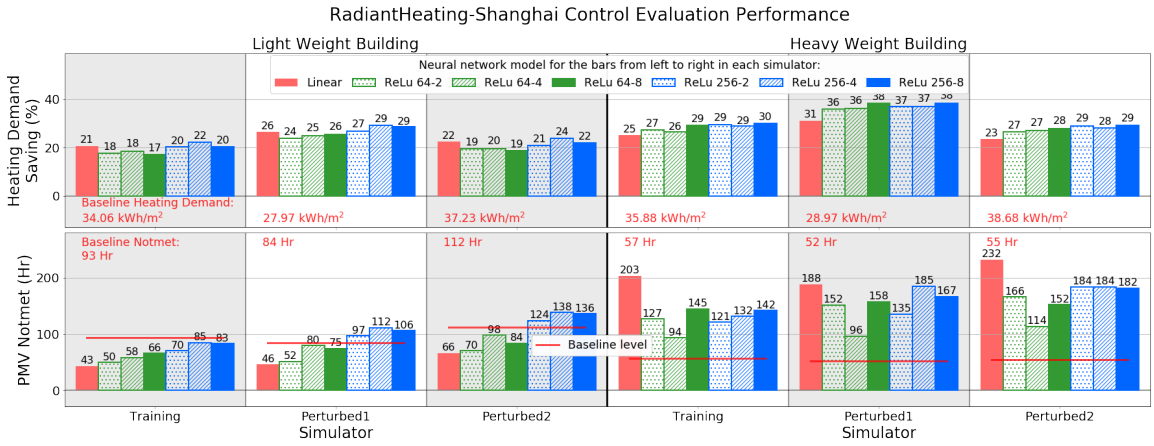

Figure 6.17: RadiantHeating: Control Performance in Shanghai Climate (the results of each neural network model are from the best-performing learning rate; baseline heating demand means the cumulative heating demand per building floor area using the baseline control strategy)

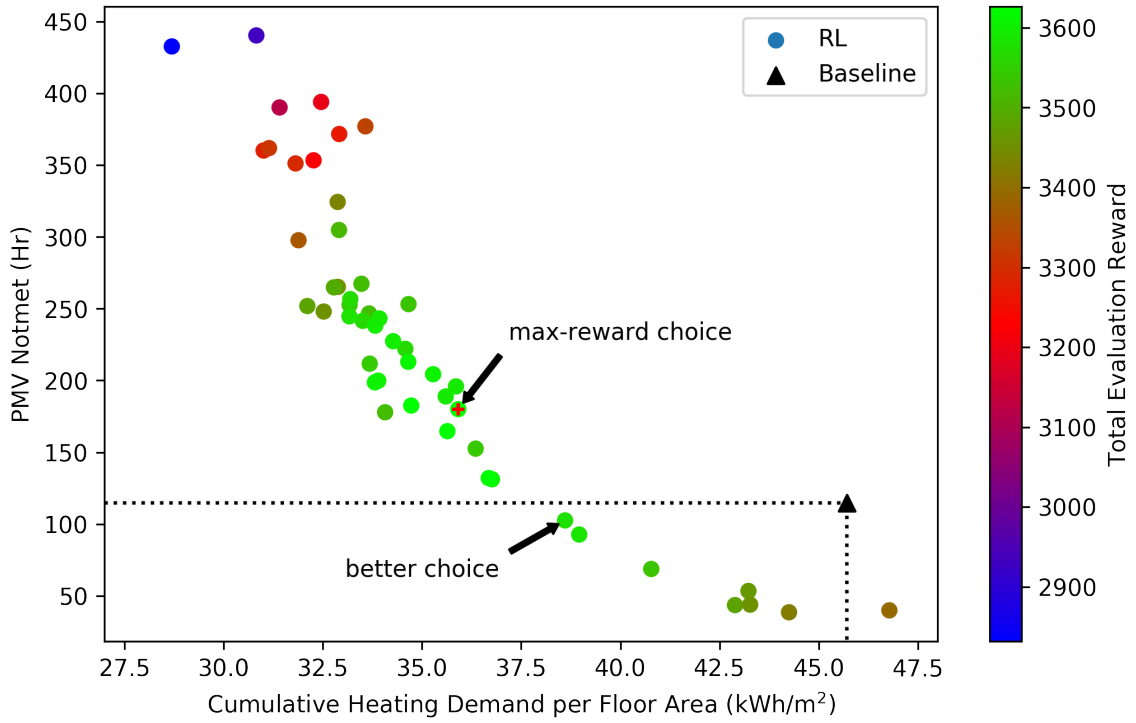

Figure 6.18: RadiantHeating: Control Performance in the Training Simulator for the Checkpointed Control Policies Obtained During the Reinforcement Learning Training for the Pittsburgh-Heavyweight-Building Scenario with the ReLu64-2 Model and the Best-performing Learning Rate (each dot in the figure represents a control policy after every 50K reinforcement learning interaction steps)

Table 6.6: Control Performance Comparison Between the “Max-reward” and “Better Choice” RL Control Policy for the Pittsburgh-Heavyweight-Building Scenario with the ReLu64-2 Model and the Best-performing Learning Rate

	Control Policy	**Training**	**Perturbed1**	**Perturbed2**
Heating Demand	”Max-reward”	21.4	27.0	22.0
Saving (%)	”Better choice”	15.5	22.9	17.7
	”Max-reward”	180	125	202
PMV Notmet (Hr)	”Better choice”	103	74	122
	Baseline	**114**	**68**	**120**

6.5 Summary and Discussion

This chapter presents the experiments related to RadiantHeating, a slow thermal response radiant heating system. The radiant heating system serves a 6-zone building but the environmental conditions of the 6 zones are averaged (effectively the system serves one zone). The control framework is applied for this system to reduce the heating demand and keep the PMV in occupied time larger than -0.5. The control variable is the supply water temperature setpoint. Six experiment scenarios are experimented, including three climates and two thermal mass levels. For each experiment scenario, 7 neural network models and 6 learning rates are tuned.

RadiantHeating has a simpler configuration than the previous VAV systems. However, the dynamics of RadiantHeating is not simple. Firstly, it has a much slower thermal response than the VAV systems. The slow thermal response hence affects the behaviors of PMV. It is demonstrated that the PMV has a delayed behavior, i.e., a large change in the supply water temperature setpoint only leads to a limited change of PMV. Secondly, the heating demand cannot be appropriately measured. This chapter uses the specific heat equation to calculate the heating demand, but this equation is accurate only for steady-state operations. As a result, the spot measurement of the heating demand is not accurate and has large variations. Thus, if the reward function is designed as the form in the VAV experiments (i.e., the reward is negatively proportional to the heating demand and PMV setpoint notmet), the reinforcement learning training may experience convergence problems (Zhang et al., 2018a).

A reward function with heuristics is designed to remediate the above problems. The heuristics includes two measures: firstly, the reward value is proportional to the PMV increase in one control time step if the PMV is below the setpoint; secondly, the reward value is negatively proportional to the heating demand and the supply water temperature setpoint (i.e., a low setpoint value has a high reward). The two heuristic measures are decided after a thorough study on the system dynamics and several trial-and-error tests. The convergence of this problem is significantly improved by using this reward function design. This indicates that adding heuristics to the reward function can be a practical solution for the convergence problems of reinforcement learning. However, it should be noted that, the heuristics in the reward function limits the exploration space of an RL agent.

The convergence results are firstly presented after the RL training. Interestingly, almost all the neural network models can converge with a wide range of learning rates, except the ReLu256-8 which is the most complex neural network model in the experiments. This is attributed to the heuristics

in the reward function that makes the whole control problem easier. It is also shown that the RL agents are easy to converge for all the tuned learning rates. This is different from the results of VAVCooling and VAVHeating where the small learning rates are more favorable for convergence. This results again indicate that the learning rate must be tuned for different experiment scenarios. There is not a unified value for the learning rate that works for all scenarios.

The control performance of the trained control policies is evaluated by the heating demand saving and PMV notmet time in one training simulator and two perturbed simulators. For the control performance in the training simulator, the RL control policies have achieved 10%-30% heating demand savings for all the experiment scenarios. However, in the heavyweight building scenarios, the RL control polices have delivered the slightly higher-than-baseline PMV notmet time. This is because the reward function does not achieve the desired balance between the two control objectives, heating demand saving and PMV constraint fulfillment. It is shown in an example that, during the training process of an RL agent, there are multiple solutions that are baseline-dominated. But those solutions have not been selected as the final control policy because the reward function weights the heating demand saving more than the PMV constraint fulfillment. The RL control polices are also evaluated in two perturbed simulators with the changed weather conditions and occupancy/plugload schedules. There are no perturbations for indoor air temperature setpoint because it is not applicable to this system. It is found the control performance is similar or even better than the training control performance.

The effects of the neural network model complexity on the control performance is also shown in this chapter. For the Shanghai scenarios, different neural network models have achieved similar control performance. For the Pittsburgh and Beijing scenarios, different neural network models have achieved different control performance, but there is not a clear relationship between the neural network model complexity and the control performance. The results do not support the hypothesis on the benefits of deep reinforcement learning. However, the conclusion is derived from the limited experiments shown in this chapter, so it cannot be generalized.

Chapter 7

Experiment 4: ChilledWater

This chapter presents the experiments related to ChilledWater, a three-chiller chilled water system. There are four experiment scenarios in this chapter, as shown in Figure 7.1. The scenarios do not include different thermal mass levels because they do not apply to this system. As usual, seven different neural network models and six learning rates are tuned for each scenario.

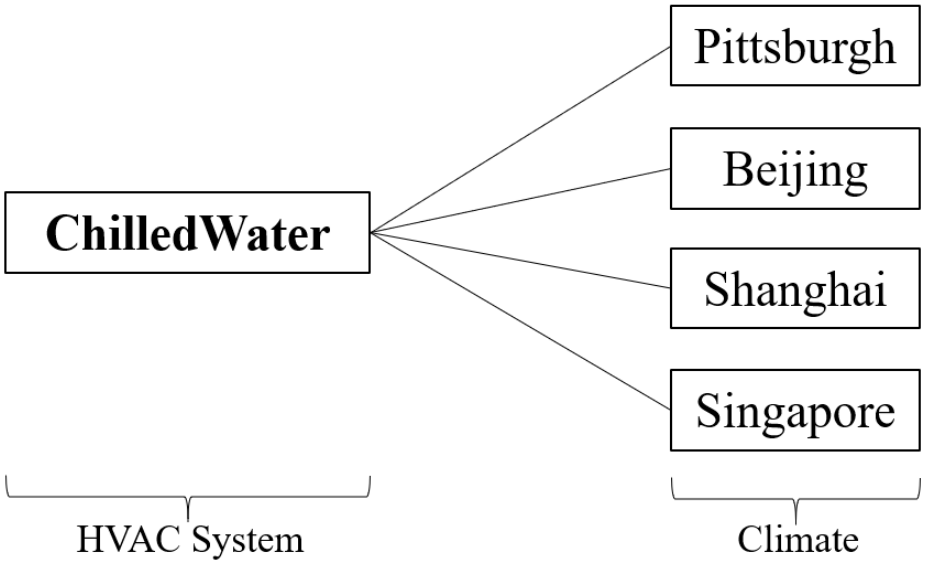

Figure 7.1: ChilledWater Experiment Scenarios

7.1 System Description

7.1.1 System Layout

ChilledWater is a multi-chiller system to supply chilled water for any cooling purposes. Different from the previous systems, ChilledWater is a **primary** system without considering secondary usages. This means this system is completely decoupled from "buildings". The system layout is shown in Figure 7.2. The system has three chillers connected in parallel, a cooling tower, and two water pumps. The system is used to generate chilled water to meet the provided cooling demands. Since no "buildings" are modeled, building thermal mass level does not apply to this system.

As shown in Figure 7.2, two of the three chillers have a larger cooling capacity ("big chiller") and the other one has a smaller cooling capacity ("small chiller"). The two big chillers have the same capacity. This is a common configuration found in practice to increase the flexibility of the chillers to handle different cooling demands. During operation, it is assumed that the cooling demand is evenly distributed to "online" chillers (i.e., the chillers that are on), and each chiller has its own internal control strategy (e.g., via adjusting the inlet vane or the motor speed) to deliver chilled water at a predefined setpoint temperature. However, the supply chilled water will be warmer than the setpoint (also called setpoint notmet) if the cooling demand exceeds the total cooling capacity of "online" chillers.

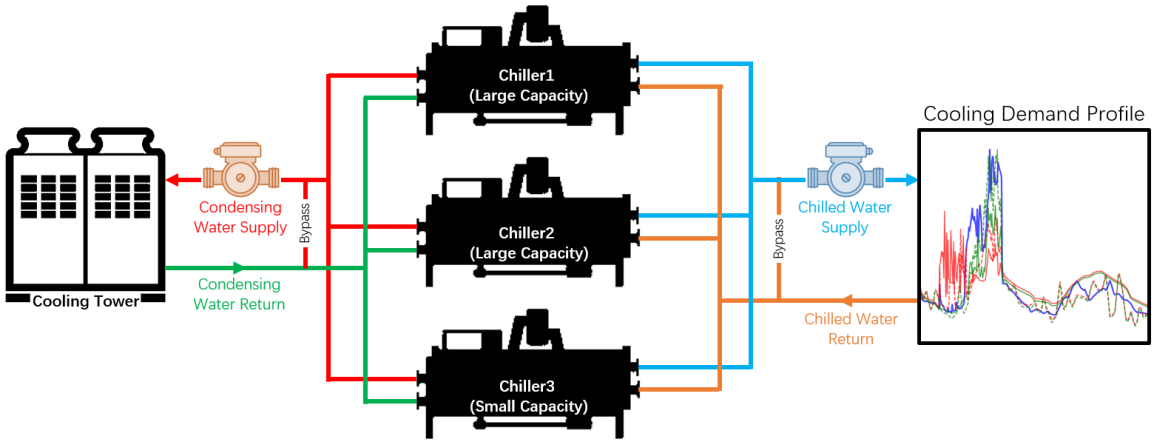

Figure 7.2: System Layout of ChilledWater

7.1.2 Target Control Variable and Baseline Control Strategy

The efficiency of a chiller is affected by multiple factors, including outdoor weather conditions, the supply water temperature and the partial load ratio (i.e., the ratio of the delivered cooling load over the total cooling capacity of a chiller). Since weather conditions and the supply water temperature cannot be manipulated (the supply water temperature setpoint is usually predefined by designers), the partial load ratio is the only variable that will affect a chiller's efficiency. Thus, the target supervisory level control variable is the on/off of each chiller. This will change the partial-load ratio of each chiller, and affect their efficiencies.

As discussed previously, the three chillers consist of two big chillers with the same capacity and a small chiller with a smaller cooling capacity. As a result, there are five operation modes for the three chillers' on/off regarding the maximum cooling capacity that can be provided, as shown in Table 7.1. The baseline control strategy is rule-based and has a simple principle:

> If the cumulative time when the partial load ratio (PLR) of one of the chillers is larger than 90% ("plr90Time") is more than 20-min, turn on a new chiller; If the cumulative time when the PLR of one of the chillers is less than 30% ("plr30Time") is more than 20 min, turn one or more chillers off.

More specifically, the baseline control strategy is shown in Algorithm 3.

Algorithm 3 Baseline Control Strategy for the On/Off of the Three Chillers of ChilledWater

procedure DETERMINEOPMODE(demand,curMode, plr90Time, plr30Time, capSmall, capBig)
▷ demand: the current cooling demand; curMode: the current on/off mode of the chillers
▷ capSmall, capBig: the cooling capacity of the small chiller and the big chiller
  $newMode = curMode$
  if $plr90Time \geq$ 20-min **then** ▷ Need start new chiller(s)
    if $curMode == 1$ **then** ▷ Currently only the small chiller is on
      $newMode = 3$ ▷ Start a new big chiller
    else if $curMode == 2$ **then** ▷ Currently only the big chiller is on
      if $\frac{demand}{2} < capSmall * 0.9$ **then** ▷ If an additional small chiller is sufficient
        $newMode = 3$ ▷ Start a new small chiller
      else
        $newMode = 4$ ▷ Start a new big chiller
    else
      $newMode = 5$ ▷ Start all chillers
  else if $plr30Time \geq$ 20-min **then** ▷ Need shut-off one chiller
    if $curMode == 5$ **then** ▷ Currently all chillers are on
      $newMode = 4$ ▷ Shut-off the small chiller
    else if $curMode == 4$ **then** ▷ Currently two big chillers are on
      if $demand < capSmall * 0.9$ **then** ▷ If the small chiller alone is sufficient
        $newMode = 1$ ▷ Only keep the small chiller on
      else if $demand < capBig * 0.9$ **then** ▷ If the big chiller alone is sufficient
        $newMode = 2$ ▷ Only keep the big chiller on
      else
        $newMode = 3$ ▷ Shut-off one big chiller, turn-on on small chiller
    else if $curMode == 3$ **then** ▷ Currently one big and one small chiller are on
      if $demand < capBig * 0.3$ **then** ▷ If one big chiller alone is over-qualified
        $newMode = 1$ ▷ Only keep the small chiller on
      else
        $newMode = 2$ ▷ Only keep the big chiller on
    else if $curMode == 2$ **then** ▷ Currently only one big chiller is on
      if $demand < capSmall * 0.9$ **then** ▷ If one small chiller alone is sufficient
        $newMode = 1$ ▷ Shut-off the big chiller, turn-on the small chiller
  return newMode

Table 7.1: Operation Modes for the Three Chillers' On/Off Status for ChilledWater

No.	Description
1	Only the small chiller is on
2	Only one big chiller is on
3	One big and one small chiller are on
4	Two big chillers are on
5	All chillers are on

7.1.3 Operational Constraints of a Chiller

A chiller cannot be arbitrarily turned-on or shut-off. Its operation should meet the following conditions:

- Avoid shortcycling: Shortcycling is the rapid turning-on/shutting-off/turning-on of a chiller. It reduces a chiller's life-span. This is because the motor in a chiller needs much higher current during starting than during stable operation. The heat must be dissipated quickly otherwise it will damage the mechanical and electrical components in the motor. The dissipation of the excessive heat is faster in a running motor. In this study, it is assumed 20-min is the minimum time interval between a turn-on action and a shut-off action or between a shut-off action and a turn-on action.

- Avoid low partial-load ratio (PLR): A chiller has a low limit for its PLR. If the PLR is lower than the limit, refrigerant in the evaporator cannot be completely vaporized so the liquid refrigerant may enter the compressor to cause severe mechanical damages. Note that, some modern chillers have a protection strategy called "false-loading", which introduces additional "artificial" cooling loads (e.g., via hot-gas bypass) to increase the PLR. However, false-loading significantly reduces the energy efficiency of a chiller.

The baseline control strategy in Algorithm 3 has already avoided most shortcycling and low-PLR problems.

Table 7.2: Basic Simulation Settings of the Energy Models for ChilledWater

Climate	**Simulation Period**	**Simulation Time Step**
Pittsburgh	June 1st-Aug 31th	10-min
Beijing	June 1st-Aug 31th	10-min
Shanghai	June 1st-Aug 31th	10-min
Singapore	Sep 1st-Nov 30th	10-min

7.1.4 Whole Building Energy Model

The experiments need two building energy models, one for the system ChilledWater, and the other one (named "context model") to generate the cooling demand profiles.

The energy model of the system is built using EnergyPlus version 8.7. The model of ChilledWater does NOT contain any information of "building" because ChilledWater is a primary system. The basic settings of the energy model are shown in Table 7.2.

In addition to the model of ChilledWater, a "context model" is still needed to generate the cooling demand profiles. This model contains the information of a building, and can be assumed to be the building that ChilledWater serves. This study uses a 10-level, 27000 m^2 office building energy model as the context model for ChilledWater. This context model is used to generate the cooling demand profiles and size the chillers. The basic settings of the context model are the same as the energy model of ChilledWater.

Based on the cooling demand profiles, the configurations of the chillers in the different climates are shown in Table 7.3.

Table 7.3: Configurations of the Chillers in the Different Climates for ChilledWater

Climate	**Chiller Name**	**Nominal Capacity**	**Reference COP**	**Min PLR**	**Max PLR**
Pittsburgh	Trane CVHE 1080 (“Big”)	1080 kW	7.39	0.20	1.05
	Trane RTHB 542 (“Small”)	542 kW	5.26	0.30	1.01
Beijing	Carrier 19XR 1294 (“Big”)	1294 kW	7.61	0.16	1.02
	Carrier 23XL 686 (“Small”)	686 kW	5.91	0.20	1.04
Shanghai	Carrier 19XR 1656 (“Big”)	1656 kW	8.24	0.17	1.04
	Carrier 19XR 869 (“Small”)	869 kW	5.57	0.18	1.03
Singapore	Carrier 19XR 1294 (“Big”)	1294 kW	7.61	0.16	1.02
	Carrier 23XL 686 (“Small”)	686 kW	5.91	0.20	1.04

7.2 Reinforcement Learning Setup

7.2.1 State Design

The state contains current and historical observations. The observation vector is shown in Table 7.4. The "cooling demand" ($Q_{cooldemand}$) and "cooling delivered" ($Q_{cooldelivered}$) in the observation vector are calculated by the following equations:

$$Q_{cooldemand} = C_p m_{chw} \left(T_{chwreturn} - T_{chwstpt}\right), \tag{7.1}$$

$$Q_{cooldelivered} = C_p m_{chw} \left(T_{chwreturn} - T_{chwsupply}\right), \tag{7.2}$$

where C_p is the specific heat capacity of water, m_{chw} is the mass flow rate of the supply chilled water, $T_{chwreturn}$ is the return chilled water temperature, $T_{chwsupply}$ and $T_{chwstpt}$ are the supply chilled water temperature and its setpoint.

The length of the history in the state is determined using the method explained in section 2.4.1. Figure 7.3 shows the relationship between the time interval n and the distance correlation $dcor_n$. One control time step is 10-min. Similar to the other systems, the $dcor_n$ has a repeated pattern after $n = 144$ because the system operation and weather conditions have daily cyclic patterns. In this study, dcorThres is set to 0.5, which means the number of the historical observations in the state is $n - 1$ where the n has the value that the $dcor_n$ firstly drops below 0.5. The results are summarized in Table 7.5.

7.2.2 Action Design

The discrete action space for the on/off of the three chillers is:

$$A_{chilledwater} = \{\text{opMode1}, \text{opMode2}, \text{opMode3}, \text{opMode4}, \text{opMode5}\}, \tag{7.3}$$

where opMode1∼opMode5 are the operation modes listed in Table 7.1.

Table 7.4: Observation Vector in the State for ChilledWater

No.	Item
1	Is weekday or not
2	Hour of the day
3	Outdoor air temperature (°C)
4	Outdoor air relative humidity (%)
5	Big chiller 1 on/off
6	Big chiller 1 shortcycling flag*
7	Big chiller 1 PLR
8	Big chiller 2 on/off
9	Big chiller 2 shortcycling flag*
10	Big chiller 2 PLR
11	Small chiller 1 on/off
12	Small chiller 1 shortcycling flag*
13	Small chiller 1 PLR
14	Supply Chilled Water Temperature (°C)
15	Supply Chilled Water Temperature Setpoint (°C)
16	Supply Chilled Water Mass Flow Rate (kg/s)
17	Cooling Demand (kW)†
18	Cooling Delivered (kW)†
19	ChilledWater total electric demand (kW)

*The flag = 1 means that shortcycling occurs.
†See Equation (7.1) and (7.2).

Table 7.5: Length of the History in the State for ChilledWater Scenarios

Climate	Length of History (control time steps)
Pittsburgh	21 (3.5 hr)
Beijing	26 (4.3 hr)
Shanghai	29 (4.8 hr)
Singapore	18 (3.0 hr)

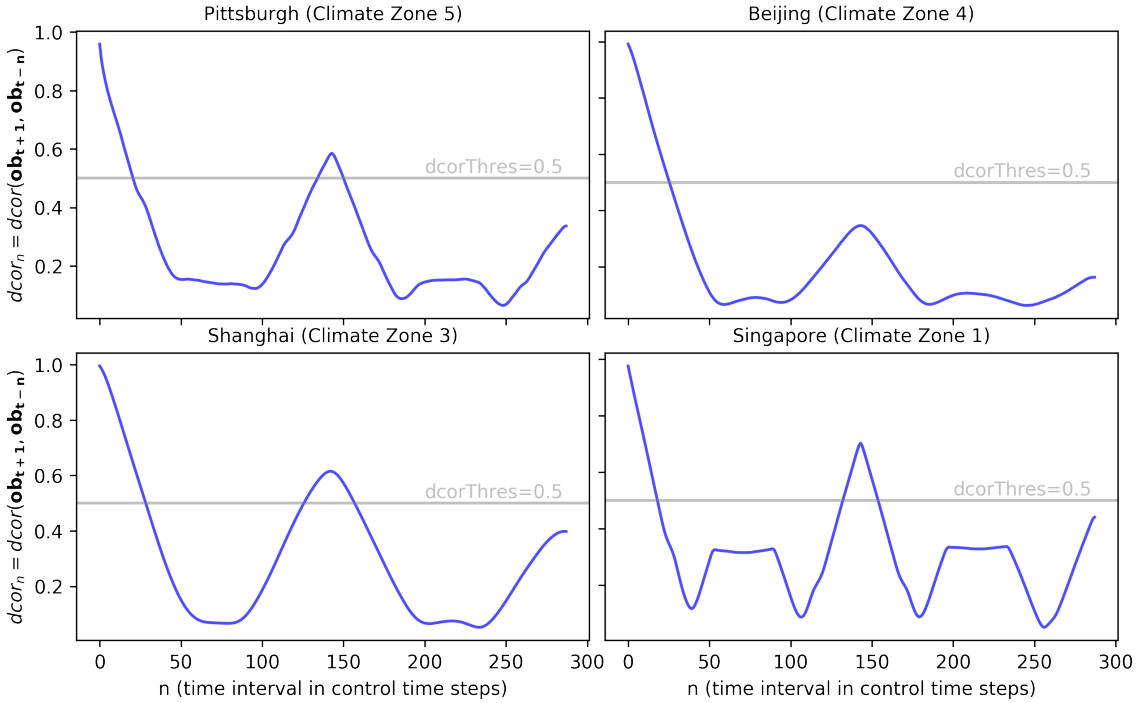

Figure 7.3: Relationship Between the Time Interval n (control time step is 10-min) and the Distance Correlation $dcor_n$ for All the ChilledWater Scenarios

7.2.3 Reward Design

The reward function is shown in Equation (7.4):

$$
\begin{aligned}
R_{chilledwater,t} &= 1.0 - \left[\beta * P_{energy} + \tau * P_{swt} + P_{sc} + P_{plr}\right]_0^1, \\
&\textit{where}, \\
P_{energy} &= E_{chilledwater,t}, \\
P_{swt} &= \left[T_{chwsupply,t} - T_{chwstpt,t}\right]^+, \\
P_{sc} &= \begin{cases} 1, & \text{if either chiller experiences shortcycling at } t, \\ 0, & \text{else}, \end{cases} \\
P_{plr} &= \begin{cases} 1, & \text{if either chiller's PLR is less than its low limit at } t, \\ 0, & \text{else}, \end{cases}
\end{aligned}
\tag{7.4}
$$

where t is a control time step, β and τ are tunable hyperparameters to control the weights on the energy penalty (P_{energy}) and setpoint notmet penalty (P_{swt}), $E_{chilledwater}$ is the normalized electric

Table 7.6: Hyperparameters for the RL Training for ChilledWater Scenarios

Item	Value	Item	Value
Simulation time step	10-min	A3C local agent number	16
Control time step	10-min	Reward discount factor	0.99
Nonlinear activation*	ReLu	RL interaction steps	2.5M
Optimizer	RMSProp	Learning batch size	5
RMSProp decay rate	0.99	Value loss weight	0.5
RMSProp momentum	0.0	τ in the reward$^+$	50
RMSPorp epsilon	$1e^{-10}$	β in the reward	1.0
Gradient clip method	L2-norm	Gradient clip threshold	5.0

Note: * nonlinear activation applies to the shared layers of the neural networks (except the one with the linear shared layers); $^+$ $\tau = 50$ means the reward value will be zero if the supply chilled water temperature exceeds the setpoint by more than 0.2 °C.

demand of ChilledWater, $T_{chwsupply}$ and $T_{chwstpt}$ are the supply chilled water temperature and its setpoint (both are normalized). Intuitively, this reward function penalizes high energy consumption, supply chilled water temperature setpoint notmet, shortcycling, and lower-than-threshold PLR.

7.2.4 Hyperparameters

The hyperparameters are summarized in Table 7.6. Severn neural network models and six learning rates will be tuned for each scenario, as shown in Figure 3.5.

Table 7.7: Comparison of the Training and Perturbed Simulators for the ChilledWater Scenarios

	Training	**Perturbed**			
		1	**2**	**3**	**4**
Weather	TMY3 for Pittsburgh, IWEC for other locations	AMY 2017	TMY/IWEC with additive white Gaussian noise	AMY 2017	TMY3/IWEC with additive white Gaussian noise
Cooling Demand Profile	Generated by the "context" building energy models based on the configurations listed in Table 7.8				

7.3 Training and Perturbed Simulators

The configurations of the training and perturbed simulators are shown in Table 7.7. The simulators are different in weather conditions and the cooling demand profiles. The cooling demand profiles are generated by the "context" building energy models (described in section 7.1.4) using the configurations listed in Table 7.8. These configurations are the same as the training and perturbed simulators of VAVCooling and VAVHeating. However, the whole building energy models with these configurations are only used to generate the cooling demand profiles, rather than to train and evaluate an RL agent. The generated cooling demand profiles at an arbitrarily short period are shown in Figure 7.4. It can be seen that the profiles are obviously different from each other during the weekdays.

Table 7.8: Configurations of the "Context" Building Energy Models to Generate the Cooling Demand Profiles for the ChilledWater Scenarios

	"Context" BEM for training simulator	**"Context" BEM for perturbed simulator**			
		1	**2**	**3**	**4**
Weather	TMY3 for Pittsburgh, IWEC for other locations	AMY 2017	TMY/IWEC with additive white Gaussian noise	AMY 2017	TMY3/IWEC with additive white Gaussian noise
Occupancy Schedule	Deterministic with additive white Gaussian noise	Stochastic			
Plug-load Schedule	Deterministic with additive white Gaussian noise	Stochastic			
IAT Setpoint	Deterministic with additive white Gaussian noise	PMV-based		Deterministic	

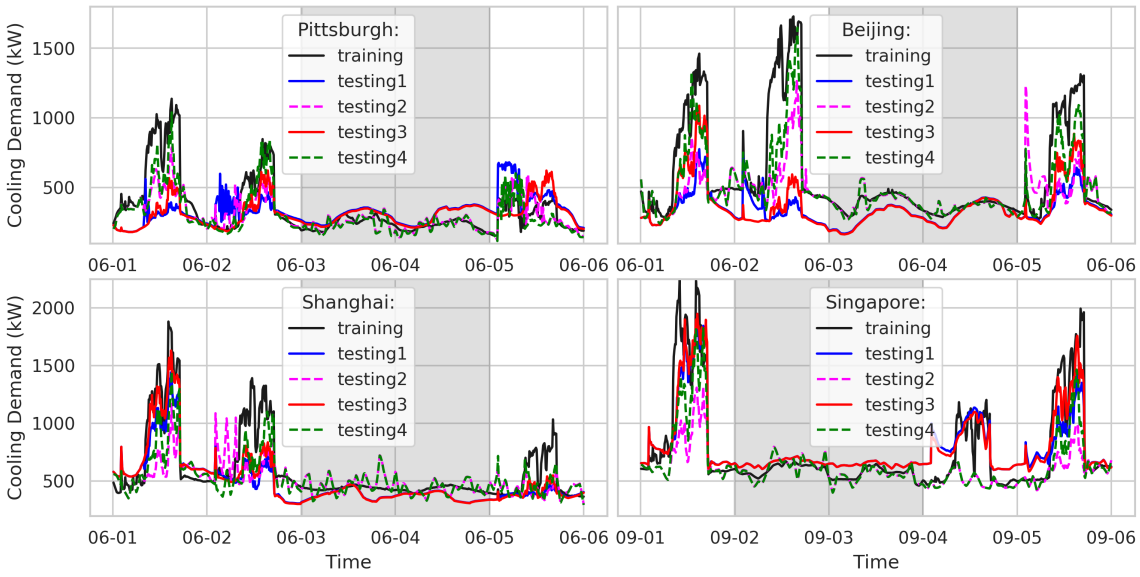

Figure 7.4: Cooling Demand Profiles at a Selected Time Period of the Training and Perturbed Simulators for ChilledWater (the time period at the shaded region is weekends)

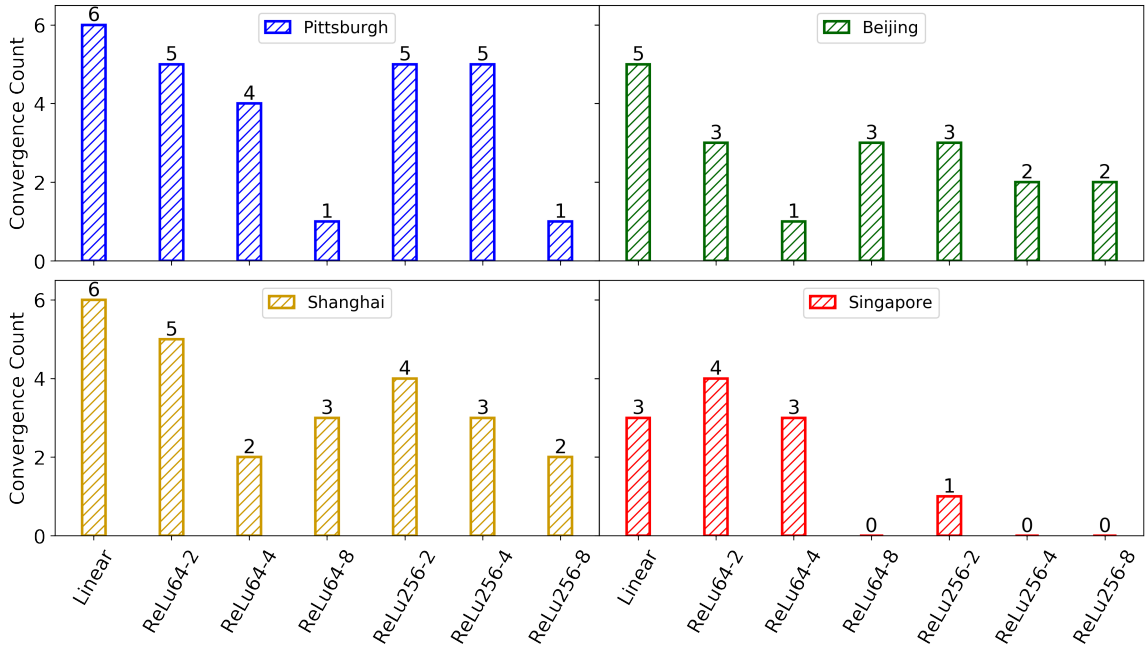

Figure 7.5: ChilledWater: Convergence Robustness to the Learning Rate (the count of convergence out of the six learning rates) vs. Neural Network Models

7.4 Results

7.4.1 Convergence Results

During reinforcement learning training, seven neural network models and six learning rates are tuned for each experiment scenario. The relationship between the convergence count out of the six learning rates and the neural network models is shown in Figure 7.5. As a general trend, the linear and shallow neural network models have a larger convergence count, which means they are more robust to the different learning rates. The results align with that in VAVCooling. Figure 7.6 shows the relationship between the convergence count out of the seven neural network models and the six learning rates. This figure shows which learning rates are favorable for the training convergence. In general, a learning rate has a larger convergence count, which is similar to the results in VAVCooling and VAVHeating.

Figures 7.7, 7.8, 7.9 and 7.10 show the training evaluation histories of the seven neural network models at the learning rate 1e-5. It is interesting to find that, for all the training evaluation histories, the total evaluation reward quickly jumps from an initial low level to a relatively high level and then

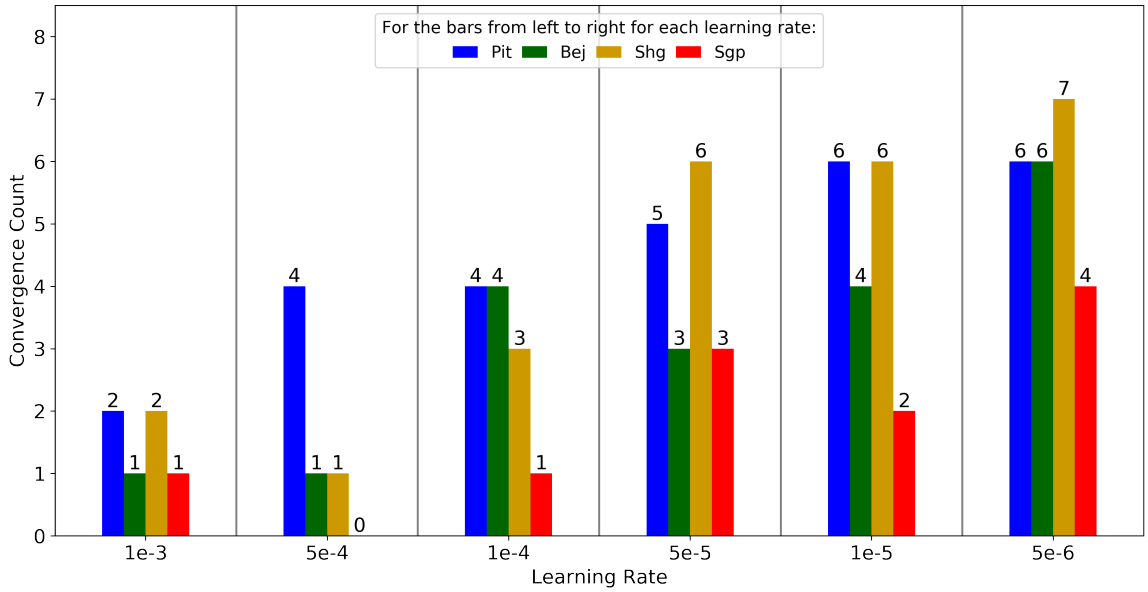

Figure 7.6: ChilledWater: Convergence Count out of the Seven Neural Network Models vs. the Learning Rate

fluctuates or stays constant at the high level. This is different from that in VAVCooling, VAVHeating, and RadiantHeating, where the total evaluation reward gradually increases (for converged cases). This indicates that the reinforcement learning agents may be stuck at some local optimal solutions. The neural network models and climates have no obvious effects on the training evaluation history.

7.4.2 Control Performance

The control performance of the RL agents is evaluated by four criteria, including:

1. The percentage saving of the cumulative electric energy consumption of ChilledWater (E_{saving}):

$$E_{saving} = \frac{E_{baseline} - E_{rl}}{E_{baseline}} * 100, \tag{7.5}$$

where $E_{baseline}$ and E_{rl} are the cumulative electricity consumption in one simulation episode under the baseline and RL control policies, respectively.

2. The number of simulation time steps of the setpoint notmet for the supply chilled water

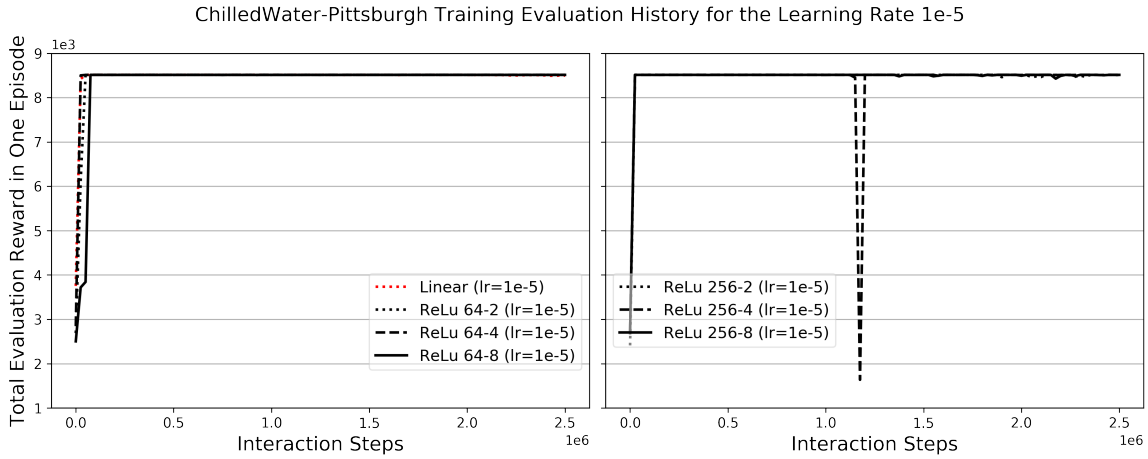

Figure 7.7: ChilledWater: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Pittsburgh Climate)

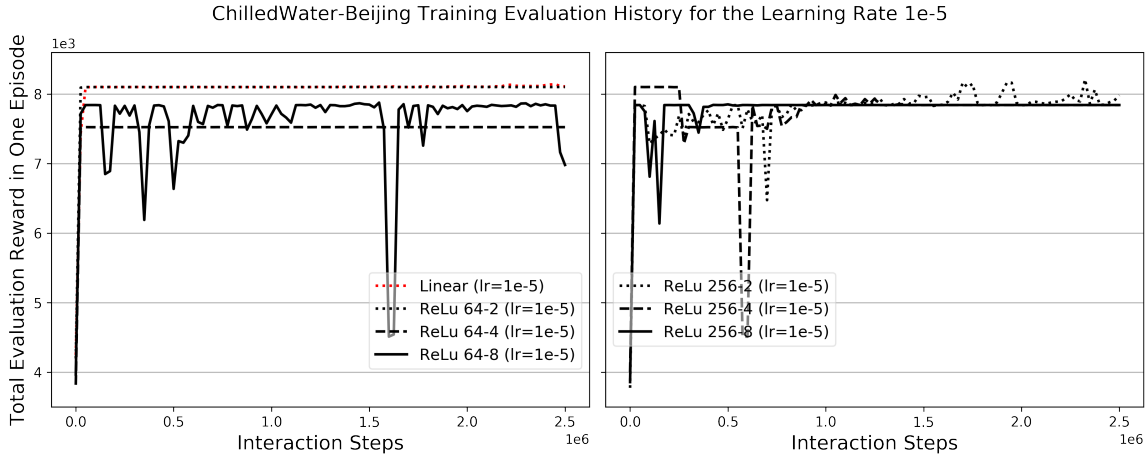

Figure 7.8: ChilledWater: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Beijing Climate)

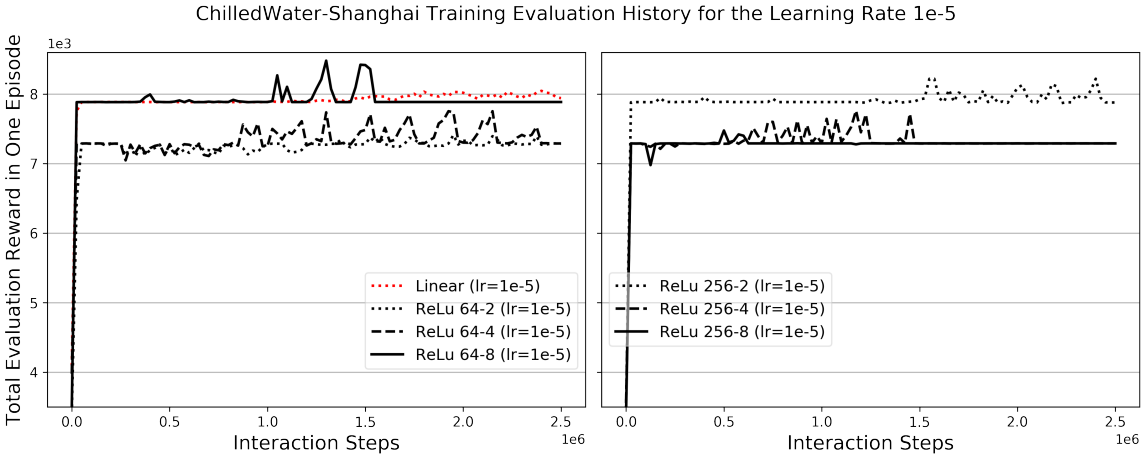

Figure 7.9: ChilledWater: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Shanghai Climate)

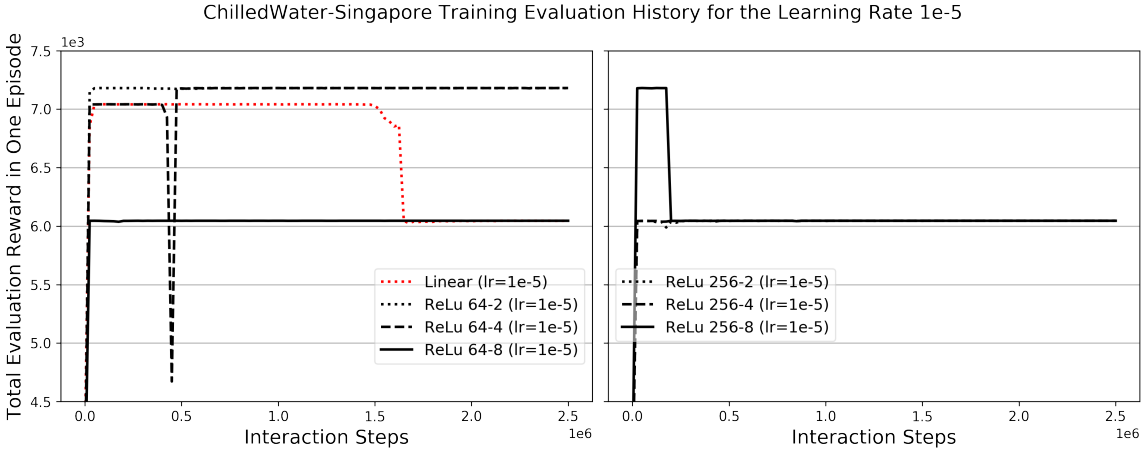

Figure 7.10: ChilledWater: Training Evaluation History for the Learning Rate 1e-5 vs. Neural Network Models (Singapore Climate)

temperature ($Tstep_{nmt}$):

$$Tstep_{nmt} = \sum_{t=0}^{T_{simend}} \big((T_{chwsupply,t} - T_{chwstpt,t}) > 0.2\big), \tag{7.6}$$

where t is one simulation time step, T_{simend} is the number of simulation time steps in one episode, $T_{chwsupply}$ and $T_{chwstpt}$ are the supply chilled water temperature and its setpoint (°C).

3. The number of simulation time steps that either chiller experiences shortcycling (turn-on then shut-off or shut-off then turn-on within 20-min).

4. The number of simulation time steps that either chiller's PLR is lower than the minimum limit (the minimum limits are shown in Table 7.3).

The control performance in both training and perturbed simulators is shown in Figures 7.11, 7.12, 7.13 and 7.14. It can be seen that:

- None of the reinforcement learning control policies could dominate the baseline performance (i.e., a control policy that has better performance than the baseline in all criteria). Even though some RL control policies have less energy consumption and less setpoint notmet time than the baseline, they face the shortcycling and/or low-PLR problems. Also, compared to the results in the other systems, the amount of energy savings (if there is any) is limited.
- There is no clear relationship between the control performance and the neural network model complexity. These results align with that in VAVCooling, VAVHeating, and RadiantHeating.
- The control performance in the perturbed simulators is similar to that in the training simulator, which indicates that the reinforcement learning agents are tolerant of the variations in weather conditions and cooling demand profiles.

Overall. the control performance of ChilledWater is not comparable to the other systems. The RL trained control policies can only achieve limited energy savings, but bring operational constraint violations such as the shortcycling and low-PLR problems. However, it should be noted that, one simulation episode contains nearly 13000 time steps. Hence, even though the absolute number of the operational constraint violations is large (i.e., tens to hundreds), it only accounts for a small proportion of the total simulation period.

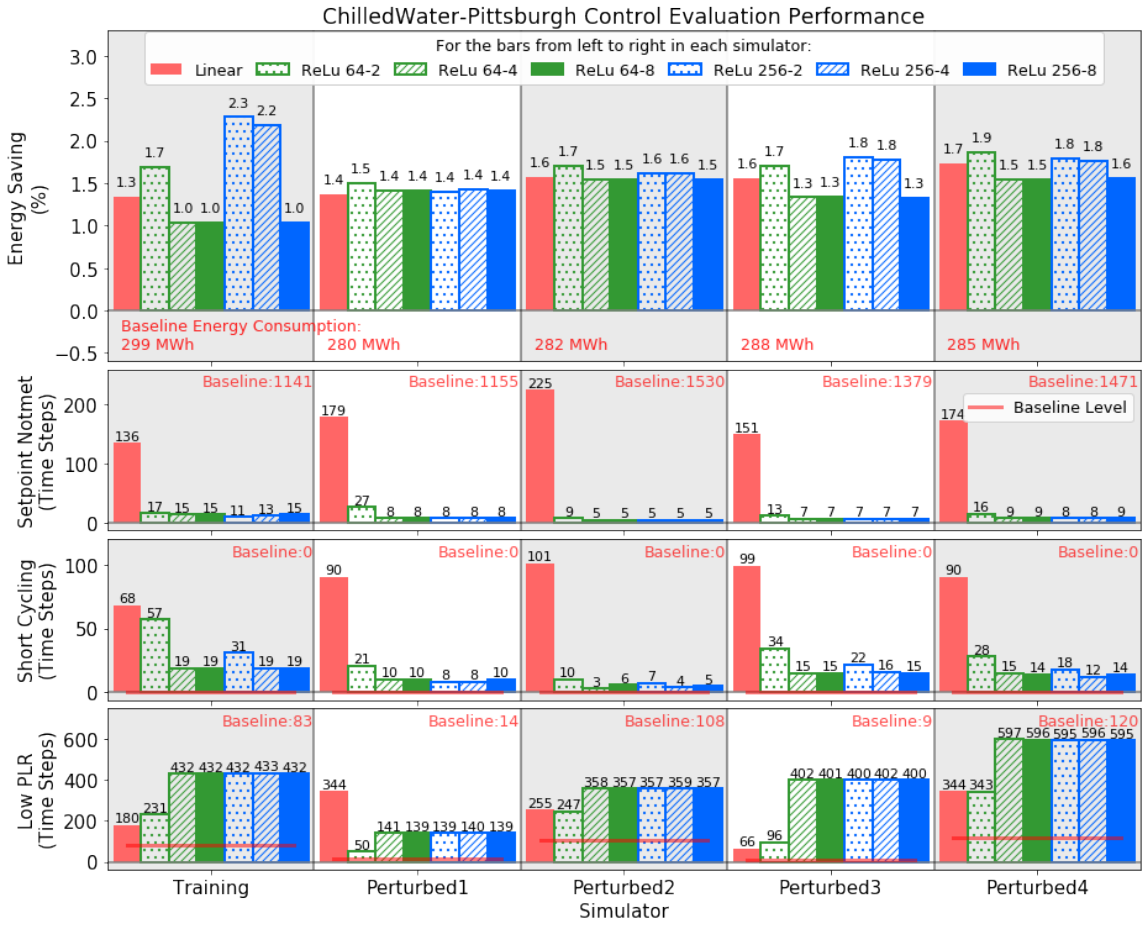

Figure 7.11: ChilledWater: Control Performance in Pittsburgh Climate (the results of each neural network model are from the best-performing learning rate)

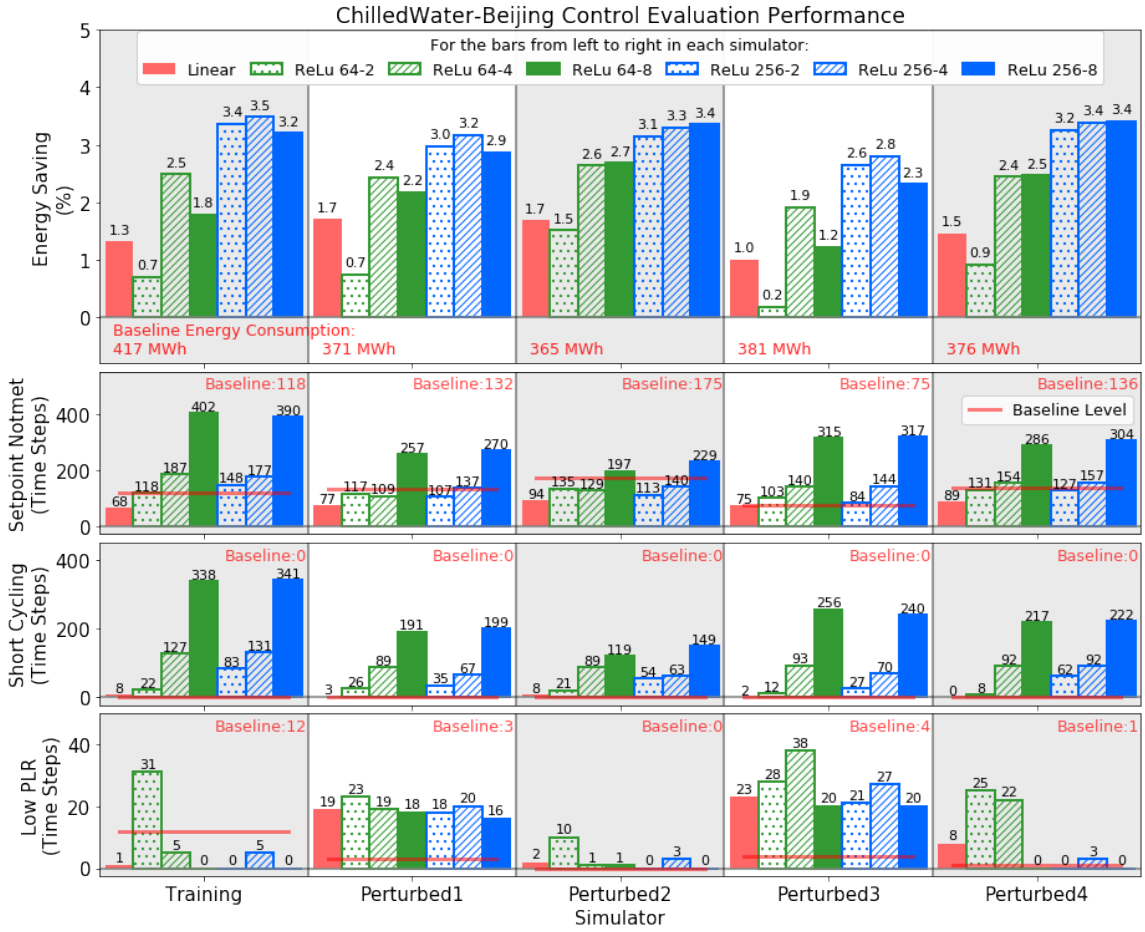

Figure 7.12: ChilledWater: Control Performance in Beijing Climate (the results of each neural network model are from the best-performing learning rate)

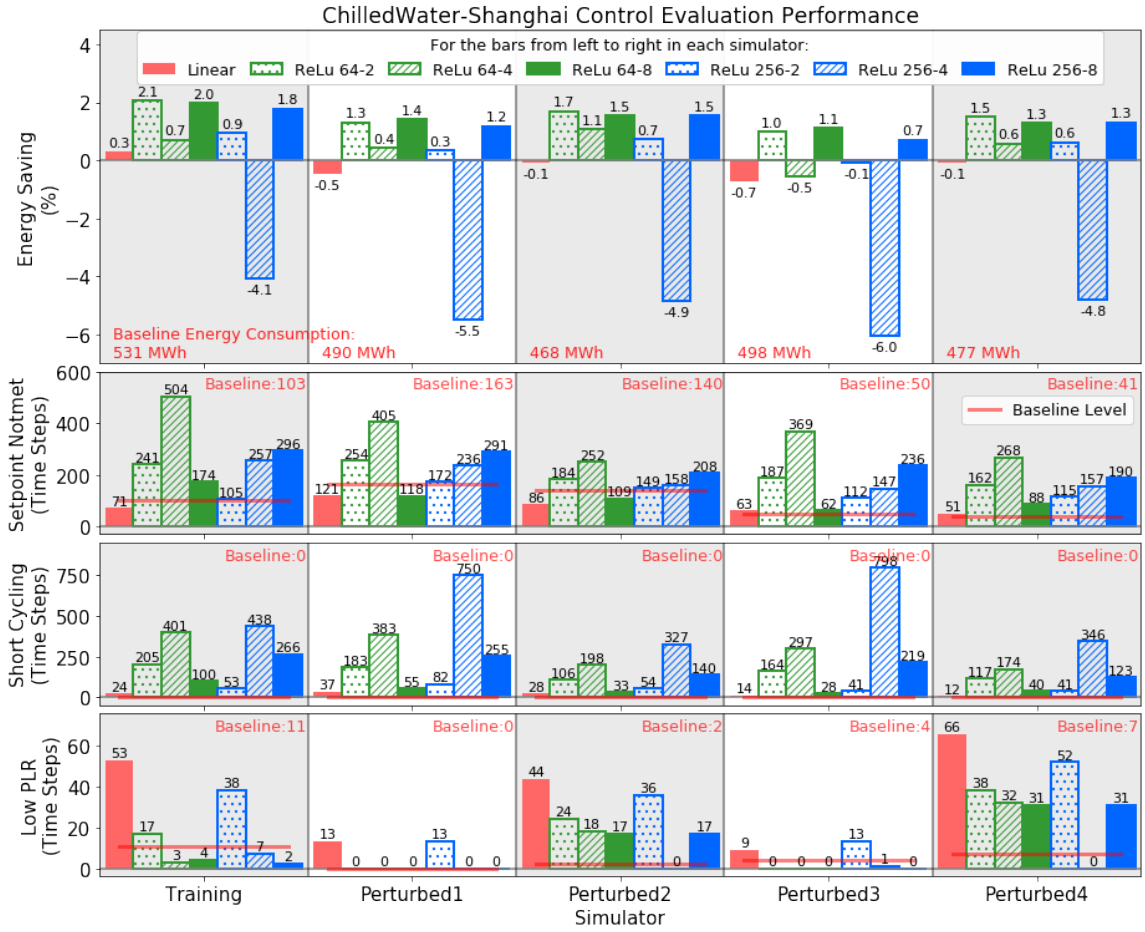

Figure 7.13: ChilledWater: Control Performance in Shanghai Climate (the results of each neural network model are from the best-performing learning rate)

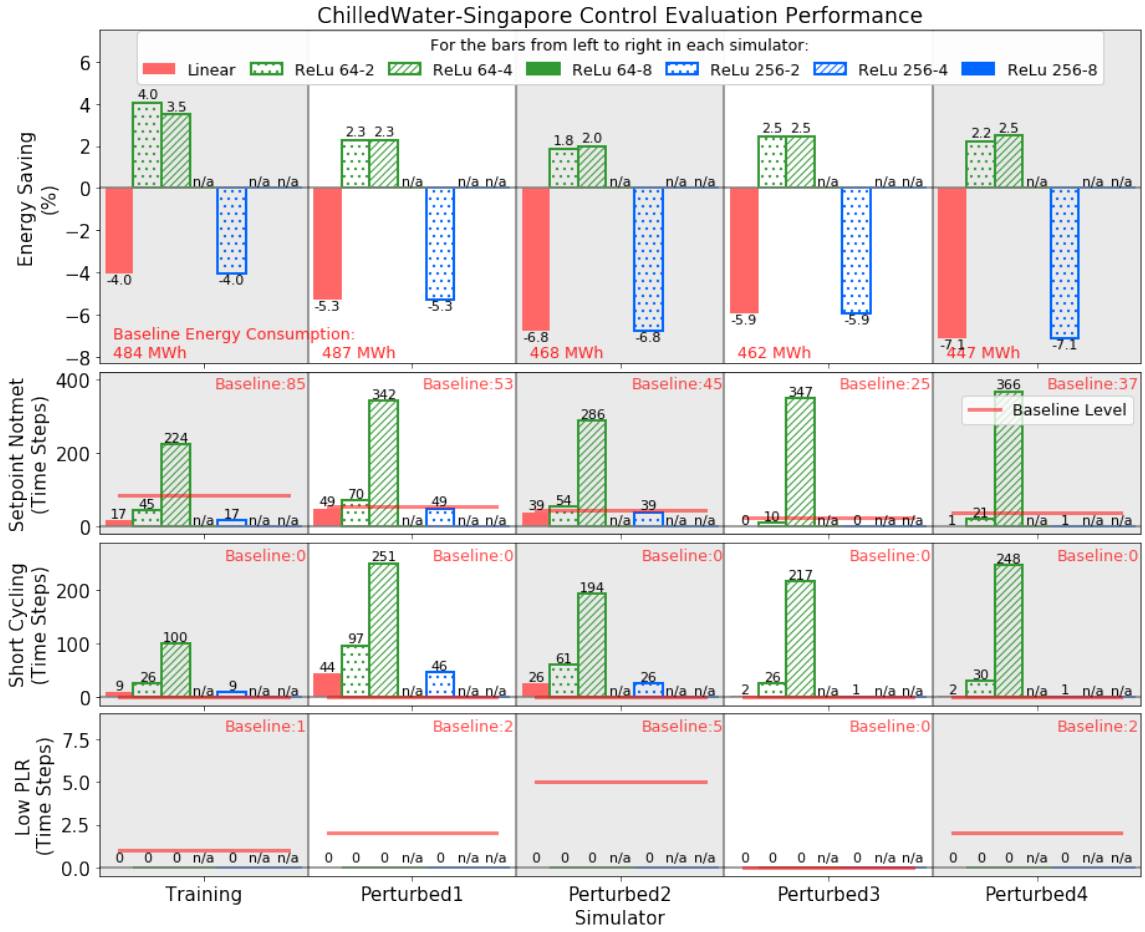

Figure 7.14: ChilledWater: Control Performance in Singapore Climate (the results of each neural network model are from the best-performing learning rate)

7.5 Summary and Discussion

This chapter presents the experiments related to ChilledWater, a three-chiller primary system. This system delivers chilled water at a predefined temperature setpoint for different cooling demands. The energy models of this system do not contain any "building" models. Instead, the cooling demand profiles are provided during the simulation. The control framework is applied to find energy-efficient control strategies for the on/off of each chiller. There are four experiment scenarios for this system, one for each climate zone. For each experiment scenario, seven neural network models and six learning rates are tuned.

The challenge of controlling this system is the stringent operational constraints of operating a chiller. Firstly, each chiller should keep its on/off status for at least 20 minutes. Secondly, the partial load ratio (PLR) of each chiller must be higher than a low limit. Failure to meet the above constraints may shorten the lifecycle of a chiller or even cause mechanical damages. Also, the system must supply chilled water at a predefined setpoint. The above operational constraints are incorporated into the reward function, i.e., the reward function output is negatively proportional to the system energy consumption if all the operational constraints are met; if one of the constraints is violated, the reward function outputs zero.

The convergence results are firstly presented. It is found that the linear and shallow neural network models are more robust to the different learning rates for convergence. This supports the thesis's hypotheses. However, the training evaluation histories have strange behaviors. The total evaluation reward quickly increases to a high level and stays there with little changes. This may be because the RL agents are stuck at some local optimal regions. Besides, the small learning rates are more favorable for convergence than the large learning rates. This result is the same as the VAVCooling experiments.

The control performance of the RL control policies is evaluated in the training simulator and four perturbed simulators. The four perturbed simulators are different in weather conditions and cooling demand profiles. For the training control performance, the energy savings are limited (2%-4%) and the operational constraints are violated for a significant number of times (several tens to several hundreds of times). These results do not support the hypotheses of the thesis that the proposed control framework saves HVAC energy consumption and meets operational constraints. The strange training evaluation histories and the poor control performance indicate that, the reinforcement learning agents may be stuck at some local optimal regions. This may be caused by the over-

complicated reward function that incorporates three stringent operational constraints. The control performance in the perturbed simulators is similar to that in the training simulator, which means the trained control polices are tolerant of the changes in weather conditions and cooling demand profiles.

Even though the trained control policies violate the operational constraints for several tens to several hundreds of times, it may not as severe as it appears. Firstly, one simulation episode has about 13,000 time steps. The operational constraint violations only occur in a small proportion of the total simulation time steps. Secondly, the violations on the supply water temperature setpoint are comparable or much less than the baseline. This means the system can deliver the chilled water with a better quality. Thirdly, the shortcycling in this thesis means a chiller cycles at a 10-minute interval. It indeed violates the 20-minute cycling constraint, but it is not a critical violation because the 20-minute constraint is determined based on experiences. Last, many modern chillers have internal safe mechanisms to prevent the potential mechanical damages caused by the low-PLR. Hence, the consequence of the low-PLR violations is also not critical for modern chillers. Nevertheless, the control performance for ChilledWater is not comparable with the other HVAC system types. Future work should develop a better method to incorporate the operational constraints. For example, a separate second level controller can be used to incorporate action-based operational constraints (e.g., shortcycling, low-PLR).

The effects of the neural network model complexity on the control performance are also shown in this chapter. It is found that there is not a clear relationship between the neural network model architecture and the control performance. The best-performing neural network model is different in different scenarios. This result does not support the hypotheses that a complex neural network model can achieve better control performance than a simple neural network model. However, it should be noted that results are from a limited number of experiments, so the conclusion cannot be generalized.

Chapter 8

Case Study: Real-life Deployment of the Control Framework

This chapter presents a real-life deployment case study of the proposed control framework. The control framework helps a real-life radiant heating system to save 16.7% heating demand, which partially demonstrates its practical feasibility and effectiveness.

Figure 8.1: The Intelligent Workplace (IW)

Figure 8.2: Hot Water Pipes Integrated on Window Mullions

8.1 Case Study Building

The case study building, the Intelligent Workplace (IW, shown in Figure 8.1), is a one-level 600 m^2 office building in Pittsburgh, PA, USA. It was built in 1997 on the roof of an existing building. The building has about 15 regular occupants and a 30-person conference room. The major heating system of IW is a novel water-based radiant heating system called "Mullion" system. It integrates hot water pipes with window mullions, as shown in Figure 8.2.

The general configurations and the baseline rule-based control strategy of the system are shown in Figure 8.3. With a constant hot water flow rate, the Mullion system adjusts its supply hot water temperature to respond to different indoor heating demands. A proportional-integral-derivative (PID) feedback controller (PID1) calculates the Mullion supply water temperature setpoint (SP2) based on the error between the IW average indoor air temperature (T1) and its setpoint (SP1). Then, another PID controller (PID2) adjusts the open state of a mixture valve based on the error

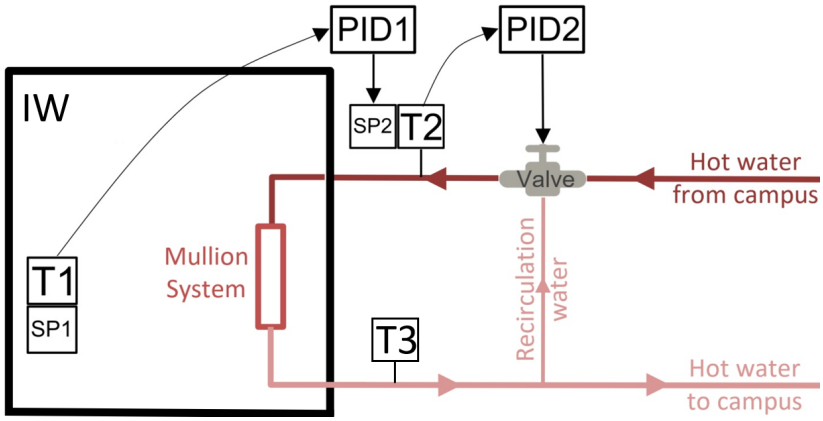

Figure 8.3: The Existing Control Principle of the Heating System in IW

between the Mullion supply water temperature (T2) and its setpoint (SP2). The open state of the mixture valve determines the mixture ratio between the hot water from the campus and the recirculation water, so the Mullion supply water temperature (T2) can be changed. Besides, the control logic shuts off the flow of the hot water from the campus when the outdoor air temperature is below 10 °C.

Since the hot water is from a district heating system of the campus, the facility manager of IW uses the system's heating demand as the energy metric, which is calculated by:

$$Q_{Mull} = C_p m(T2 - T3), \tag{8.1}$$

where C_p is the specific heat of water at constant pressure, m is the mass flow rate of the supply water of the Mullion system. Note that this equation does not consider the system transient behaviors so its output may not be accurate when there is a sudden change in the system operation (e.g., when the supply water temperature suddenly increases). This may cause a large variability in the heating demand data.

8.2 Control Objectives and Control Variable

This case study aims to develop a control policy to reduce the Mullion system heating demand and maintain an acceptable overall thermal comfort quality. In the RL training, the thermal comfort metric is the calculated predicted percentage of dissatisfied (PPD) based on Fanger's model (Fanger, 1970). Even though PPD may not represent the actual thermal comfort profile of the IW occupants, it is still used for the training because of the unavailability of the thermal comfort data. In the deployment, the overall thermal comfort quality is obtained based on the real-time feedback from the IW occupants.

The control variable is the Mullion system supply water temperature setpoint, which is SP2 as shown in Figure 8.3.

8.3 Deployment Procedure

An overview of the deployment procedure is illustrated in Figure 8.4, which includes four steps:

1. Building energy modeling: A EnergyPlus model is first created. As a building built over 20 years ago, IW does not have a design-stage BEM available for use.
2. Model calibration: The BEM built in the previous step is calibrated using the observed data from the actual system operation. This is to ensure the BEM can accurately model the thermal and energy behaviors of the IW system.
3. RL training via the proposed control framework: The calibrated BEM is used to train an RL agent off-line to develop an energy-efficient control policy for the target system.
4. Deployment: The trained RL control policy is deployed in the building automation system to generate control signals for the target HVAC system in real-time.

8.3.1 Building Energy Modeling

The IW building envelope, thermal zones and the heating system are modeled in EnergyPlus. However, the Mullion system cannot be directly modeled because of its uniqueness. As a workaround,"low

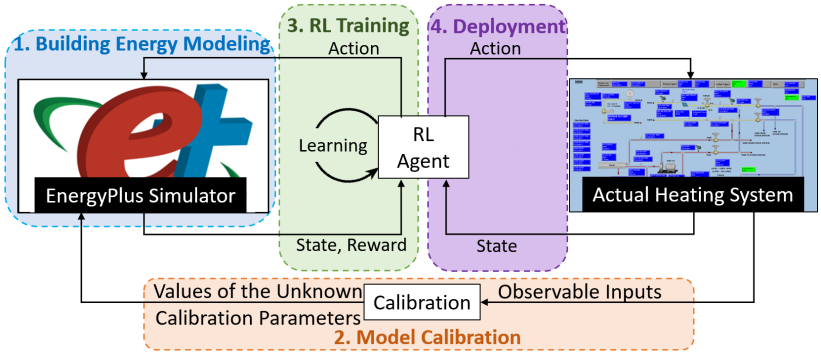

Figure 8.4: Deployment Procedure of the IW Case Study

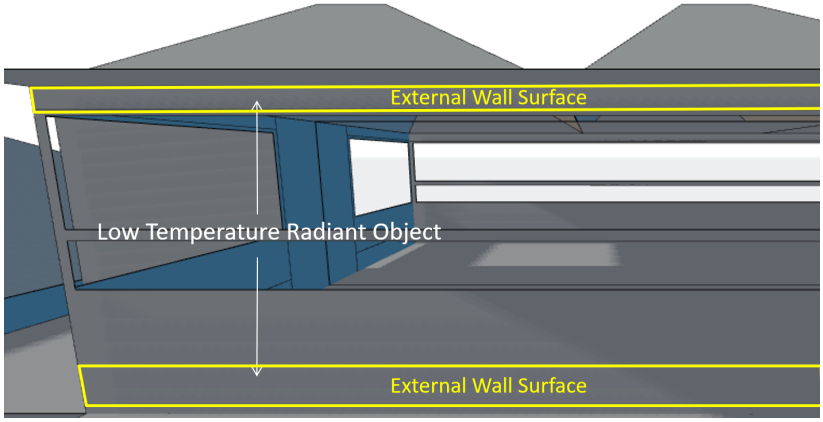

Figure 8.5: Mullion System Modeling in EnergyPlus

temperature radiant" object of EnergyPlus is selected to model this system. However, it should be noted that the "low temperature radiant" object is designed for modeling the radiant surfaces with heavy thermal mass, such as concrete underfloor heating.

The Mullion system modeled in EnergyPlus is schematically shown in Figure 8.5. The top and bottom surfaces of the external walls are modeled as the "low temperature radiant" objects of EnergyPlus (named "Mullion radiant surface" in the later sections). Figure 8.6 shows the cross-section of the Mullion radiant surface, where the internal source is an abstraction of the hot water pipes in EnergyPlus simulation. The location of the Mullion radiant surfaces (in this case, the top and the bottom of the external wall) will not affect simulation results because EnergyPlus (and most other BEM simulation engines) assumes well-mixed air and 1-D heat conduction across envelopes.

As a workaround for the Mullion system modeling, the characteristics of the Mullion radiant surface is significantly different from the actual Mullion system. The modeling parameters related to the Mullion radiant surface (such as the radiant surface area and insulation R-value) cannot

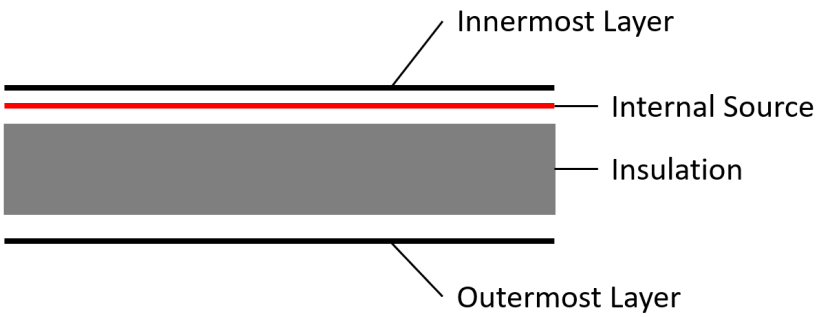

Figure 8.6: Cross-section of the Mullion Radiant Surface in the EnergyPlus Model

be determined directly. These parameters will be found in the calibration section to ensure the simulated thermal behaviors match the actual observation.

8.3.2 Model Calibration

The EnergyPlus model is calibrated in this section to ensure the model can accurately predict the thermal and energy behaviors of the actual system. More specifically, the calibration objective is to minimize the gap between the simulated and observed Mullion heating demand and average indoor air temperature.

Calibration Parameter Types and Calibration Methods The EnergyPlus model consists of two types of parameters to be calibrated, including dynamic schedules and static properties:

- The dynamic schedules, such as occupancy schedules and plug-load schedules, are manually calibrated based on the author's observations on the occupancy pattern of IW. Since IW is a small office/classroom building with only 15 regular occupants, manual calibration is sufficient in this case.
- The static properties of the model, such as infiltration rate and wall insulation level, are automatically calibrated using Bayesian calibration (i.e., a statistical method for computer model calibration).

This section mainly presents the process of Bayesian calibration for the static properties.

Selection of the Static Calibration Parameters The EnergyPlus model consists of a large number of static properties. Initially, 12 calibration parameters are manually selected based on

Table 8.1: Selected Four Calibration Parameters for the IW EnergyPlus Model

Parameter	Range
Insulation (Polyisocyanates) thickness of the Mullion radiant surfaces†	1 mm to 5 mm
Total area of the Mullion radiant surfaces†	6.7% to 26.7% of the external wall
Internal mass area*	200 to 1000 m^2/zone
Infiltration rate	0.01 to 0.30 ACH

Note:†The Mullion system is modeled as the "low temperature radiant" surfaces in the EnergyPlus model. *Internal mass is modeled in the EnergyPlus model as 5 cm thick concrete.

the authors' judgment on their effects on the modeling accuracy. They include: (1) insulation thickness of the pitched roof, (2) insulation thickness of the flat roof, (3) insulation thickness of the external wall, (4) infiltration rate, (5) U-value of the external window, (6) solar heat gain coefficient (SHGC) of the external window, (7) heat conductivity of the innermost layer of the Mullion radiant surface, (8) insulation thickness of the Mullion radiant surface, (9) total area of the Mullion radiant surfaces, (10) air mixing rate across the zones, (11) electric equipment power density, (12) internal mass area. Then, we further screen out 4 calibration parameters out of the 12 using sensitivity analysis (Morris method Morris (1991)) and manual trial-and-error tests. The selected calibration parameters are listed in Table 8.1 with their calibration ranges. The ranges are determined based on experience. Note that the four selected calibration parameters are not entirely determined based on the sensitivity analysis. With the sensitivity analysis results as a reference, a number of manual experiments are also conducted.

Datasets for Bayesian Calibration The case study focuses on the control for heating seasons. Hence, the calibration is conducted using the three-month observed data from Jan 1st, 2017 to Mar 31th, 2017. The calibrated model is then evaluated using the one-month observed data from Nov 1st, 2017 to Nov 30th, 2017. The time resolution of all the datasets is 5 minute. Table 8.2 shows the items contained in the datasets.

Implementation of Bayesian Calibration The case study adopts the method proposed by Chong and Menberg (2018) for Bayesian calibration. This section briefly describes the theoretical basis of Bayesian calibration and the implementation details can be found in their original paper.

Table 8.2: Items in the Datasets for Bayesian Calibration of the IW EnergyPlus Model

Item	Type
Outdoor air temperature (°C)	Inputs to the EnergyPlus model
Outdoor air relative humidity (%)	
Direct solar radiation (W/m^2)	
Diffuse solar radiation (W/m^2)	
Wind speed (m/s)	
Wind direction (degree from North)	
Mullion supply water temperature (T2) (°C)	
Mullion supply water mass flow rate (kg/s)	
Average indoor air temperature (°C)	Calibration objectives
Mullion system heating demand (kW)	

The statistical formulation of Bayesian calibration is

$$y(x) = \zeta(x) + \epsilon(x) = \eta(x, t^*) + \delta(x) + \epsilon(x), \tag{8.2a}$$

$$\text{s.t. } \delta(x) = 0, \tag{8.2b}$$

where y, ζ, η are the observed/true/simulated building performance behavior (i.e., calibration objectives such as HVAC energy consumption) respectively, x is the observable input parameter (i.e., the parameters that can be observed but cannot be manipulated such as weather conditions), t is the calibration parameter with the unknown true value t^* (i.e., the manipulable BEM parameters such as infiltration rate), δ is a discrepancy term to correct any model inadequacy, ϵ is the observation error. Note that in Equation (8.2b), $\delta(x)$ is forced to be zero because we assume the BEM can adequately model the target heating system.

If we assume the distribution of ϵ is Gaussian, we can write the posterior distribution of t given y as follows based on Bayes' theorem:

$$P(t|y) \propto L\big(y|\eta(t)\big) \times P(t) \tag{8.3}$$

where P represents a probability distribution and L represents the likelihood function. Markov chain Monte Carlo (MCMC) sampling is then used to numerically obtain the posterior distribution $P(t|y)$ in Equation (8.3). The modes of the distribution are regarded as the calibrated parameter values that are fed back into the EnergyPlus model.

The computation time of Bayesian calibration increases exponentially with the size of the cali-

bration dataset, i.e., the number of data entries of $y(x)$ and $\eta(x,t)$. To adapt Bayesian calibration for sub-hourly resolution data, we adopt the method proposed in Chong et al. (2017) to down-sample the original calibration dataset to a smaller one.

One limitation of Chong and Menberg (2018)'s method is that it is designed for single-objective calibration. However, the EnergyPlus model needs to be calibrated for both heating demand and indoor air temperature. Hence, a convex combination method (Zhang et al., 2018b) is proposed to combine multiple building performance metrics into one. As a result, the y and η in Equation (8.2) are obtained by the convex combination of two building performance metrics, i.e.,

$$\begin{cases} y = \mu_1 y_1 + \mu_2 y_2, \\ \eta = \mu_1 \eta_1 + \mu_2 \eta_2, \end{cases} \tag{8.4a}$$

$$\text{s.t. } \mu_1 + \mu_2 = 1, \tag{8.4b}$$

$$\mu_1, \mu_2 \geq 0, \tag{8.4c}$$

where μ_1 and μ_2 are the convex combination weights, y_1 and y_2 are the observed average indoor air temperature and observed Mullion system heating demand, and η_1 and η_2 are the simulated average indoor air temperature and simulated Mullion system heating demand. Different combinations of μ_1 and μ_2 are tested.

Calibration Results Table 8.3 shows the modeling errors on both calibration dataset and evaluation dataset. The evaluation dataset is an unseen dataset that is different in time period from the calibration dataset. In the table, we use normalized mean bias error (NMBE) and coefficient of variation of the root mean square error (CVRMSE) as the modeling error metrics. They are recommended error metrics in ASHRAE Guideline 14 (ASHRAE, 2014). As a reference, ASHRAE Guideline 14 recommends the hourly NMBE and CVRMSE of a whole building energy model should be less than 10% and 30% respectively.

The table shows that, after calibration, the average indoor air temperature achieves less than 5% errors in both calibration and evaluation datasets. The hourly CVRMSE of the heating demand is higher than 30% but the daily CVRMSE is still relatively small. Figure 8.7 shows the hourly and 5-min comparison between the observed and simulated heating demand using the evaluation dataset. It can be seen that, even though the calibrated EnergyPlus model can capture the overall trend of the heating demand, it fails to do so for some extremes (e.g., the sudden jumps and falls

Table 8.3: Modeling Errors after Bayesian Calibration of the IW EnergyPlus Model (IAT: indoor air temperature)

Objective	**Metric**	**Calibration Dataset**	
		Calibration	Evaluation
Average IAT (C)	5-min NMBE	0.52%	-1.01%
	5-min CVRMSE	4.82%	2.65%
Heating Demand (kWh)	Hourly NMBE	0.43%	-1.66%
	Hourly CVRMSE	35.93%	59.91%
	Daily CVRMSE	10.46%	12.95%

of the heating demand). As the time interval of the data increases, the simulated data can better match the observations. The phenomenon can be probably explained by the following two reasons. Firstly, the observed heating demand has over-estimated variability because it is calculated using a steady-state specific heat equation without considering the system transient behaviors; secondly, the current simulation engine of EnergyPlus cannot adequately model the Mullion system.

8.3.3 RL Training

As stated in section 8.2, the control objectives are to reduce the heating demand consumption and maintain an acceptable indoor thermal comfort level. In the training, we use PPD as the metric for indoor thermal comfort.

State Design The state follows the structure defined in Equation (2.19) and each *ob* consists of the 15 items as shown in Table 8.4.

Note that all the observation items can be easily accessed through the building automation system (BAS) of IW, except *IW average PPD* which will be replaced by the occupants' real-time thermal comfort feedback during the deployment.

Action Design The action space is the Mullion system supply water temperature setpoint (°C). The discrete action space includes the following actions: $\{\text{turn-off heating}, 20°C, 25°C, ..., 65°C\}$.

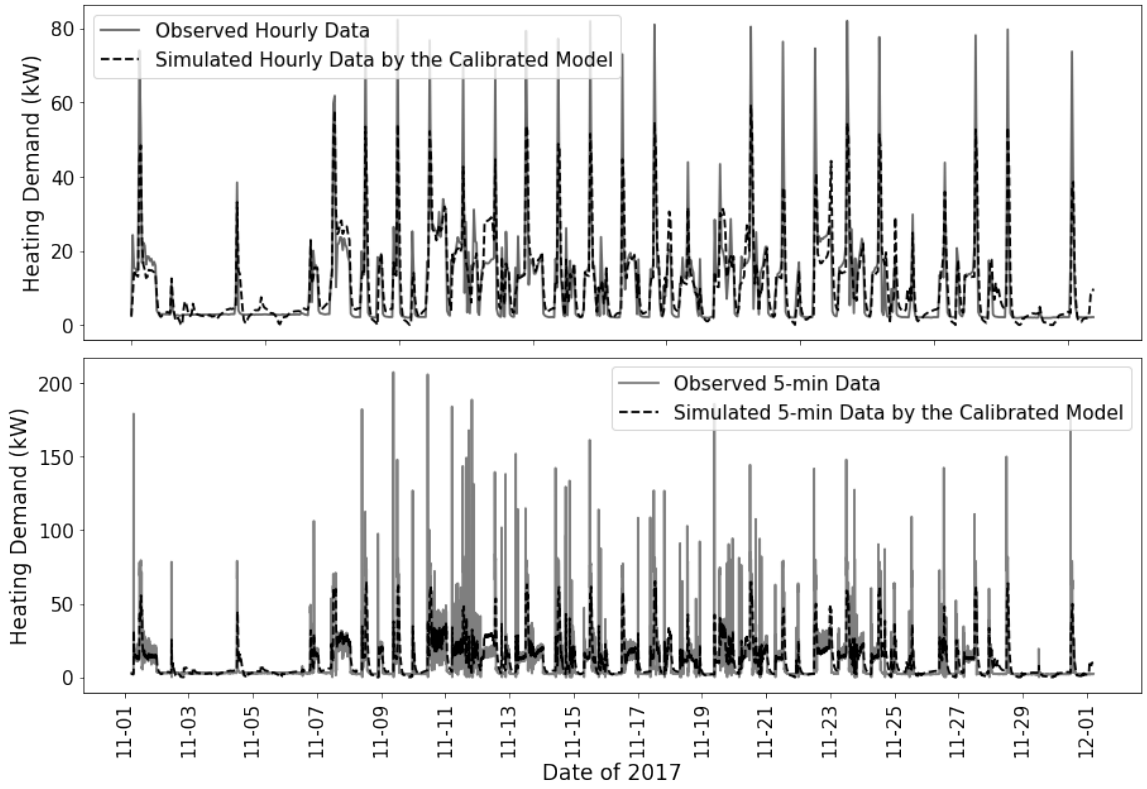

Figure 8.7: Hourly and 5-min Comparison between the Simulated (after Bayesian Calibration) and Observed Heating Demand in the Evaluation Dataset

Table 8.4: Observation Vector in the State for the RL Training of the IW Case Study

No.	Item	ob_{min}	ob_{max}
1	Day of the week	0	6
2	Hour of the day	0	23
3	Outdoor air temperature (°C)	-13	26
4	Outdoor air relative humidity (%)	0	100
5	Wind speed (m/s)	0	11
6	Wind direction (degree from north)	0	360
7	Diffuse solar radiation (W/m^2)	0	378
8	Direct solar radiation (W/m^2)	0	1000
9	IW steam heat exchanger enable setpoint (°C)*	-30	30
10	IW average PPD	0	100
11	Mullion system supply water temperature setpoint (°C)	20	65
12	IW average indoor air temperature (IAT, °C)	18	25
13	IAT setpoint (°C)	18	25
14	IW occupancy mode flag$^\diamond$	0	1
15	IW average heating demand since last observation (kW)	0	85

Note: *The outdoor air temperature setpoint below which the IW steam heat exchanger will be enabled. $\diamond$The scheduled occupancy mode flag (the flag is 1 for the period 7:00 AM - 7:00 PM of weekdays and 8:00 AM - 6:00 PM of weekends).

Reward Design The reward function is defined in Equation (8.5), that is:

$$R_t = -\begin{cases} \left[\tau * \left([PPD_t - 0.1]^+ * \rho\right)^2 + \beta * Q_{Mull,t}\right]_0^1 |_{Occp_t=1} \\ \left[\tau * [SP_{thres,t} - T1_t]^+ * \lambda + \beta * Q_{Mull,t}\right]_0^1 |_{Occp_t=0}, \end{cases} \tag{8.5}$$

where subscript t is a control time step, Q_{Mull} is the average Mullion system heating demand since the last control time step (kW), $T1$ is the average indoor air temperature, $Occp$ is the occupancy mode flag, and $\tau, \beta, \rho, \lambda$, SP_{thres} are tunable hyperparameters to control the relative weight between the heating demand and indoor thermal comfort.

Hyperparameters The RL training hyperparameters are summarized in Table 8.5. Note that the hyperparameters in the table are not sufficiently tuned. An RL agent is trained using the calibrated IW EnergyPlus model with Pittsburgh TMY3 weather data, and one simulation episode lasts from Jan 1st to Mar 31th.

Table 8.5: RL Training Hyperparameters for the IW Case Study

Item	Value	Item	Value
Neural network width$^{\diamond}$	512	A3C local agent number	16
Neural network depth$^{\diamond}$	4	Reward discount factor	0.99
Nonlinear activation$^{\diamond}$	ReLu	Entropy weight (κ)*	1.0,0.1,0.01,0.001
Optimizer	RMSProp	$\boldsymbol{\kappa}$ decay steps*	[2M, 4M, 6M]
RMSProp decay rate	0.9	Value loss weight	0.5
RMSProp momentum	0.0	RL total interaction times	10 millions
RMSProp epsilon	$1e^{-10}$	Learning batch size	5
Learning rate	0.0001	State history window	3
Gradient clip method	By the L2-norm	$\boldsymbol{\tau}$ in the reward	1.0
Gradient clip threshold	5.0	$\boldsymbol{\beta}$ in the reward	2.5
Simulation time step^{+}	5 min	$\boldsymbol{\rho}$ in the reward	10
Control time step^{+}	15 min	$\boldsymbol{\lambda}$ in the reward	5.0

Note: * The RL training uses an entropy term to encourage the exploration level of an RL agent. κ is a staircase decayed constant to control the entropy, and the decay happens at the 2M, 4M and 6M interaction time steps, e.g., from step 0 to 2M (non-inclusive), $\kappa = 1.0$, from step 2M to 4M, $\kappa = 0.1$, etc; $^{\diamond}$The neural network is the shared fully connected feed-forward neural network in Figure 2.3. $^{+}$The control time step is sparser than the simulation time step because the slow-response Mullion system needs long time to respond to a new control action.

Table 8.6: IW Case Study: Simulated Performance of the Selected RL Control Policy

	Total heating Demand (kWh)	PPD Mean(%)	PPD Std(%)
Baseline Rule-based Control	43709	9.46	5.59
RL Control Policy	37131	11.71	3.76

Simulated Control Performance The trained RL control policy is then evaluated in an EnergyPlus simulator with AMY2017 weather data. One simulation episode is Jan 1st-Mar 31th of 2017. The results are shown in Table 8.6. In the simulator, the selected RL agent saves 15% heating demand but the mean PPD is increased (with decreased standard deviation) compared to the baseline control.

8.3.4 Deployment

The trained RL control policy is deployed in the IW system to control the Mullion supply water temperature setpoint. The deployment experiment lasts for 78 days, from Feb 6th to Apr 24th of 2018. The deployment architecture is shown in Figure 8.8. The trained RL agent obtains sensory

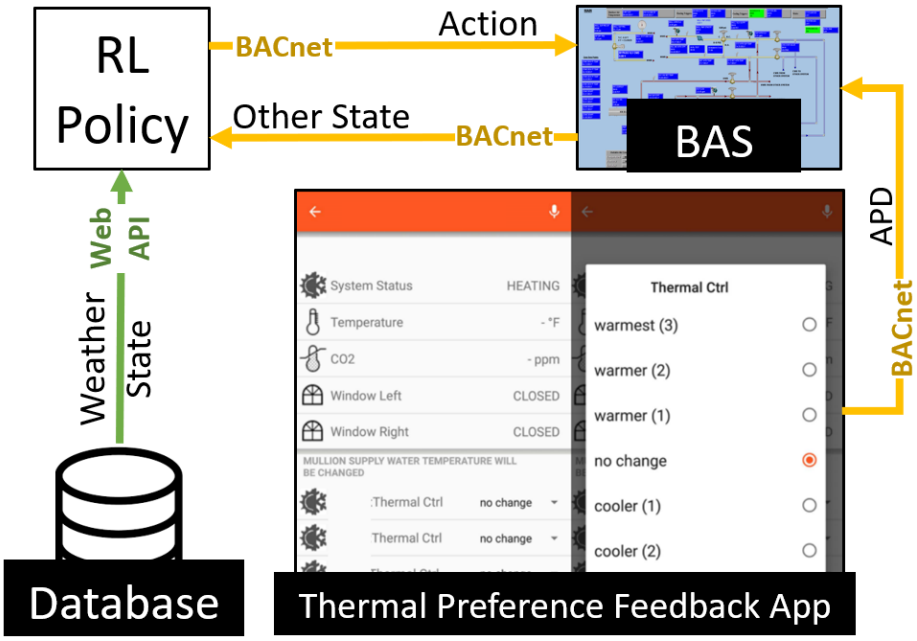

Figure 8.8: Deployment Architecture of the RL Control Policy in the Intelligent Workplace

information from the system's BAS and a web-based database, and writes control actions back to the system through BACnet protocol. In the deployment, the state design is the same as the training, except the PPD is replaced by the real-time thermal comfort feedback from the IW occupants. Details of the implementation setup and the thermal preference feedback system can be found in Zhang and Lam (2018).

8.4 Normalized Energy Efficiency Analysis

8.4.1 Method Description

The energy consumption of the Mullion system is influenced by multiple factors, such as control strategy, outdoor air temperature, solar radiation, indoor air temperature, etc. Hence, to evaluate the energy saving of the RL control policy compared to the old rule-based control, all energy-influencing factors should be constant. This can be easily realized in computer simulations but is difficult in real-world implementations.

This study proposes a data-driven approach to perform a normalized energy saving evaluation of a new control strategy using real-world operation data. The approach is similar to the Weather Normalized Energy analysis method in ENERGY STAR (ENERGY STAR, 2017; Kissock et al., 1998), but it is extended to include multiple energy-influencing factors, non-linear input-output relationships and stochasticity. The workflow of this approach is shown in Figure 8.9, which is divided into two parts, model fitting and sampling:

- Model fitting: This part fits a Gaussian process (GP) (Rasmussen and Williams, 2006) model using the historical data in the old rule-based control period. This GP model is treated as a *baseline daily heating demand* model. The GP model's form is:

$$GP(\mathbf{x}) = \mathcal{N}(\mu, \sigma), \tag{8.6}$$

  where $\mathbf{x}$ is the independent variables, μ and σ are the mean and standard deviation of the daily heating demand.

  The GP model fitting follows a standard machine learning process, including cross-validation for feature selection and model testing. If the testing accuracy (based on the mean of the prediction of the GP model) passes a predefined threshold (i.e., $R^2 \geq 0.9$), the GP model can be used for the sampling part.

- Sampling: This part uses the fitted GP model to create a sampling distribution of the *baseline total heating demand at the new control period*. Each sample of the baseline total heating

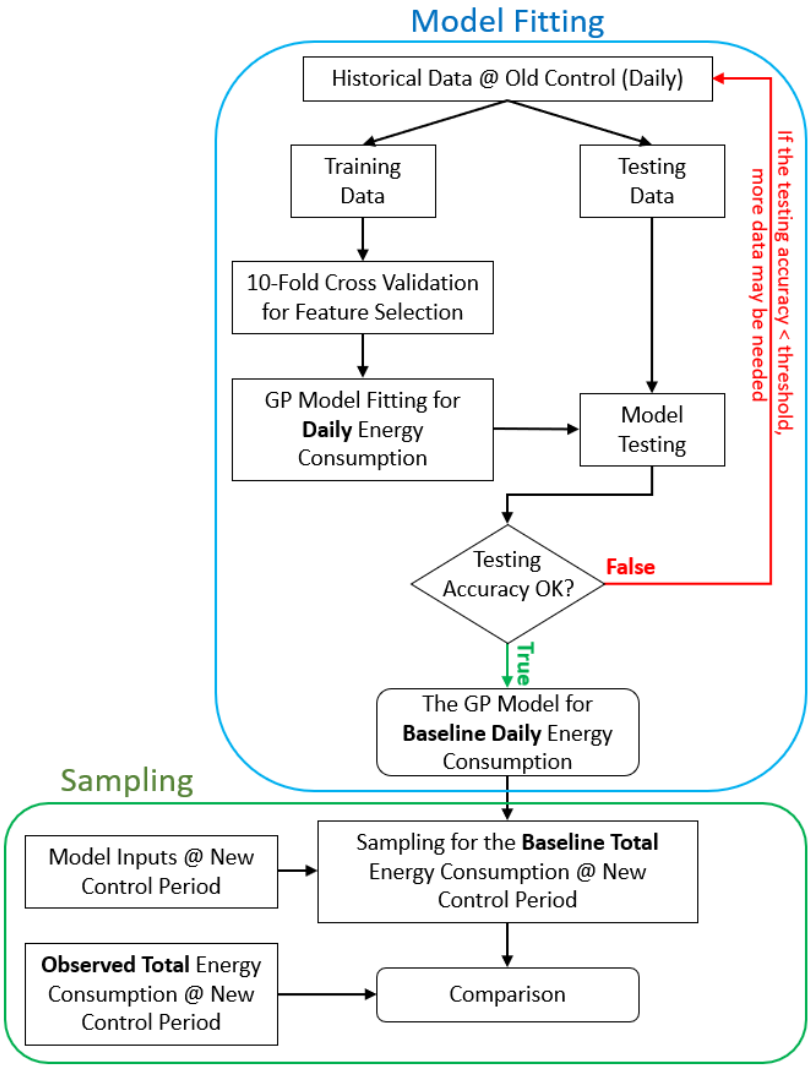

Figure 8.9: Workflow of the Normalized Energy Saving Performance Evaluation Approach

demand ($E_{basetotal}$) is generated using the following equation:

$$E_{basetotal,j} = \sum_{i}^{m} \big(E_{basedaily,i} \sim GP(\mathbf{x}_i)\big), \quad j \in \{1, 2, ..., n\}, \tag{8.7}$$

where $E_{basedaily}$ is the *baseline daily heating demand at the new control period* sampled from the fitted GP model, i represents a day in the new control period of m days, j represents a sample of the n baseline samples.

After sampling for n times using Equation (8.7), a set of n values (i.e., $\{E_{basetotal,1}, E_{basetotal,2}, ..., E_{basetotal,n}\}$) is obtained representing the sampling distribution of the *baseline total heating demand at the new control period*. Thus, a distribution function can be approximated, and it can be compared with the *observed total heating demand at the new control period* to create a statistically solid energy saving conclusion.

8.4.2 Results

The above method is applied to the IW case study. The GP baseline model is built based on the data of 357 days in the old rule-based control period. The baseline model has the inputs ($\mathbf{x}$ in Equation (8.6)) including outdoor air temperature, global solar radiation, and indoor air temperature, which are selected by a 10-fold cross validation process. The GP baseline model is used to generate 10000 samples of the daily heating demand over the RL control deployment period (Feb 6th-Apr 24th, 2019). Figure 8.10 shows 50 random examples of the 10000 samples. Each sample represent a possible profile of the daily heating demand if the old rule-based control had still been used over the RL control deployment period.

The total heating demand of the 10000 samples is calculated based on Equation (8.7), so a sampling distribution is generated for the baseline total heating demand over the RL control deployment period. The baseline samples of the total heating demand are shown in Figure 8.11. It can be seen that the samples are generally shaped like a normal distribution but noises exist. Kernel Density Estimation (KDE) is then applied with Gaussian kernel to generate an approximated probability density function (PDF). The PDF represents the distribution of the baseline total heating demand. To make the energy comparison statistically solid, we select the baseline heating demand at around the 5th percentile of the PDF, which is 28940 kWh at the 4.99th percentile as shown in Figure 8.11. This indicates that the baseline heating demand can be higher than this value with more than 95%

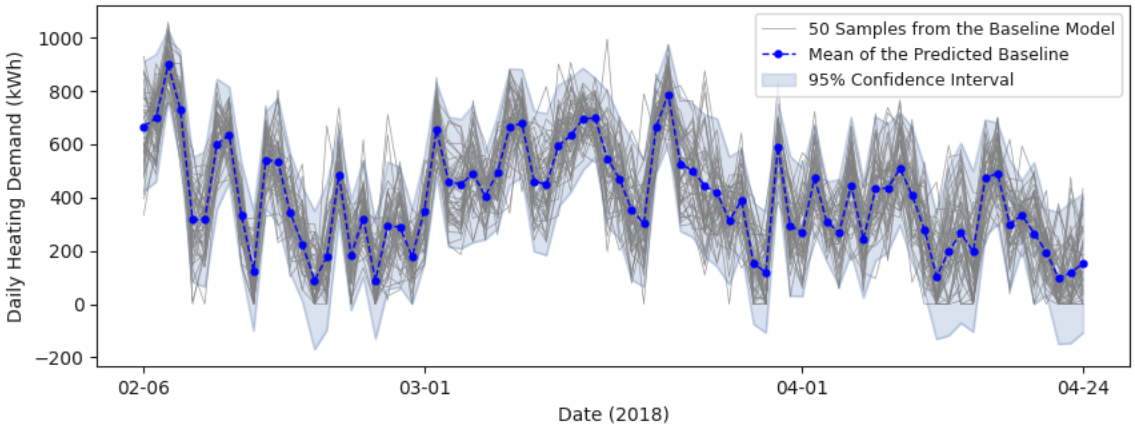

Figure 8.10: Baseline Daily Heating Demand Samples generated from the GP Model (50 out of 10000 samples are shown)

Table 8.7: Comparison between the Observed Total Heating Demand of the RL Control Policy and the GP Baseline Total Heating Demand at the 4.99th Percentile

DRL Observed	**GP Baseline**	**Save**
24103 kWh	28940 kWh	16.7%

probability. By comparing this value with the observed total heating demand, it is concluded that the DRL control has achieved 16.7% heating demand reduction compared to the rule-based control, as summarized in Table 8.7.

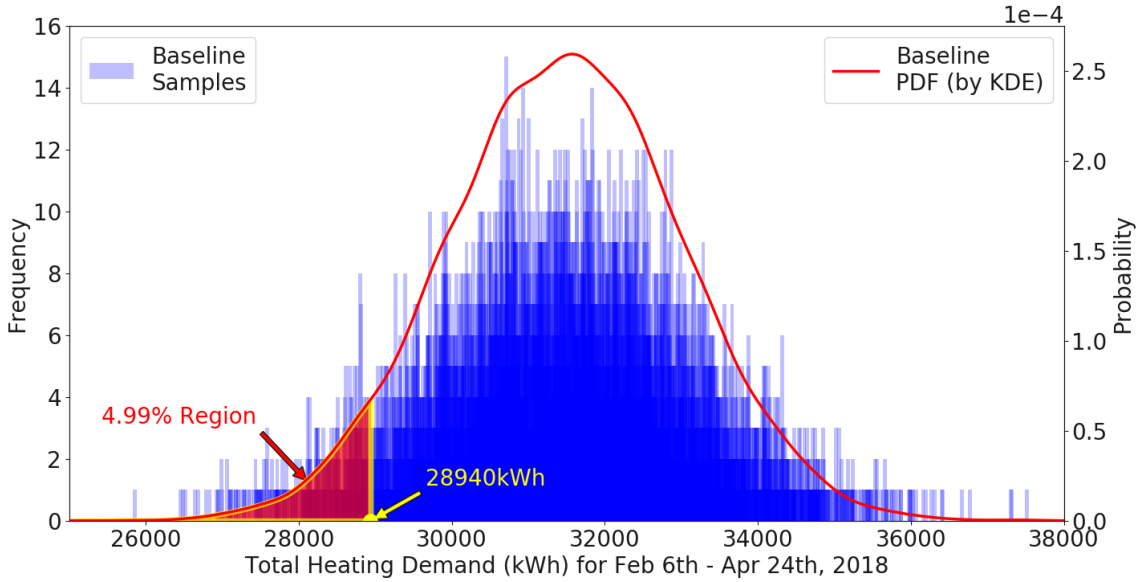

Figure 8.11: Baseline Total Heating Demand Samples and the Estimated Probability Density Function (PDF) Generated by Kernel Density Estimation (KDE) (The shaded area shows the lowest 4.99% of the distribution)

8.5 Summary and Discussion

This chapter presents a case study of implementing and deploying the proposed control framework in an actual radiant heating system. The case study consists of four steps, including EnergyPlus modeling, model calibration, RL training, and deployment. A normalized energy saving analysis method is also proposed to fairly evaluate the energy saving potential of the RL control policy using the 78-day real-life deployment data. It is found that, compared to the old rule-based control strategy, the RL control policy has saved 16.7% heating demand.

EnergyPlus model calibration is an important step for the real-life deployment. An EnergyPlus model must accurately predict the actual system's thermal and energy behaviors. In this case study, both dynamic schedules and static properties of the model are calibrated. After calibration, the calibrated EnergyPlus model can accurately predict the 5-min average indoor air temperature with less than 5% error, and can predict the daily heating demand with around 10% error.

The calibrated EnergyPlus model with TMY3 weather data is then used as a simulator to train an RL agent offline. The simulated control performance shows that, compared to the rule-based control, the RL control policy saves 15% heating demand with a similar level of thermal comfort.

However, the hyperparameters, such as the neural network architecture, reward function design, the length of the history in the state, etc., are not extensively tuned. More tuning experiments may further improve the control performance.

The trained control policy is then deployed in the actual heating system through the building automation system. The deployment experiment lasts from Feb 6th-Apr 24th of 2018. A baseline heating demand distribution is generated by using a data-driven normalized energy saving analysis method. By comparing with the generated baseline, the RL control policy has saved 16.7% heating demand with more than 95% possibility.

An interesting finding is that, the calibrated EnergyPlus model has a relatively large modeling error for the heating demand, but the control policy trained via this model still shows obvious energy saving in the real-life deployment experiment. This means the trained control policy has some levels of versatility to tolerate differences between the simulator and the reality. This result can be related to the previous simulation experiments where the trained control policies have demonstrated the tolerance for the variations of weather conditions and internal load schedules. It is necessary to further investigate the versatility of an RL-trained control policy in future work.

The successful real-life implementation case study partially proves the practical feasibility and effectiveness of the control framework. It also demonstrates a potential practical implementation process, including BEM modeling, model calibration, RL training, and deployment. However, the deployment period of the case study is still relatively short compared to the lifecycle of a building. A long-term deployment may require a method or framework to adapt to changing building characteristics, which is not considered in this case study.

Chapter 9

Usage Guidelines for the Control Framework

This chapter presents the usage guidelines for the control framework based on the results of the simulation experiments and the deployment case study.

9.1 For Offline Training

Avoid misuse Artificial intelligence is not magic. It is invented by humans, designed by humans, and used by humans. Hence, any artificial intelligence technologies, including the one used in this control framework, must be properly used with the understanding of their working principles. Misuse of the control framework may lead to unexpected results, such as limited energy efficiency improvements and violated operational constraints. Common misuse behaviors include:

- Improper design of the state, action and reward in the control framework;
- Improper selection of the hyperparameters, such as the neural network model and learning rate.

The guidelines are only served as a basic starting point. Advanced users should further study the detailed theoretical backgrounds of this control framework (see Chapter 2) to understand the working

principles.

Non-guaranteed optimal control solution This control framework cannot guarantee that the control solution is optimal, so it is cannot be called "optimal control". Even though the derived control policy may provide significant energy efficiency improvements, it may be just one of many sub-optimal solutions. It is highly possible that, with other more advanced algorithms, an even better control policy may be found.

Control performance is scenario-dependent This control framework cannot guarantee a good control performance, i.e., HVAC energy efficiency improvement and operational constraints fulfillment. The control performance can be dramatically different for different scenarios, including different types of HVAC systems, different climates and other unlisted different characteristics of an HVAC system. In some scenarios, the control framework may deliver significant energy savings while fulfilling operational constraints; but in some other scenarios, energy savings may be limited and operational constraints may be violated. Hence, users must conduct simulation experiments (as demonstrated in Chapters 4, 5, 6 and 7) to obtain the potential control performance of a scenario.

Application Scope of the Control Framework This control framework is designed to use a whole building energy model (BEM) to develop HVAC supervisory control strategies. Thus, it is a suitable choice when users have an existing BEM and hope to extend its usage. It is also suitable if the target HVAC system has complicated configurations and dynamics, such as a multi-zone VAV system or the Mullion radiant heating system shown in Chapter 8. Traditional optimal control methods (such as MPC) are not practical for such cases because they usually require a low-order model for the complicated high-order system dynamics.

The control framework is not suitable for simple HVAC systems, or the systems whose dynamics can be well identified by low-order models. For such systems, model-based control methods (such as MPC) is a better choice because they usually have optimality guarantee. In contrast, the control framework cannot guarantee any optimal solutions.

Computational Cost of the Offline Training The computations are mainly for the following two tasks:

- EnergyPlus simulations: One instance of EnergyPlus simulation can only be performed on

one CPU. Its computation cannot be paralleled on multiple CPUs. However, this control framework uses A3C, which will fire multiple instances of EnergyPlus simulation. Therefore, multiple CPUs will still improve the overall computing speed of the offline training, but GPUs will not help.

- Gradient descent optimization: This task can be highly paralleled, so GPUs will significantly improve the computing speed of this task.

The actual computational time varies significantly on different experiment scenarios and different computers. As a rough reference, it takes 4-6 hours for an RL agent to complete 2.5M interaction steps (with 16 parallel local RL agent workers) on a high-end desktop computer without using any GPUs (the computer has a 10-core Intel Xeon W-2155 CPU with 4.5GHz turbo frequency, 32GB memory, 7200rpm SATA hard drive). Note that some scenarios may require larger interaction steps, so the computational time will also be longer.

Design the reward function The reward function is crucial for reinforcement learning (RL). It determines the learning objective of an RL agent and can significantly affect the learning convergence performance. In designing the reward function,

1. Avoid prior knowledge if possible. Users should first design a reward function with the minimum prior knowledge included. Prior knowledge means the knowledge that an RL agent possesses before it interacts with the environment (i.e., HVAC simulator). For example, "give a high reward value if the agent turns off the heating when the room is not occupied" is a reward function with strong prior knowledge. It contains specific instructions that come from a user's experience. This reward function may limit the exploration ability of an RL agent, and hence lead to sub-optimal solutions. A minimum-prior-knowledge reward function should be an objective description of the state, for example, "give a high reward value if the heating energy consumption is low and occupants (if any) feel comfortable". More formally, the

minimum-prior-knowledge reward function should be:

$$
\begin{aligned}
R_t &= 1.0 - [\beta * P_{energy} + \tau_1 * P_{constraint,1} + \tau_2 * P_{constraint,2} + ... + \tau_n * P_{constraint,n}]_0^1, \\
&where, \\
P_{energy} &= \text{Normalized energy metric at the control time step t,} \\
P_{constraint,i} &= [ActualObservation_{t,i} - Setpoint_{t,i}]^+, \text{ (For positive setpoint error constraints)} \\
P_{constraint,i} &= [Setpoint_{t,i} - ActualObservation_{t,i}]^+, \text{ (For negative setpoint error constraints)} \\
P_{constraint,i} &= OtherFunctions, \text{ (For other non-setpoint constraints)}
\end{aligned}
\tag{9.1}
$$

where β and τ are tunable hyperparameters that control the relative weight of different penalties.

This reward function simply penalizes the state with high energy consumption and high setpoint notmet (and other non-setpoint constraint violations such as shortcycling). It does not include any specific instructions but only a description of the state.

2. Balance among different control objectives. From the optimization point of view, the reinforcement learning algorithm solves a single objective optimization problem. However, most HVAC control problems have multiple objectives, such as minimizing energy consumption and fulfilling operational constraints. Hence, multiple objectives must be combined into a single reward function, as shown in Equation (9.1) which combines the energy minimization goal with other constraint-violation penalties. The relative weights of different optimization objectives are controlled by β and τ. If β is large and τ is small, the final control solution may have low energy consumption but large constraint violations, and vice versa.

   The values of β and τ cannot be determined beforehand. This is because, firstly, the effects of the two hyperparameters differ in different scenarios. Secondly, it depends on users' preference to select a pro-energy-saving or pro-constraint-fulfillment combination of the hyperparameters. Users must conduct parametric experiments for the values of β and τ, and determine the most appropriate selection based on the resulting control performance.

3. Avoid too complicated operational constraints. As discussed before, the control framework uses a single reward function to combine multiple control objectives. This design may lead to an over-complicated reward function that jeopardizes the reinforcement learning training process. An example is shown in the experiments of Chapter 7 where the reward function of a chilled water system includes three different operational constraints. The control performance of the

experiments is poor because the RL agents fail to fulfill the three constraints simultaneously.

A practical solution is to have a rule-based "safety control logic" to accommodate action-based operational constraints (e.g., shortcycling constraint), so the reward function can be simpler. This safety control logic will block "unsafe" control actions that are selected by an RL agent. For example, for the chilled water system of Chapter 7, a safety control logic can block the shortcycling control actions, so the shortcycling constraint can be removed from the reward function.

4. Add prior knowledge to the reward function if convergence is poor. In some scenarios, a simple minimum-prior-knowledge reward function may lead to poor convergence performance. For example, a reward function based on Equation (9.1) leads to a poor convergence for a slow-response radiant heating system (Zhang et al., 2019). The cause of the problems is multifold and theoretical solutions are still under study. Practically, adding prior knowledge to the reward function could improve convergence performance. Intuitively. this is because the prior knowledge reduces the exploration space of an RL agent. However, it should be noted that the prior knowledge may also lead to suboptimal control solutions.

   It is scenario-dependent about adding the right prior knowledge to the reward function. It requires a deep understanding of the dynamics of an HVAC system. A good example is shown in Chapter 6, where the heuristic functions are added to encourage low-setpoint control actions and PMV increase during occupied hours. The heuristic functions are obtained based on a thorough analysis of the system dynamics and trial-and-error experiments.

Design the state The state is a stack of current and historical observation vectors. Users need to make two decisions: 1) the items in the observation vector; 2) the length of the history in the state.

1. For the items in the observation vector

As a general rule, the observation vector (ob) should include sufficient sensory information to determine a control action to maximize the cumulative future reward value (R). Users can view it as a feature selection problem for fitting a regression model $f : ob \rightarrow R$. For example, weather conditions must be included in ob because they affect HVAC energy consumption and hence affect the reward value.

Usually, the observation vector should include the following items:

- Time information, such as day type (weekdays or weekends) and hour of a day. Most HVAC

systems have time-related operational patterns, so time information has indirect effects on the reward value. Users should decide the specific types of time information based on the knowledge for the target HVAC system.

- Weather conditions, such as outdoor air temperature and solar radiation. Weather conditions affect a building's heating/cooling loads and HVAC equipment efficiency. Users should decide the specific types of weather conditions based on the characteristics of the target HVAC system. For example, wind direction may not affect the energy consumption of a primary chilled water system, so it should not be included in the observation vector.

- HVAC energy metric, such as electric demand and heating/cooling demand. It is included in the observation vector because it may affect the reward value directly.

- Sensory information related to operational constraints. It should be included because it may directly affect the reward value. The specific items should be decided based on the design of a reward function. For example, if a reward function includes an operational constraint for indoor air temperature, the observation vector should include indoor air temperature and its setpoint.

All the items in the observation vector must be normalized to the range 0-1. Min-max normalization is recommended for this control framework.

2. For the length of the history in the state

The length of the history in the state can be determined using Algorithm 1. However, the algorithm contains a hyperparameter $dcorThres$. $dcorThres$ ranges from 0 to 1, and a larger $dcorThres$ means the state will include a longer history of the observation vectors. This thesis simply uses 0.5 for $dcorThres$.

However, if computing power allows, the value of $dcorThres$ should be tuned. Figure 9.1 shows the results of a $dcorThres$ tuning experiment. This experiment is based on the VAVCooling-Pittsburgh-Lightweight scenario. The tested $dcorThres$ values are 0.8, 0.7, 0.6, 0.5, 0.4 and 0,3, which lead to the length of the history $n = 1$ (10 minutes), 9 (1.5 hours), 22 (3.7 hours), 35 (5.8 hours), 51 (8.5 hours), 73 (12.1 hours) respectively. The figure shows that the different lengths of the history do not result in dramatically different control performance in the training, Perturbed3 and Perturbed4 simulators. However, a longer history leads to better energy efficiency improvements in the Perturbed1 and Perturbed2 simulators. This means that a long history can improve

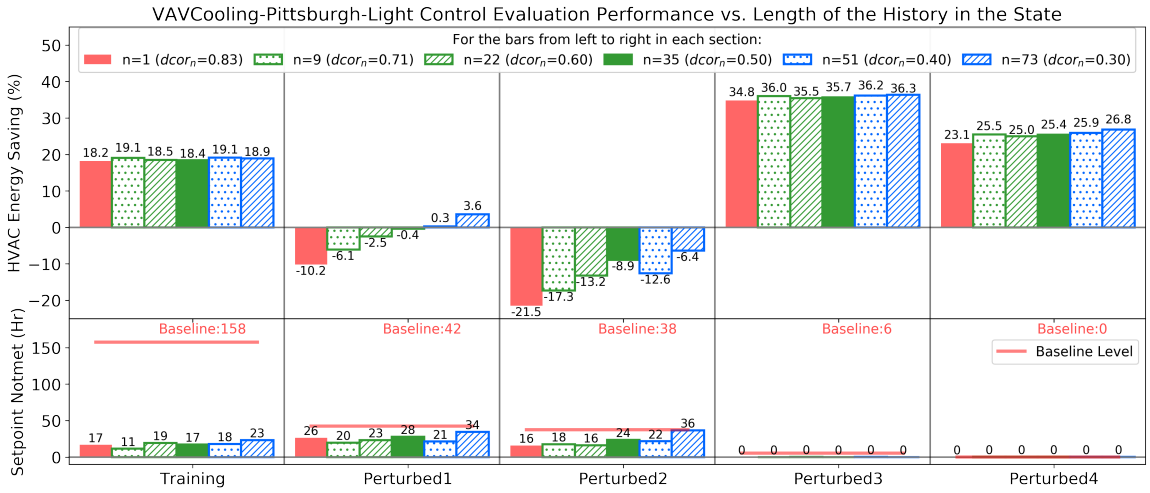

Figure 9.1: Control Performance for Different Values of *dcorThres* in the VAVCooling-Pittsburgh-Lightweight Scenario with the ReLu64-2 Neural Network Model (Note: the results are from the best-performing learning rate)

the resulting control policy's versatility to tolerate the variations in indoor air temperature setpoint. However, a longer history makes the state more complicated, which may jeopardize the convergence of the training. Users should conduct a tuning experiment to determine the most suitable value for *dcorThres*.

Select the action variable Currently, this control framework only supports a single control variable with a discrete action space. Hence, one should select a control variable that has a significant effect on the operation of an HVAC system, such as air-handling-unit supply air temperature setpoint. The control variable can be determined via analyzing the system operation principles, literature review or simulation experiments.

Select the neural network model A more complex neural network model does not necessarily lead to better control performance (e.g., more energy savings). This is because complex neural network models are more difficult to optimize through gradient descent. However, an over-simplified neural network model (e.g., a linear model) may also lead to poor control performance.

If computing power is sufficient, users should tune from a linear model to a narrow and shallow nonlinear model and finally to a wide and narrow nonlinear model. It is usually not necessary to use deep neural network models.

If there is a lack of computing power, ReLu 64-2 can be a default choice.

Select the learning rate The convergence of reinforcement learning is sensitive to its learning rate. Unfortunately, no default learning rate works for all scenarios. Users should tune from a small learning rate (e.g., 5e-6) to larger learning rates until a good convergence is achieved.

Select the control and simulation time step The time step size should neither be too small or too large. A small time step gives an RL agent the flexibility to respond to changes, but it may harm the training convergence if the target HVAC system has a slow response. Also, a small time step increases the computational time of EnergyPlus simulations. A large time step benefits the training convergence and saves the computational time of EnergyPlus simulations, but an RL agent loses the control flexibility.

It is recommended to use 10 minutes for both control and simulation time step. For some slow-response HVAC systems, the control time step size can be longer than the simulation time step size by using action-repeat (i.e., repeat the same control action for multiple simulation time steps).

Assess the convergence of the RL training The convergence of the training is reflected in the *cumulative reward that an agent receives in one simulation episode of the training simulator* ($R_{trainCumulative}$). Users should plot the training evaluation history which shows $R_{trainCumulative}$ at different RL interaction steps. The training evaluation history can be obtained by setting checkpoints during the training (e.g., every 50K interaction steps), which pause the training and use the current trained control policy to control the training simulator to record $R_{trainCumulative}$. If the general trend of $R_{trainCumulative}$ is increasing and then stable, this means the training converges.

Figure 9.2 shows an example of the training evaluation history of different RL agents with different neural network models. It can be seen that some curves are fluctuating without increasing (e.g., the curve for ReLu 256-8), and some are increasing with settlements at high levels (e.g., the curve for ReLu 256-2). This figure clearly shows which agents converge and which do not.

Terminate the training and select an appropriate control policy Different problems (i.e., different HVAC systems, different climates, different reward functions, etc.) have different training behaviors. Hence, it cannot be determined beforehand that whether or not an RL agent could converge and how many interaction steps are needed for the convergence. Besides, since multiple

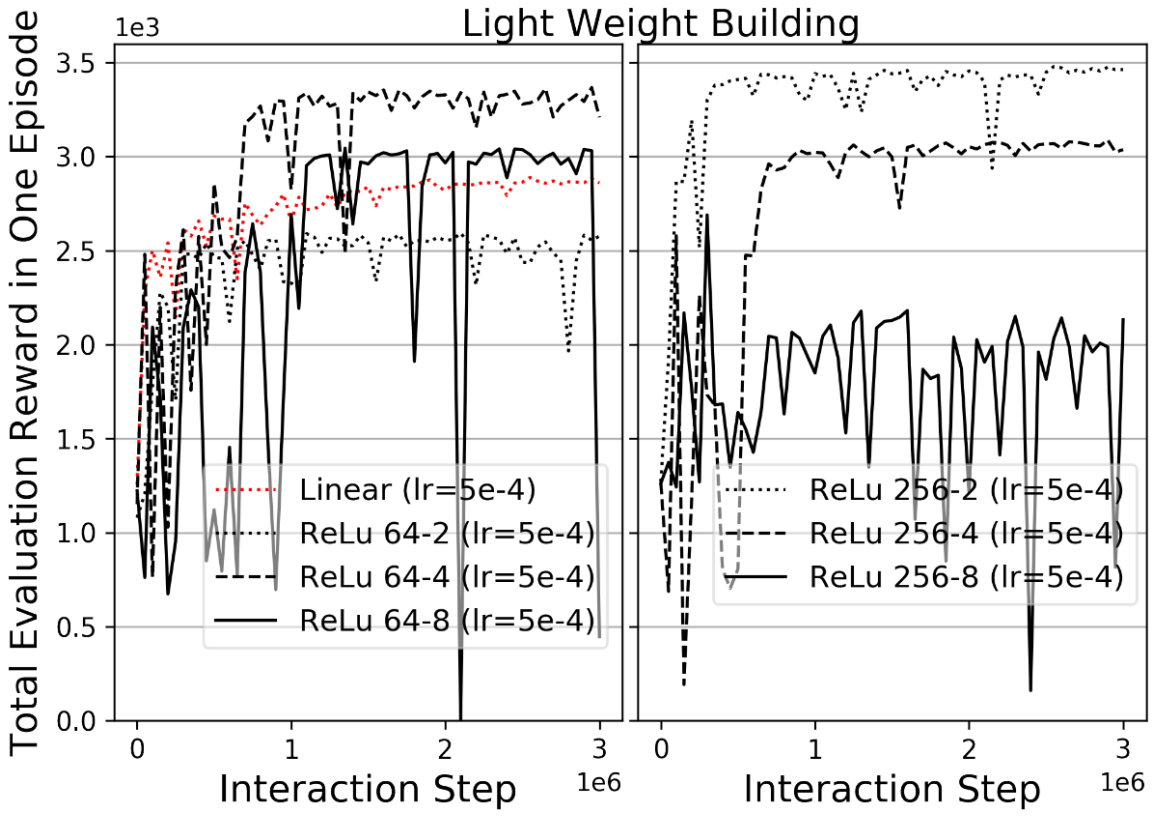

Figure 9.2: Example of the Training Evaluation History

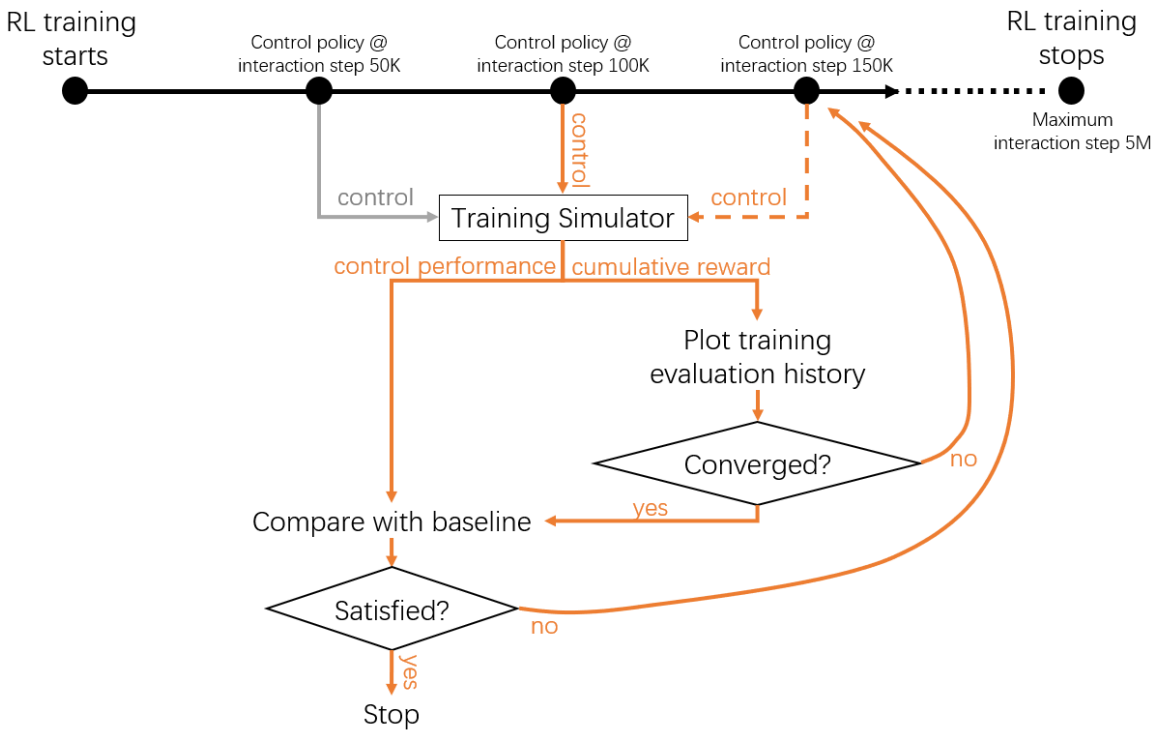

Figure 9.3: A Procedure to Terminate the Training and Select a Control Policy (assuming the checkpoints are at every 50K interaction steps)

control objectives are formulated into a single reward function, a control policy with the maximum cumulative reward may not achieve a balanced control performance. For example, it may have low energy consumption but excessive setpoint notmet time.

Figure 9.3 shows a procedure to terminate the training and select an appropriate control policy. In this procedure, checkpoints are set at every 50K interaction steps. The checkpoints pause the training and use the current control policy to control the training simulator for one episode. Then, users should plot the training evaluation history to assess the convergence. If the training converges, the training control performance will be compared with its baseline. If users are satisfied with the control performance, then the training will be terminated and the control policy is selected. If the training does not converge or the control performance is not satisfactory, the training continues till it reaches the maximum interaction step.

Tune the hyperparameters It cannot be guaranteed that the reinforcement learning converges for all control problems. In addition, a reinforcement learning agent may converge to a locally optimal solution that fails to achieve expected control performance. Hyperparameter tuning is necessary for

these cases.

Theoretically, all hyperparameters could affect results. One should start with the choices of the hyperparameters listed in this thesis (Tables 4.5, 6.4 and 7.6). If the reinforcement learning still does not converge or the control performance is poor, the following hyperparameters can be tuned:

- Reward function. Redesign the reward function to provide an RL agent with a clear rewarding/penalizing signal.
- Gradient-clip threshold. This value limits the magnitude of the gradients during the gradient descent optimization of reinforcement learning. Reducing this value may lead to a more stable convergence (but an RL agent may have less chances to jump out of local optimal regions).
- Learning batch size. This is the n of Equation (2.13). Increasing this value leads to a more accurate estimation for the "true" state-values (see section 2.3.3), but it also increases the computation time.
- Optimizer. Different optimizers, such as Adam, RMSProp, AdaMax, etc., have different characteristics. Users should refer to the manual of an optimizer to select a suitable one.

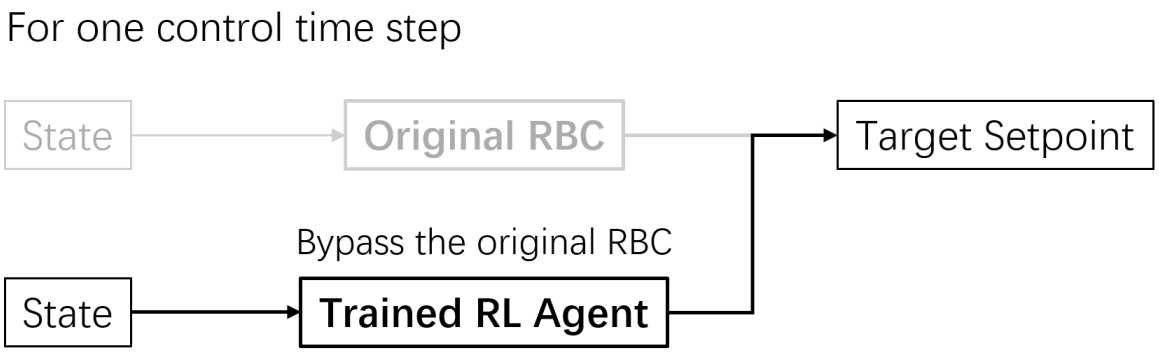

Figure 9.4: Deploy a Trained RL Agent to Substitute an Original Rule-Based Control (RBC)

9.2 For Control Policy Deployment

Calibrate the training simulator This control framework assumes that an "accurate" simulator is accessible for the reinforcement learning training. "Accurate" means the training simulator accurately represents an actual building's behaviors, including operation schedules, energy consumption, thermal response, etc. Hence, the training simulator must be calibrated using actual building operation data.

Deploy the trained RL agent as a static function After the RL training is finished, the trained RL agent becomes a static function, i.e., $f : S_t \rightarrow A_t$. This means that, at each control time step, the trained RL agent uses the current state to calculate the current control action. Hence, the deployment of the RL agent has no difference from a rule-based control logic. As shown in Figure 9.4, the trained RL agent substitutes the original rule-based control logic to provide values for the target setpoint.

Access the sensory data for the observation vector from HVAC systems During the design of the state, users should ensure all the items in the observation vector are accessible in the target HVAC system. Also, users should note resolution differences between actual sensory data and the corresponding data in the training simulator. For example, some indoor air temperature sensors give only integer values (the resolution is 1 $°C$) but EnergyPlus uses 32-bit floating-point for all continuous values. The effect of the data resolution difference is not studied by this control framework.

Check the validity of the sensory data for the observation vector A trained RL agent can only run with valid state inputs. It cannot check the validity of its inputs. Thus, invalid state inputs may cause the trained RL agent to crash or give invalid control actions.

During the deployment, users should also deploy a computational program to automatically check the validity of the input to an RL agent. Some invalid conditions include:

- Data is out of its normal range, such as a negative value for solar radiation.
- Data is not a number.
- Other invalid conditions, such as suspiciously low indoor air temperature.

If any invalidity occurs, the trained RL agent should be terminated, and original rule-based control logic should be turned on.

It is also recommended to deploy a safety logic to block any invalid control actions given by a trained RL agent.

Chapter 10

Conclusion

10.1 Summary of Findings

Whole building energy model (BEM) is a physics-based modeling method to predict a building's thermal and energy performance. It has been widely used by building designers for design decision support, but has seldom been used for HVAC control. This study presents a reinforcement learning (RL) method to use a whole building energy model to develop HVAC supervisory-level control strategies. The derived control strategies can potentially improve the energy efficiency of HVAC systems compared to widely-used rule-based control.

The reinforcement learning method is first evaluated through computer simulations. The simulation experiments include four common HVAC systems, including a multi-zone variable-air-volume (VAV) system for cooling (VAVCooling), a multi-zone VAV system for heating (VAVHeating), a slow response radiant heating system (RadiantHeating) and a multi-chiller chilled water system (ChilledWater). These systems have complex configurations and operational constraints, so their dynamics is highly complicated. For each system, multiple simulation experiment scenarios are created with different climate zones and different building thermal mass levels. The control performance of an RL control policy, including its energy saving compared to rule-based control and operational constraints fulfillment, is first evaluated using the training simulator. The control performance is then evaluated using different perturbed simulators for versatility evaluation. Each perturbed simulator is different from the training simulator in weather conditions, occupancy/plug-load schedules and indoor air temperature setpoint schedules. It is found that:

- The control framework can successfully use BEMs to generate control policies to achieve obvious energy savings (10% to over 30%) and less-than-baseline operational constraint violations in VAVCooling, VAVHeating, and RadiantHeating for all the climate zones and building thermal mass levels. The results support the first hypothesis of the thesis, which states that the control framework can use a BEM to develop a control strategy to improve HVAC energy efficiency.

- ChilledWater is an outlier since the control policies from the control framework have caused a significant amount of operational constraint violations, including the shortcycling and too-low partial load ratio of the chillers. Nevertheless, the control policies have achieved lower-than-baseline supply water temperature setpoint notmet time and a small amount of energy savings (less than 5%).

- The trained control policies from the control framework can still provide significant energy efficiency improvements after the variations in weather conditions (i.e., from typical weather data to actual meteorological year weather data) and occupancy/plug-load schedules (i.e., from deterministic schedules to stochastic schedules). This demonstrates a certain level of versatility of the trained control policies.

- For the VAV systems, the trained control policies cannot tolerate the variations in indoor air temperature setpoint. This indicates that an accurate profile of indoor air temperature setpoint must be obtained when creating a training simulator.

- System types have the most significant effects on the amount of energy savings. For the same HVAC system type, the control framework is robust for the different climates and different building thermal mass levels.

The reward function design is found to be crucial for the convergence and the control performance of the proposed control framework. A reinforcement learning agent should receive a clear signal to reward or penalize a state/action pair. A reward function with the minimum prior knowledge is designed for the VAV systems, where the reward value is only proportional to the system energy consumption and the level of operational constraint fulfillment at current time steps. This design leads to effective RL training that delivers significant energy efficiency improvements. However, for the radiant heating system with a delayed response and an ambiguous energy metric, specially-designed heuristics is necessary to help an RL agent to learn. With the simple heuristics shown in the thesis, the trained RL control policies have also achieved significant improvements for the energy efficiency of the radiant heating system. A limitation of the control framework is that, the reward function simply combines multiple control objectives (e.g., energy saving) and constraints (e.g., thermal comfort, a chiller's cycling time) into a single value output. This may make the reward function over-complicated for some cases, and hence leads to poor control performance. This is shown in the experiments of ChilledWater, where the reward function simply combines three stringent operational constraints into a single output. The trained control policies have led to a significant amount of operational constraint violations and a limited amount of energy savings.

The simulation experiments also study the effects of the neural network model complexity on the control performance. It is found that, there is no obvious relationship between a neural network model and the control performance. In some experiment scenarios like VAVCooling and RadiantHeating, the different neural network models, from a simple linear model to a "deep and wide" nonlinear model, have achieved similar control performance; in some experiment scenarios like VAVHeating

and ChilledWater, the different neural network models have different control performance, but there is not a unified relationship between the neural network model complexity and the control performance. This means that, the deep and wide neural network models do not show any consistent advantages in the control performance compared to the simple neural network models, so the second hypothesis of the thesis is not supported. In general, the experiment results also show that the complex neural network models are more difficult to converge than the simple neural network models. However, the simplest is not always the best. The linear model shows poor control performance in VAVHeating, which is attributed to its insufficient representational capacity. The results indicate that, the neural network model architecture is an important hyperparameter that needs to be tuned for different experiment scenarios.

The control framework is also implemented and deployed for an actual radiant heating system in real-life. The radiant heating system has a unique design and serves a 600 m^2 office building located in Pittsburgh, PA, USA. A four-step deployment procedure is presented, including building energy modeling, model calibration, offline RL training, and deployment. As shown in the simulation experiments, model calibration is an important step to ensure a training simulator can accurately predict an actual system's thermal and energy behaviors. In this case study, the calibrated BEM can accurately predict the 5-min indoor air temperature with less 5% error, and predict the daily heating demand with around 10% error. Then, the control framework is used to train an RL control policy. The trained control policy was deployed in the actual system for 78 days in 2018 spring. A data-driven normalized energy saving analysis shows that, the control policy has saved 16.7% heating demand compared to the original rule-based control. This case study partially demonstrates the practical feasibility and effectiveness of the control framework, and presents a practical implementation and evaluation procedure.

Based on the simulation and implementation results, the usage guidelines are provided. The guidelines are separated into two parts, including the part for the offline training to develop control policies, and the part for the deployment of the control policies. The usage guidelines are served as a starting point for general users to properly use this control framework. Advanced users should further understand the principles of reinforcement learning and HVAC system dynamics to make better use of this framework.

It is worth mentioning that the thesis has no intentions to compare the proposed control framework with the more commonly used control method MPC. This is because the two methods have different application scopes. The control framework is designed to use BEMs for HVAC energy-

efficient control. BEM is a complicated high-order simulation program, so MPC cannot be directly applied for it. The control framework uses BEM-assisted reinforcement learning to generate a control policy that is more energy-efficient than a rule-based control strategy. However, the control policy may not be the optimal one. In contrast, MPC is an optimal control method with optimality guarantee. However, it is designed for the systems whose dynamics can be identified by low-order models. Thus, the two methods are suitable for different control problems.

10.2 Limitations

Since RL for HVAC control, especially the BEM-assisted method, is still a developing topic, this thesis is served as an exploring study on this topic. Therefore, the experiments cover a wide range of different scenarios but lack sufficient in-depth studies on some specific problems. The major limitations of the study are summarized below:

- The hyperparameters of reinforcement learning, such as the neural network architecture, learning rate, the length of the history in the state, etc., have not been adequately tuned. Thus, the control performance results shown in the thesis may not be the best ones. Some poor control performance results may be significantly improved by extensive hyperparameter tuning.
- Only "one-shot" experiment has been conducted for each experiment scenario, so the statistical significance of the results cannot be determined. Thus, the results obtained from the experiments may not be generalized to the conditions beyond the experiment settings.
- The thesis contains only empirical results with limited theoretical investigations. This is partly because the experiment environments are over-complicated so the number of influencing factors is very high. The complicated experiment environments are designed according to the general objective of the thesis, which is to develop a control method for complicated HVAC systems.
- The thesis does not compare different RL algorithms. Since RL is still under development, there are new algorithms published every month. Some algorithms may be more suitable and efficient for HVAC control than others. For example, the RL algorithm in this thesis, A3C, is not sample-efficient and it needs multiple paralleled "local RL workers" to collect a large number of samples for learning. Recent research has developed more sample-efficient RL algorithms, such as Hester and Stone (2013) who uses a random forest model to approximate system dynamics, Schulman et al. (2015) who uses importance sampling to guide the gradient descent update, and Buckman et al. (2018) who combines model-based and model-free RL approaches to increase the RL sample efficiency.
- The control framework is trying to fit multi-objective optimization problems (e.g., minimize energy consumption and operational constraint violations) into a single objective optimization method, i.e., all control objectives are formulated in a single reward function through "weighted sum". This design may make the reward function design over-complicated, which harms the convergence of RL. Besides, additional hyperparameter tuning is necessary to balance different

control objectives in the reward function. There are multiple "native" multi-objective reinforcement learning approaches in the literature, such as W-learning approach, ranking approach, geometric approach, etc. (Liu et al., 2015). They are worth studying in the future.

- The control method assumes a single reinforcement learning agent, so the control action space contains only one control variable. This limits its application for the control of multiple setpoints, such as controlling multiple zone air temperature setpoints or coordinating the operation of multiple systems. Further energy efficiency improvements are expected if an RL-trained control policy could handle multiple control variables simultaneously.

10.3 Future Work

As an exploring study on the topic of BEM-assisted RL for HVAC control, this thesis opens some questions that need to be answered in the future.

Firstly, the control framework still requires extensive hyperparameter tuning for a new scenario. Hyperparameter tuning is computationally expensive, which may potentially limit the practical feasibility of the control framework. Larger-Scale simulation experiments are hence necessary to derive a guideline for the selection of key hyperparameters. The future experiments should also include Monte Carlo analysis to provide statistical confidence for the results. In addition, an automatic hyperparameter tuning software tool should be developed to minimize manual work.

The future work of this thesis should also include a thorough study on the versatility of a trained RL control policy. The versatility study should consider the effects of different types of variations in HVAC dynamic operational conditions, such as low and high occupancy density patterns, extreme and mild weather conditions, etc. In addition, the versatility study should consider potential modeling errors in HVAC static properties, such as equipment efficiencies, building thermal properties, etc. This work could help to create a practical guidelines for building energy modeling, including the modeling assumptions and the model calibration error thresholds. Besides, methods to improve the versatility of an RL control policy should be explored.

It is also necessary to study the adaptability of an RL control policy to continuously changing building characteristics. Buildings are dynamic objects. A building's thermal and energy behaviors may change as time goes by. For example, the usage pattern of a building can be significantly changed after a new tenant moves in. A control policy generated by the control framework is not adaptive since it is used as a static function. Future work should develop a method to allow a pre-trained RL control policy to adapt to changing building characteristics in a fast and safe manner.

The thesis has not conducted any theoretical investigations on the results. In the future, theoretical studies should be conducted based on simpler experiment environments, such as a single room with a heater or cooler. Questions such as "how versatile is a trained control policy" or "how to interpret a trained control policy" can be more definitely answered. The results from the theoretical studies can help the design of the control framework for more complicated systems.

Bibliography

Aftab, M., Chen, C., Chau, C. K., and Rahwan, T. (2017). Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system. *Energy and Buildings*, 154:141–156.

American Society of Heating, Ventilating, and Air Conditioning Engineers (2016). ANSI/ASHRAE/IES Standard 90.1-2016 Energy Standard for Buildings Except Low-Rise Residential Buildings. Standard, American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia.

Anderson, C. W. (1986). *Learning and Problem Solving with Multilayer Connectionist Systems*. PhD thesis, University of Massachusetts, Amherst.

Anderson, C. W., Hittle, D. C., Katz, A. D., and Kretchmar, R. (1997). Synthesis of reinforcement learning, neural networks and pi control applied to a simulated heating coil. *Artificial Intelligence in Engineering*, 11(4):421 – 429. Applications of Neural Networks in Process Engineering.

Ascione, F., Bianco, N., De Stasio, C., Mauro, G. M., and Vanoli, G. P. (2016). Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. *Energy and Buildings*, 111:131–144.

ASHRAE (2014). Guideline 14-2014, Measurement of Energy, Demand, and Water Savings. Standard, American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia.

ASHRAE (2017). Thermal Environmental Conditions for Human Occupancy. Standard, American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia.

AT&T (1950). Claude shannon demonstrates machine learning. Accessed on May 5, 2019.

Barrett, E. and Linder, S. (2015). Autonomous hvac control, a reinforcement learning approach. In Bifet, A., May, M., Zadrozny, B., Gavalda, R., Pedreschi, D., Bonchi, F., Cardoso, J., and

Spiliopoulou, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 3–19, Cham. Springer International Publishing.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846.

Bellemare, M. G., Dabney, W., and Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. *arXiv e-prints*, page arXiv:1707.06887.

Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P. S., and Munos, R. (2015). Increasing the Action Gap: New Operators for Reinforcement Learning. *arXiv e-prints*, page arXiv:1512.04860.

Bellman, R. (1957). A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684.

Blundell, C., Uria, B., Pritzel, A., Li, Y., Ruderman, A., Leibo, J. Z., Rae, J., Wierstra, D., and Hassabis, D. (2016). Model-Free Episodic Control. *arXiv e-prints*, page arXiv:1606.04460.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *CoRR*, abs/1606.01540.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. (2018). Sample-Efficient Reinforcement Learning with Stochastic Ensemble Value Expansion. *arXiv e-prints*, page arXiv:1807.01675.

BuildSimHub, Inc. (2018). Buildsimhub. Accessed on Jan 18, 2018.

Chen, J., Augenbroe, G., and Song, X. (2018a). Lighted-weighted model predictive control for hybrid ventilation operation based on clusters of neural network models. *Automation in Construction*, 89(February):250–265.

Chen, X., Wang, Q., and Srebric, J. (2016). Occupant feedback based model predictive control for thermal comfort and energy optimization: A chamber experimental evaluation. *Applied Energy*, 164:341–351.

Chen, Y., Norford, L. K., Samuelson, H. W., and Malkawi, A. (2018b). Optimal control of hvac and window systems for natural ventilation through reinforcement learning. *Energy and Buildings*, 169:195 – 205.

Chong, A., Lam, K. P., Pozzi, M., and Yang, J. (2017). Bayesian calibration of building energy models with large datasets. *Energy & Buildings*, 154:343–355.

Chong, A. and Menberg, K. (2018). Guidelines for the bayesian calibration of building energy models. *Energy and Buildings*, 174:527 – 547.

Coninck, R. D. and Helsen, L. (2016). Practical implementation and evaluation of model predictive control for an office building in brussels. *Energy and Buildings*, 111:290 – 298.

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

Crawley, D. B., Lawrie, L. K., Pedersen, C. ., Liesen, R. J., Fisher, D. E., Strand, R. K., Taylor, R. D., Winkelmann, F. C., Buhl, W. F., Huang, Y. J., and Erdem, A. E. (1998). Beyond BLAST and DOE-2: EnergyPlus, a New-Generation Energy Simulation Program. In *ACEEE Summer Study on Energy Efficient Buildings*, Pacific Grove, California.

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

DeepMind (2019). Alphastar: Mastering the real-time strategy game starcraft ii. Accessed on May 15, 2019.

Dong Li, Zhao, D., Zhu, Y., and Xia, Z. (2015). Thermal comfort control based on mec algorithm for hvac systems. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6.

Energy Market Authority of Singapore (2018). Singapore energy statistics 2018.

ENERGY STAR (2017). Portfolio Manager Technical Reference: Climate and Weather. Technical reference. Accessed on Aug 27, 2018.

EnergyPlus (2019). Input Output Reference. Technical reference. Accessed on May 30, 2019.

Fanger, P. O. (1970). *Thermal comfort: analysis and applications in environmental engineering*. Danish Technical Press, Copenhagen, Denmark.

Fazenda, P., Veeramachaneni, K., Lima, P., and O'Reilly, U.-M. (2014). Using Reinforcement Learning to Optimize Occupant Comfort and Energy Usage in HVAC Systems. *Journal of Ambient Intelligence and Smart Environment*, 6(6):675–690.

Fielsch, S., Grunert, T., Stursberg, M., and Kummert, A. (2017). Model Predictive Control for Hydronic Heating Systems in Residential Buildings. *IFAC-PapersOnLine*, 50(1):4216–4221.

Fortunato, M., Gheshlaghi Azar, M., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. (2017). Noisy Networks for Exploration. *arXiv e-prints*, page arXiv:1706.10295.

Fragkiadaki, K. (2018). Carnegie mellon university lecture notes in iterative linear quadratic regulator.

Gao, G., Li, J., and Wen, Y. (2019). Energy-Efficient Thermal Comfort Control in Smart Buildings via Deep Reinforcement Learning. *arXiv e-prints*, page arXiv:1901.04693.

García, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and practice —a survey. *Automatica*, 25(3):335 – 348.

Gong, X. and Claridge, D. E. (2006). Study of mullion heating and cooling system of intelligent workplace at carnegie mellon university.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA. `http://www.deeplearningbook.org`.

He, F. S., Liu, Y., Schwing, A. G., and Peng, J. (2016). Learning to Play in a Day: Faster Deep Reinforcement Learning by Optimality Tightening. *arXiv e-prints*, page arXiv:1611.01606.

Hester, T. and Stone, P. (2013). Texplore: Real-time sample-efficient reinforcement learning for robots. *Mach. Learn.*, 90(3):385–429.

Huang, H., Chen, L., and Hu, E. (2015). A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study. *Building and Environment*, 89:203 – 216.

Huo, T., Ren, H., Zhang, X., Cai, W., Feng, W., Zhou, N., and Wang, X. (2018). China's energy consumption in the building sector: A statistical yearbook-energy balance sheet based splitting method. *Journal of Cleaner Production*, 185:665 – 679.

Jain, A., Nghiem, T. X., Morari, M., and Mangharam, R. (2018). Learning and control using gaussian processes: Towards bridging machine learning and controls for physical systems. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '18, pages 140–149, Piscataway, NJ, USA. IEEE Press.

James J. Hirsch Associates (1998). PowerDOE. Accessed on May 6, 2019.

James J. Hirsch Associates (2019). Doe-2. Accessed on May 7, 2019.

Jia, R., Jin, M., Sun, K., Hong, T., and Spanos, C. (2019). Advanced building control via deep reinforcement learning. *Energy Procedia*, 158:6158 – 6163. Innovative Solutions for Energy Transitions.

Kazmi, H., Suykens, J., Balint, A., and Driesen, J. (2019). Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. *Applied Energy*, 238:1022 – 1035.

Killian, M. and Kozek, M. (2018). Implementation of cooperative fuzzy model predictive control for an energy-efficient office building. *Energy and Buildings*, 158:1404 – 1416.

Kissock, J., Reddy, T., and Claridge, D. (1998). Ambient-temperature regression analysis for estimating retrofit savings in commercial buildings. *Journal of Solar Energy Engineering, Transactions of the ASME*, 120(3):168–176.

Kusuda, T. (1976). NBSLD, the Computer Program for Heating and Cooling Loads in Buildings.

Kusuda, T. (1999). Early history and future prospects of building system simulation. *Proceedings of Building Simulation*.

Kwak, Y., Huh, J., and Jang, C. (2015). Development of a model predictive control framework through real-time building energy management system data. *Applied Energy*, 155:1–13.

Lawrence Berkeley National Laboratory (2016). Building Controls Virtual Test Bed.

Li, S., Joe, J., Hu, J., and Karava, P. (2015). System identification and model-predictive control of office buildings with integrated photovoltaic-thermal collectors, radiant floor heating and active thermal storage. *Solar Energy*, 113:139 – 157.

Liang, W., Quinte, R., Jia, X., and Sun, J.-Q. (2015). MPC control for improving energy efficiency of a building air handler for multi-zone VAVs. *Building and Environment*, 92:256–268.

Lindelöf, D., Afshari, H., Alisafaee, M., Biswas, J., Caban, M., Mocellin, X., and Viaene, J. (2015). Field tests of an adaptive, model-predictive heating controller for residential buildings. *Energy and Buildings*, 99:292 – 302.

Liu, C., Xu, X., and Hu, D. (2015). Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398.

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

Ma, J., Qin, S. J., and Salsbury, T. (2014a). Application of economic mpc to the energy and demand minimization of a commercial building. *Journal of Process Control*, 24(8):1282 – 1291. Economic nonlinear model predictive control.

Ma, Y., Matuško, J., and Borrelli, F. (2014b). Stochastic Model Predictive Control for Building HVAC Systems : Complexity and Conservatism. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, pages 1–16.

MacArthur, J. and Foslien, W. (1993). A Novel Predictive Strategy for Cost-optimal Control in Buildings. In *ASHRAE Winter Conference*, pages 1025–1036, Chicago, IL.

Mahdavi, A. (2001). Simulation-based control of building systems operation. *Building and Environment*, 36(6):789 – 796. Building and Environmental Performance Simulation: Current State and Future Issues.

May-Ostendorp, P., Henze, G. P., Corbin, C. D., Rajagopalan, B., and Felsmann, C. (2011). Model-predictive control of mixed-mode buildings with rule extraction. *Building and Environment*, 46(2):428–437.

Michie, D. and Chambers, R. A. (1968). BOXES: An experiment in adaptive control. In Dale, E. and Michie, D., editors, *Machine Intelligence 2*, pages 137—152. Oliver and Boyd, Edinburgh, Scotland.

Miezis, M., Jaunzems, D., and Stancioff, N. (2017). Predictive Control of a Building Heating System. *Energy Procedia*, 113:501–508.

Miller, C., Quintana, M., and Glazer, J. (2019). Twenty years of building simulation trends: Text mining and topic modeling of the bldg-sim email list archive.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. In *33rd International Conference on Machine Learning*, volume 48, New York, NY, USA.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning.

Morris, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174.

Mozer, M. C. (1998). The Neural Network House: An Environment that Adapts to its Inhabitants. In *AAAI Spring Symposium, Intelligent Environments*, volume 58, Palo Alto, CA, USA.

Nagy, A., Kazmi, H., Cheaib, F., and Driesen, J. (2018). Deep Reinforcement Learning for Optimal Control of Space Heating. *ArXiv e-prints*.

O'Dwyer, E., De Tommasi, L., Kouramas, K., Cychowski, M., and Lightbody, G. (2017). Prioritised objectives for model predictive control of building heating systems. *Control Engineering Practice*, 63(March):57–68.

onebuilding.org (2019). Bldg-sim-Users of building energy simulation tools. Accessed on May 6, 2019.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep Exploration via Bootstrapped DQN. *arXiv e-prints*, page arXiv:1602.04621.

Ostrovski, G., Bellemare, M. G., Oord, A. v. d., and Munos, R. (2017). Count-Based Exploration with Neural Density Models. In *34 th International Conference on Machine Learning*, Sydney, Australia.

Peng, K. S. and Morrison, C. T. (2016). Model predictive prior reinforcement learning for a heat pump thermostat. In *IEEE International Conference on Automatic Computing: Feedback Computing*, volume 16.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA.

Razmara, M., Maasoumy, M., Shahbakhti, M., and Robinett, R. D. (2015). Optimal exergy control of building HVAC system. *Applied Energy*, 156:555–565.

Ruelens, F., Claessens, B. J., Vandael, S., De Schutter, B., Babuška, R., and Belmans, R. (2017). Residential demand response of thermostatically controlled loads using batch reinforcement learning. *IEEE Transactions on Smart Grid*, 8(5):2149–2159.

Ruelens, F., Iacovella, S., Claessens, B. J., and Belmans, R. (2015). Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning. *Energies*, 8(8):8300–8318.

Schmidt, M., Moreno, M. V., Schülke, A., Macek, K., Mařík, K., and Pastor, A. G. (2017). Optimizing legacy building operation: The evolution into data-driven predictive cyber-physical systems. *Energy and Buildings*, 148:257 – 279.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. (2015). Trust Region Policy Optimization. *arXiv e-prints*, page arXiv:1502.05477.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv e-prints*, page arXiv:1707.06347.

Shi, J., Yu, N., and Yao, W. (2017). Energy Efficient Building HVAC Control Algorithm with Real-time Occupancy Prediction. *Energy Procedia*, 111(September 2016):267–276.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., and Sifre, L. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359.

Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D' Innocenzo, A., and Mangharam, R. (2018). Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied Energy*, 226:1252 – 1272.

Sun, B., Luh, P. B., Jia, Q., and Yan, B. (2013). Event-based optimization with non-stationary uncertainties to save energy costs of hvac systems in buildings. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 436–441.

Sutton, R. S. and Barto, A. G. (2017). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, second edi edition.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2008). Measuring and testing dependence by correlation of distances. *arXiv e-prints*, page arXiv:0803.4101.

Tesauro, G. (1995). Temporal difference learning and td-gammon. *Commun. ACM*, 38(3):58–68.

The U.S. Department of Energy (2019a). EnergyPlus. Accessed on May 6, 2019.

The U.S. Department of Energy (2019b). Spawn-of-EnergyPlus (SOEP). Accessed on May 6, 2019.

The U.S. Energy Information Administration (2017). How much energy is consumed in u.s. residential and commercial buildings? Accessed on May 5, 2019.

The U.S. Green Building Council (2019). Leed v4 for building design and construction.

Turing, A. M. (1948). Intelligent Machinery, A Heretical Theory*. *Philosophia Mathematica*, 4(3):256–260.

Urieli, D. and Stone, P. (2013). A learning agent for heat-pump thermostat control. volume 2, pages 1093–1100. cited By 20.

Valladares, W., Galindo, M., Gutiérrez, J., Wu, W.-C., Liao, K.-K., Liao, J.-C., Lu, K.-C., and Wang, C.-C. (2019). Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm. *Building and Environment*, 155:105 – 117.

van Hasselt, H., Guez, A., and Silver, D. (2015). Deep Reinforcement Learning with Double Q-learning. *arXiv e-prints*, page arXiv:1509.06461.

Vana, Z., Cigler, J., Siroky, J., Zacekova, E., and Ferkl, L. (2014). Model-based energy efficient control applied to an office building. *Journal of Process Control*, 24(6):790 – 797. Energy Efficient Buildings Special Issue.

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

Vázquez-Canteli, J. R., Ulyanin, S., Kämpf, J., and Nagy, Z. (2019). Fusing tensorflow with building energy simulation for intelligent energy management in smart cities. *Sustainable Cities and Society*, 45:243 – 257.

Wang, S. and Ma, Z. (2007). Supervisory and Optimal Control of Building HVAC Systems: A Review. *HVAC&R Research*, 14(1):3–32.

Wang, Y., Velswamy, K., and Huang, B. (2017). A Long-Short Term Memory Recurrent Neural Network Based Reinforcement Learning Controller for Office Heating Ventilation and Air Conditioning Systems. *Processes*, 5(46).

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and de Freitas, N. (2015). Dueling Network Architectures for Deep Reinforcement Learning. *arXiv e-prints*, page arXiv:1511.06581.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK.

Wei, T., Wang, Y., and Zhu, Q. (2017). Deep reinforcement learning for building hvac control. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6.

West, S. R., Ward, J. K., and Wall, J. (2014). Trial results from a model predictive control and optimisation system for commercial building hvac. *Energy and Buildings*, 72:271 – 279.

Winkelmann, F., Lokmanhekim, M., Mitchell, H. C., Rosenfeld, A. H., Graven, R. M., Hunn, B. D., and Cumali, Z. (1977). CAL-ERDA, A New Computer Program for Building Energy Analysis,.

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

Yu, N., Salakij, S., Chavez, R., Paolucci, S., Sen, M., and Antsaklis, P. (2017). Model-based predictive control for building energy management: Part ii –experimental validations. *Energy and Buildings*, 146:19 – 26.

Yu, Z. and Dexter, A. (2010). Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning. *Control Engineering Practice*, 18(5):532 – 539.

Zhang, Z., Chong, A., Pan, Y., and Lam, K. P. (2018a). Multi-objective Whole Building Energy Model Calibration for Model-based Optimal Control of HVAC Systems (Submitted). In *Building Simulation and Optimization 2018*, Cambridge, England.

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

Zhang, Z., Chong, A., Pan, Y., Zhang, C., Lu, S., and Lam, K. P. (2018b). A Deep Reinforcement Learning Approach to Using Whole Building Energy Model for HVAC Optimal Control. In *2018 Building Performance Analysis Conference and SimBuild*, Chicago, IL, USA.

Zhang, Z. and Lam, K. P. (2017). An Implementation Framework of Model Predictive Control for HVAC Systems: A Case Study of Energyplus Model-Based Predictive Control. In *ASHRAE 2017 Annual Conference*, Long Island, CA, USA.

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

Zhao, J. (2015). *Design-Build-Operate Energy Information Modeling (DBO-EIM) for Occupant-oriented Predictive Building Control*. PhD thesis, Carnegie Mellon University.

Zhao, J., Lam, K. P., Ydstie, B. E., and Karaguzel, O. T. (2015). Energyplus model-based predictive control within design–build–operate energy information modelling infrastructure. *Journal of Building Performance Simulation*, 8(3):121–134.